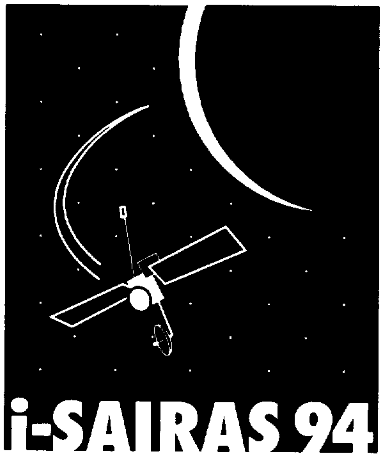
3rd International Symposium on Artificial Intelligence, Robotics, and Automation for Space

PROCEEDINGS

October 18–20, 1994

Pasadena, California, U.S.A.

October 1994
JPL Publication 94-23



Küenstliche Intelligenz, Robotik, und
Automatisierung für die Raumfahrt

Kunstig Intelligens, Robotter, og
Automatisering for Rumfart

Inteligencia Artificial, Robotica, y
Automatización para el Espacio

Intelligence Artificielle, Robotique,
et Automatisation pour l'Espace

Intelligenza Artificiale, Robotica,
ed Automazione per lo Spazio

Kunstmatige Intelligentie, Robotica,
en Automatisering ten behoeve van
Ruimtevaart

宇宙用人工知能/ロボット/オートメーション

Artificial Intelligence, Robotics, and
Automation for Space

This publication was prepared by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

FOREWORD

Artificial intelligence, robotics, and automation represent key technologies for space endeavors of the future. i-SAIRAS 94 — like the two symposia that preceded it — has the objective of creating an international forum that will facilitate an effective exchange of information and cooperation among the many engineers, researchers, and managers who are developing and applying these technologies to space programs.

The first two i-SAIRAS symposia successfully provided a mechanism for people involved in space automation and robotics (A&R) to form a sense of community: to get to know one another and develop common bonds. The result has been a large increase in communication among these professionals. i-SAIRAS 94 continues to widen and strengthen this worldwide community of space A&R professionals by providing a forum for talks on successful applications, ongoing applications, and research and development in space A&R. This symposium also includes presentations that place specific projects in the context of national programs, along with talks about the recent history of space A&R, current programs, current technical activities, and the future plans of national space agencies.

On behalf of this year's other chairpersons — Ichiro Nakatani and François Allard — and the program committee, I would like to welcome the participants in this year's symposium and express my gratitude to the many participants and speakers who have contributed so much in making this event a success.

Melvin Montemerlo

Melvin Montemerlo, Chairperson

ABSTRACT

The Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS 94) is being held October 18–20, 1994, in Pasadena, California, USA. This symposium is jointly sponsored by the National Aeronautics and Space Administration (NASA), the European Space Agency, and the National Space Development Agency of Japan, and is hosted by the Jet Propulsion Laboratory (JPL) of the California Institute of Technology. JPL is NASA's lead center for automated planetary exploration.

i-SAIRAS 94 features more than 100 presentations covering a wide variety of technical and programmatic topics, ranging from underlying basic technology to specific applications of artificial intelligence and robotics to space missions. i-SAIRAS 94 also features a special workshop on planning and scheduling that parallels other symposium technical sessions.

i-SAIRAS 94 provides scientists, engineers, and managers with a unique opportunity to exchange theoretical ideas, practical results, and program plans in such diverse areas as space mission control, space vehicle processing, scientific data analysis, autonomous spacecraft, space robots and rovers, satellite servicing, and intelligent scientific instruments.

Contents

PLENARY SESSIONS

Session PL.1

Presentations by Symposium Chairs
Tuesday, 18 October 1994

The first three presentations of i-SAIRAS 94 describe trends and current developments in space automation and robotics in Japan, Europe, and the United States.

A New Era of Space Exploration: "Smaller, Faster, and Cheaper" with AI and Robotics

Ichiro Nakatani
Institute of Space and Astronautical Science
Japan

Large-scale missions are now in jeopardy all over the world because they need large amounts of money, long development periods, and are easily influenced by political and economic conditions within the countries that support them. At the same time, the nationalistic urge to send human beings to other planets has disappeared since the Cold War ended.

Suddenly, a new space era has opened up with the key words "smaller, faster, and cheaper." In this new era, AI and robotics are the key players in space. Almost anything that can be done in space by human beings can be done by robots, and in a much better way, as long as the system provides proper robot-friendly interfaces. We are in an age of great transition — from large-scale missions to smaller ones where space AI and robotics are rapidly becoming more and more important.

Merging Technologies for Space System Automation in Europe

François Allard
European Space Research and Technology Centre
The Netherlands

AI, autonomy, and robotics for space system automation are increasingly growing toward synergy and integration of technologies. This talk will review the achievements in individual areas and describe the plans, as they are known, for future development in Europe.

The Evolution of U.S. Space Automation and Robotics from i-SAIRAS 90 to i-SAIRAS 94

Melvin Montemerlo
NASA
USA

This session will look at the evolution of AI and robotics for space during the time between i-SAIRAS 90 and i-SAIRAS 94. It will describe the advances that were anticipated then, what actually happened, and the state of AI and robotics for space right now. It will conclude with examples of some things that i-SAIRAS has caused to happen in the field of space automation and robotics.

Session PL.2

Keynote Speech
Tuesday, 18 October 1994

Robots beyond the Lab and Factory

William L. (Red) Whittaker
Carnegie Mellon University
USA

A decade ago, the ancestors of today's field robots were feeble laboratory curiosities. Today, these machines navigate highways, mine coal, and service our nuclear facilities. A decade ago, machine vision, machine planning, and robot autonomy were scientific mysteries. Today, some of the scientific abstractions have been broken and some of the activity has moved from the laboratory to enterprise. This talk profiles the evolution of robotics technology and speculates on robotics science of the future.

This presentation inquires into the world of advanced robotics — what they are, how they work, what they do, and where they are going. It considers such questions as, What is possible in the world of robots? How do the best robots sense, reason, and move? How are they changing our world? What are the scientific drivers of the future?

Red Whittaker develops advanced robots and their technologies. His machines clean up nuclear accidents, navigate rugged terrain, mine coal, and explore active volcanoes. Red is a pioneer in the specialty of field robots — competent machines that work outside factories. Red is a principal research scientist with Carnegie Mellon University's world-renowned Robotics Institute, and chief scientist of RedZone Robotics, Inc. He holds doctor's and master's degrees from Carnegie Mellon and a bachelor's from Princeton. He is committed to the development and use of advanced robots in the world.

4

Session PL.3

Artificial Intelligence/New Robotic Systems
Thursday, 20 October 1994

Expert Systems at NASA: Mining the Golden Nuggets, Building for the Future

Peter Friedland
NASA Ames Research Center
USA

The NASA Artificial Intelligence Program has achieved an impressive record of practical successes in delivering expert systems and other artificial intelligence technologies for solving real NASA problems. This has been done by the classic technique of solving important — but technologically simple — problems first, then building the necessary confidence in the user community for tackling longer term, more challenging issues.

This talk will survey those "golden nuggets" of operational success for a wide range of NASA missions. It will also discuss the research necessary to achieve equivalent success for the next phase of more difficult problems.

The Next Decade of Space Robotics

Dave Lavery
NASA
USA

Charles Weisbin
Jet Propulsion Laboratory
USA

In the same way that the launch of Yuri Gagarin in April 1961 announced the beginning of human space flight, last year's flight of the German ROTEX robot flight experiment is heralding the start of a new era of space robotics. After a gap of twelve years since the introduction of a new capability in space remote manipulation, ROTEX is the first of at least ten new robotic systems and experiments that will fly before the year 2000.

6

N95-23673

The Next Decade of Space Robotics

Dave Lavery
Telerobotics Program Manager
Office of Advanced Concepts and Technology
National Aeronautics and Space Administration
Washington, DC 20546
dave.lavery@hq.nasa.gov
tel:202-358-4684
fax: 202-358-2697

Charles Weisbin
Telerobotics Project Manager
Jet Propulsion Laboratory
Pasadena, CA 91109
charles_r_weisbin@ccmail.jpl.nasa.gov

KEY WORDS AND PHRASES

Robotics applications, robotic exploration, robotic servicing, space robotics

ABSTRACT

In the same way that the launch of Yuri Gagarin in April 1961 announced the beginning of human space flight, last year's flight of the German ROTEX robot flight experiment is heralding the start of a new era of space robotics. After a gap of twelve years since the introduction of a new capability in space remote manipulation, ROTEX is the first of at least ten new robotic systems and experiments which will fly before the year 2000.

Biting Off Too Much

Historically, the space robotics community has pursued the goal of creating fully autonomous, self-contained robotic systems with considerable onboard intelligence as the next major objective in space robotics evolution. Systems such as the Flight Telerobotic Servicer (FTS) were intended to provide near-human levels of intelligence and dexterity, capable of interpreting very high level command structures and autonomously executing the commands without human intervention. The robot was designed to replace a full-time human operator with automated sensing, perception, planning and reasoning sufficient to conduct daily operations.

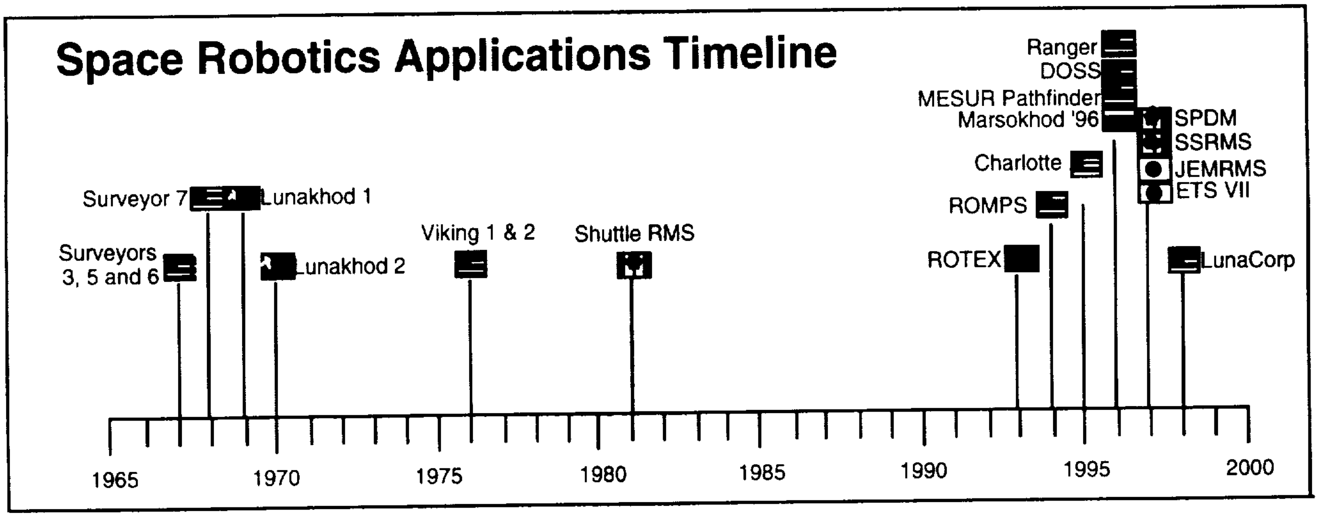
Since the initiation of the FTS and similar ambitious undertakings, the robotics community has gained new understanding of the research still required to create the technologies needed for such systems.

A New Focus

While the technology to support fully autonomous intelligent robotics is not yet available, operational needs for capable remote manipulation and locomotion still exist. In the space robotics arena, a significant paradigm shift is taking place to contend with these needs. Rather than attempting to force the use of immature technology to emulate the "smarts" of a local astronaut operator, the new focus is to utilize advanced teleoperation technology to move the operator from close proximity on-orbit to the ground. Technology elements including predictive displays, low-level reactive planners, sensor-based command execution and dynamic world modeling enable the ability to contend with problems associated with relocation of the operator, such as time-delayed communications, limited viewing options and limited command stream bandwidth. While there is still a long-term goal of developing intelligent autonomy for robots, the short-term goal has become the development of technology to push forward "intelligent teleoperation."

The major impact of this shift in development philosophy is the new opportunity to move robotics out of the laboratory and into the field. The maturation of advanced teleoperation technologies has helped increase confidence in

8


Space Robotics Applications Timeline

the ability of robotic systems to robustly perform real tasks. With this increased confidence has come the acceptance of the potential benefits offered by space robotics technology, and the challenge to "fly it and prove it" with a series of robotic flight experiments and demonstrations. The new push to "get things flying" will yield multiple new space robotic systems before the end of the 1990's.

Robotic systems to be flown during the next five years fall into three categories: Extra-Vehicular Robotic (EVR) servicers, science payload servicers, and planetary rovers.

EVR Servicers

The EVR servicer systems are robotic systems deployed in Earth orbit for use outside of pressurized, controlled environments. Such systems are typified by the Shuttle RMS, which was first flown on the STS-2 mission is 1981. Target applications for these systems include on-orbit satellite assembly, maintenance, repair and servicing, robotic enhancement of Shuttle payload bay operations, and ground-control robotic servicing of external Space Station payloads.

Canada is providing two space robots for use on the International Space Station. The Space Station Remote Manipulator System (SSRMS) is a 55-foot long, 7-Degree Of Freedom (7-DOF) manipulator similar to the Shuttle RMS. Designed to maneuver and locate large payloads along the Space Station truss structure, the SSRMS can transfer power, data and video signals from attached payloads via the latching end effectors at both ends of the arm.

The second Canadian system is the Special Purpose Dexterous Manipulator (SPDM), a dual-arm dexterous robotic system composed of two 7-DOF manipulators, a Power-Data Grapple Fixture, and supporting structures and tooling. The SPDM is controlled during teleoperations with two 3-DOF hand controllers and via keyboard entry and/or preprogrammed sequences for automated trajectory control. Each manipulator is controlled separately, in addition to independent control for the SPDM body and the SSRMS (during operations where the SPDM is positioned by the SSRMS).

At the same time, Japan is preparing a dual-manipulator system as an element of the Space Station Japanese Experiment Module (JEM). Composed of the Main Arm and Small Fine Arm, the JEM Remote Manipulator System (JEMRMS) is intended to provide maintenance, servicing and changeout of science packages placed on the JEM exposed experiment carrier. The Main Arm, similar in configuration to the SRMS and SSRMS, is a 6-DOF positioning tool used to transport large payloads and provide coarse positioning for smaller, more dexterous manipulators. The Small Fine Arm is a 6-DOF manipulator which can be operated either from the end of the Main Arm, or from a fixture on the exposed experiment facility.

Under development by Martin-Marietta Corporation and the NASA Johnson Space Center, the Dexterous Orbiter Servicing System (DOSS) is being developed to provide dexterous manipulation capability for operations in the Space Shuttle payload bay. The DOSS is a MPESS-mounted robot that can operate from a fixed base or from the end of the Shuttle RMS. The purpose of this 7-DOF manipulator is to provide the Shuttle crew and mission controllers with a tool to augment and potentially replace selected EVA activities in the payload bay. These activities include EVA worksite setup, nominal and contingency payload operations (ie. opening lens covers, removing GAScan lids, etc.), and technology development activities.

Ranger is being integrated at the University of Maryland, under sponsorship of the NASA Telerobotics Program. Scheduled for flight in late 1996 aboard an expendable launch vehicle, Ranger is a dual-arm free flying telerobotics flight experiment which will conduct on-orbit validation and verification of many of the technologies developed by the NASA program. Utilizing telepresence ground-based control, coordinated manipulator operation, automated rendezvous and docking technology, and a hybrid propulsion system, Ranger will conduct a simulated satellite servicing exercise to characterize the operational capabilities of free-flying robotic systems. The project will correlate neutral buoyancy robotic simulations by developing nearly identical underwater and space flight units, and performing identical tasks in both environments.

Japan is also developing a free-flying robotic servicing experiment, scheduled for flight in 1997 aboard a H-II rocket. A target vehicle and chase vehicle will be deployed to exercise technologies including GPS receivers, rendezvous radar, proximity CCD sensors, docking mechanisms and onboard guidance computers. Simultaneously, a 6-DOF manipulator mounted on the chase vehicle will be used to demonstrate cooperative control of the chase vehicle attitude as it reacts to manipulator position, ground-based teleoperation of the manipulator, demonstration of on-orbit satellite servicing including fuel transfer and battery exchange, and target vehicle acquisition, grappling and restraint.

Science Payload Servicing

Science payload servicing robotics differ from the EVR systems in that they are designed to maintain experiment payloads in controlled environments, and are specifically designed as elements of nominal experiment operations (i.e., the robot is intended to be a functional component of the overall experiment, performing tasks such as reagent replenishment, product harvesting, sample collection, etc.), and not just as contingency and repair systems in the event of experiment failure or malfunction.

At least two such systems are currently in the final stages of preflight integration. McDonnell-Douglas has recently completed development of Charlotte. Charlotte is a small robot physically connected to it's work environment with a series of eight Kevlar strands. The strands extend from the corners of the robot's rectangular body to hard points at the extreme corners of the workspace, which may be the interior of SpaceLab, SpaceHab or a space station module. By increasing and releasing tension on selected strands, the body of the robot is able to translate throughout the entire volume of the workspace.

The Robotic Operated Materials Processing Systems (ROMPS) is a joint project between the NASA Goddard Space Flight Center, the Michigan Space Automation and Robotics Center, and the Zymark Corporation. ROMPS will demonstrate low-cost on-orbit processing through the use of robotics to autonomously produce semi-conductor materials. Scheduled for launch on STS-64, this GAScan experiment will investigate zero-gravity annealing of semi-conductor thin films. The robot will utilize low-level automation to maintain the materials furnace, supply source substrates to the furnace and harvest processed thin films.

The European Space Agency is investigating the incorporation of a large-scale science payload maintenance robot into the Columbus module of Space Station. This system would have a work envelope encompassing the entire interior of the module, and would provide logistics support for science experiments and materials production systems.

Planetary Surface Systems

Planetary surface robotics is the area in which the largest breadth of knowledge exists, although it is somewhat dated. As early as 1967, the Surveyor missions carried simple remotely-operated manipulators to the surface of the Moon to collect samples of the Lunar regolith. Followed by the Russian Lunakhods in 1969 and 1980, and the Viking missions to Mars in 1976, these early efforts identified the fundamental environmental constraints and technology obstacles to be surmounted to enable the development of robust, long-lived planetary surface robotics.

It was traditionally accepted that the next generation of robotic rovers for unmanned Lunar and Mars missions would be large (800-kg or more), monolithic, highly intelligent and autonomous devices which would require significant development and operational support in terms of technology, budget, computational and human resources. Then in 1989, a small group of rogue technologists at MIT and JPL began a new initiative in micro-rover technology based on subsumption architectures. Making use of progressively smaller computers, increasingly advanced sensors, and maturing mobility systems, a series of micro-rover testbeds was developed which culminated in the MESUR (Mars Environmental Survey) Pathfinder Rover. This six-wheeled 5 kg-class rover is scheduled for launch to Mars in 1996, and will perform technology validation experiments in addition to science investigations and instrument deployment. Control for the rover will be shared between Earth and the limited onboard intelligence of the rover. By combining sensory input with predefined "behaviors" the rover will autonomously navigate between the waypoints, avoiding rocks, crevasses and other impassable terrain.

Scheduled for flight in 1998, Russia intends to launch the Mars '98 mission which will include the Marsokhod rover. Being developed by the Institute for Space Research (IKI) and the Babakin Center of NPO Lavochkin, the Marsokhod is a six-wheeled, 100-kg rover that will use radioisotopic thermal generators (RTG) for power generation and thermal control. Because of the RTGs, the rover will be able to operate during the Martian night, and is expected to have a long surface lifetime (one year or more) with a potential total excursion distance of over 100 kilometers. The Marsokhod enables exceptional mobility characteristics through the use of unique bi-conic titanium wheels and a segmented three-part chassis.

In addition to these Mars-bound rovers, LunaCorp has announced plans for a lunar rover project slated for launch in 1998. Rather than driven by science needs, the incentive for this project is primarily entertainment - the goal of the project is to provide the world's first interactive space exploration event by giving the public the opportunity to drive the rover on the moon. The rover will be remotely operated via telepresence control from workstations located in theme parks around the country. Capitalizing on NASA rover technology developments, LunaCorp is working with Carnegie-Mellon University to transfer these technologies into the first commercial lunar rover application.

Technology Requirements for Future Systems

With the advent of these new experimental and operational space robotic systems, the ability for remote manipulation to offer significant improvements to mission operations, cost effectiveness and mission safety will be proven. But these will still be early generations of advanced space robotic applications. As successive waves of space robotic applications are deployed beyond the year 2000, the goal of intelligent, autonomous space robotics will become more and more important. Technology drivers for these systems include enhanced collision detection and avoidance, advanced local proximity sensing, task level control workstations, improved command and control architectures, fault tolerant architectures, reduced mass and volume, worksite recognition and representation, improved robotic dexterity, advanced supervisory control, and improved overall system robustness.

By combining these next-generation technologies with the operational knowledge gained from applications being flown in the next few years, the first intelligent space robotic systems will be within reach. By then combining the technologies with the development procedures utilized by the current suite of applications, the next generation of space robotic applications will be affordable, even within the ever-tightening budget environment of today's space program.

SESSION CO

Commercialization

N95- 23674

A New Containerless Image Furnace with Electro-static Positioning Device

Toshihiko Yamawaki, Chihiro Tsukishima*, Toshio Abe, Atsuko Kaneko****

National Space Development Agency of Japan (NASDA)
2-1-1, Sengen, Tsukuba, Ibaraki 305, JAPAN
Tel. Int-298-52-2772, Fax. Int-298-52-1597

*Central Research Laboratory, Mitsubishi Electric Corporation
8-1-1, Tsukaguchihonmachi, Amagasaki, Hyogo 661, JAPAN
Tel.Int-6-497-7125, FAX.Int-6-497-7288, E-Mail Tsukishima@mec.crl.melco.co.jp

**Kamakura Works, Mitsubishi Electric Corporation,
325 Kamimachiya, Kamakura, Kanagawa 247, JAPAN
Tel. Int-467-47-2136, FaX. Int-467-47-1874

ABSTRACT

A new containerless image furnace with a microwave discharge plasma lamp and electrostatic positioning device is developed for the use of the microgravity experiment on the Japanese experimental module (JEM).

The electrostatic positioning system was tested under the reduced gravity environment in the MU-300 aircraft. Solid specimens (maximum weight is 1.3 gr and 10 mm in diameter) and water drops (maximum weight is 0.11 gr and 6 mm in diameter) were successfully controlled under the 0.02G environment.

Rotation control of the dielectric specimen was also possible by means of supplying a rotating electric field while the specimen is levitating. The measured rotation speed of the glass shell specimen (0.08 gr, 10 mm) was up to 110 rpm, when the rotating field frequency was 6 Hz.

INTRODUCTION

A fuzzy reasoning electrostatic positioning system of the containerless image furnace is developed. The electrostatic positioning is first developed at JPL[1] and many types of electrode configurations have been studied. One of the features of the electrostatic positioning is its potential for the low level acceleration control. The acceleration level of the specimen can be easily adjusted depending on the feed back control rules. To isolate the specimen from the vibration of the positioning chamber, a free floating region concept is suggested by JPL[1]. Another feature lies on its capability of handling various materials.

This system is tested under the reduced gravity environment and various specimens are successfully controlled. The free floating concept is also tested adjusting the membership functions used in the fuzzy reasoning.

ELECTRO-STATIC POSITIONING SYSTEM

Outline of the Positioning System

Figure 1 shows the configuration of the positioning system[2]. From the requirement for the configuration with the imaging mirror, a ring type electrode is chosen for our positioning system. The ring type electrodes are used to control the vertical and the radial components of the electric field. The electric potential between the electrodes is derived as follows:

$$\phi = a_1 z + a_2\left(z^2 - \frac{1}{2}r^2\right) + C \tag{1}$$

where the first term is the dipole component, the second term is the quadruple component, and C is the other high order components.

The electric field can be obtained from E=grad ϕ, thus,

$$E_z = -\frac{\partial\phi}{\partial z} = a_1 - 2a_2 z \tag{2}$$

14

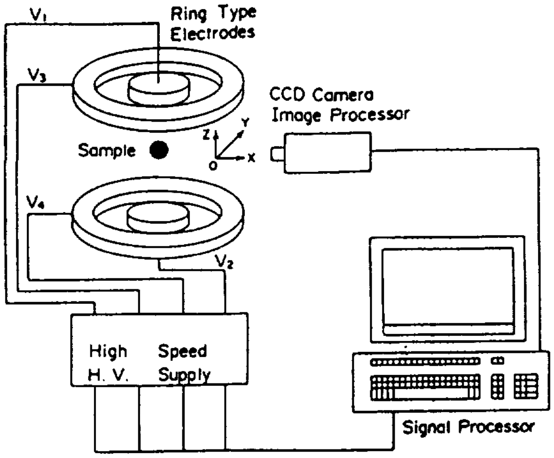


Fig. 1 Configuration of the electrostatic positioning system

$$E_r = -\frac{\partial \phi}{\partial r} = a_2 r \tag{3}$$

$$E_\theta = -\frac{1}{r}\frac{\partial \phi}{\partial \theta} = 0 \tag{4}$$

The motion equation of the charged specimen on the flame of the control system is given by

$$m\frac{d^2x}{dt^2} = qE + F \tag{5}$$

where m is the mass of the particle and F is the external force such as residual gravity.

When the quadruple component (a2) appeared in eq. (2) and (3) is a negative value, a feedback system of the electric field is necessary for the position control in the z direction.

A CCD camera (120 Hz) is set on the horizontal plane to monitor the position of the specimen and a fast high voltage power source with high resolution (1kV/1ms, 12 bit, 4 ch) is used as a voltage supply of the electrodes.

Fuzzy reasoning is performed as the feedback calculation at a fuzzy processor in the control computer. The positioning error and the velocity of the specimen are chosen as the fuzzy inputs and control voltages of 4 electrodes are obtained as the reasoning result. The calculation time of the reasoning is no more than 1 ms (4 inputs, 4 outputs with 17 rules), and is enough shorter than the control cycle which is restricted by the transport rate of the position data from the CCD camera (8.3 ms). Parameters of the positioning system are listed in the table 1.

Table 1. Parameters of the positioning system

Electrode	gap width inner electrode dim. outer electrode dim.	20 mm ~ 40 mm 40 mm 80 mm
Voltage source of positioning	max. output rise time outputs accuracy	10 kV 4 ms 4 ch ~1%
Voltage source of rotation	max. output frequency	3 kV 0 ~ 100 Hz
CCD Camera	frequency accuracy	120 Hz 10 mm/95dot
Control System	sampling Fuzzy reasoning	8.3 ms 4 inputs/4 outputs

Positioning of a Solid Specimen

The positioning system was tested under the reduced gravity environment in MU-300 aircraft. During the 20 sec parabolic flight, the reduced gravity environment which has the amplitude of ~10^{-2}G and frequency of few Hz is obtained.

Figure 2 shows the results of the positioning in the z direction when the specimen is a 1.3 gr

(10 mm in diameter) spherical aluminum. In the figure the membership functions of the position error in the z direction are shown with the positioning results.

In the run–1 the membership functions defining negative and positive small position error (NM, PM) include the center position, although the membership function used in the run–2 does not include the center.

As the result, the specimen in the run–1 is controlled near by the center position, while the specimen in the run–2 is not controlled during the positioning error is small (±2 mm).

In the result of run–2, the freely floating specimen is isolated from the oscillatory disturbances of the aircraft. This free floating time is very short (less than a second) in this experiment because the residual acceleration of the aircraft is still strong in the low frequency region. However, in the spacecraft, the low frequency component of the acceleration is much less than the aircraft (~10^{-6}G), the floating time will become much longer.

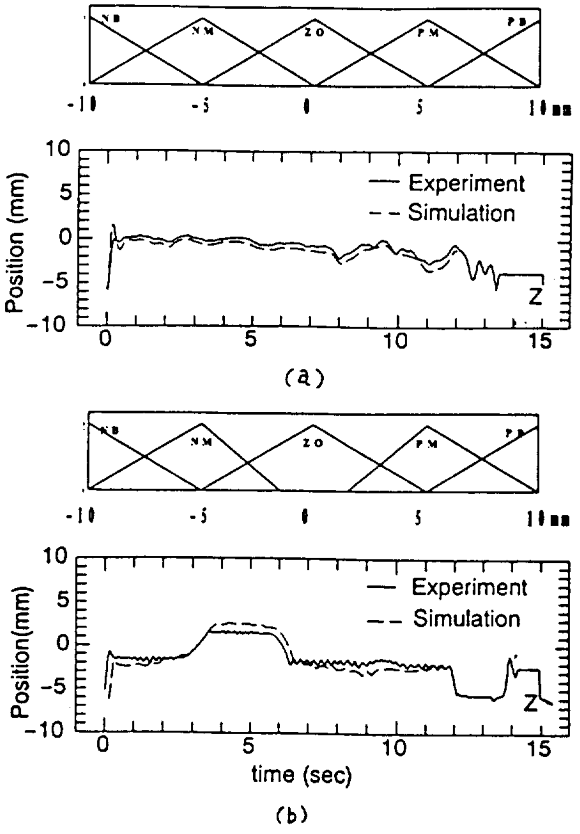


Fig. 2. Experimental results of the positioning in the z direction and used membership functions. (Aluminum 1.3 gr, 10 mm in diameter) (a) run–1, (b) run–2.

Positioning of the Liquid Drop

To obtain the performance of the liquid drop positioning, water drop was tested to levitate. A coaxial nozzle was inserted in the gap of the electrodes through a pin hole of the center electrode to supply the water drop. At first, water drop of 0.11 gr (6 mm in diameter) is made on the top of the inner nozzle, then is departed by means of the air jet come from the outer nozzle. Figure 3 shows the video view of the water drop positioning. The drop kept spherical shape during the levitation and successfully controlled.

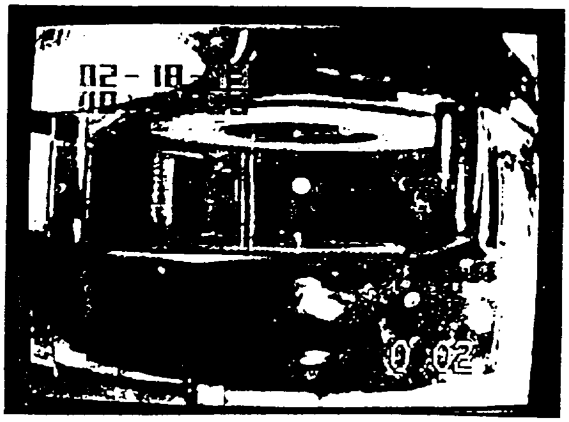


Fig. 3. Video view of the water drop positioning (0.09 gr, 6 mm in diameter)

Rotation Control of the Dielectric Specimen

Rotation control of the dielectric specimen is also possible by means of generating rotating electric field while the specimen is levitating. The ring electrode is divided in four electrodes along the theta direction.

External sine wave voltages of four phases (0 deg, 90 deg, 180 deg, 270 deg) are supplied to four electrodes in addition to the positioning control voltage of the ring electrode. As the induced charge on the surface of the dielectric specimens has the time delay to the rotating electric field, the specimen suffers the torque in the theta direction. Figure 4 shows the dependence of the rotating speed on the external voltage frequency and amplitudes when a grass shell of 0.09 gr (10 mm in diameter) is used as the specimen. The maximum rotating speed of 100 rpm is obtained when the frequency is 6 Hz and voltage amplitude is 3 kV.

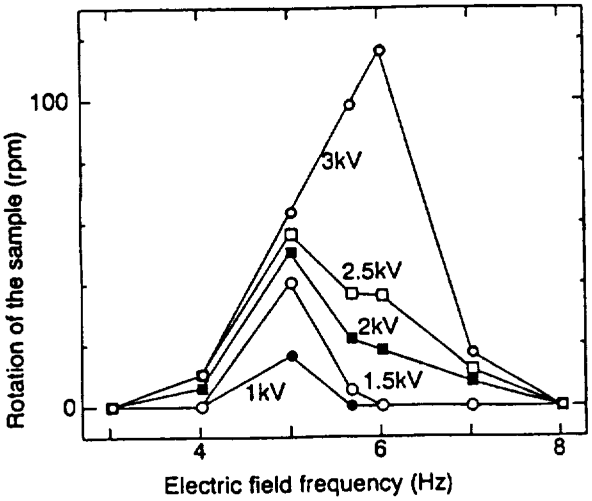


Fig. 4. Dependence of the rotating speed on the external voltage frequency and amplitudes (Glass shell, 0.09 gr, 10 mm in diameter)

Conclusion

A positioning system of the containerless image furnace is developed and tested in the reduced gravity environment. A Solid specimen (1.3 gr, 10 mm aluminum) and a water drop (0.11 gr 6 mm) are successfully position controlled. Rotation control of a dielectric specimens (0.08 gr 10 mm) is also possible while the specimen is levitating. The maximum rotating speed of 110 rpm is obtained when the rotating field frequency is 6 Hz.

REFERENCES

1. W. K. Rhim, M. Collender, M. T. Hyson, W. T. Simms and D. D. Elleman, "Development of an electrostatic positioner for space material processing," Rev. Sci. Inst. No. 56, 1985, pp 307-317.
2. K. Ikegami, C. Tsukishima and S. Okuda, "Control System of Electric Levitator," in the proc. of 17th International Symposium on Space Technology and Science, 1990, pp 2191-2196.

Software Agents for the Dissemination of Remote Terrestrial Sensing Data

Christopher N. Toomey, Evangelos Simoudis,
Raymond W. Johnson, William S. Mark
Lockheed Artificial Intelligence Center
3251 Hanover St. O/96-20 B254F
Palo Alto, CA 94304
(415) 354-5802 (415) 354-5235 (fax)
toomey@aic.lockheed.com

KEY WORDS AND PHRASES

Software agents, artificial intelligence, remote terrestrial sensing data dissemination, knowledge sharing, national information infrastructure.

INTRODUCTION

Remote terrestrial sensing (RTS) data is constantly being collected from a variety of space-based and earth-based sensors. The collected data, and especially "value-added" analyses of the data, is finding growing application for commercial, government, and scientific purposes. The scale of this data collection and analysis is truly enormous; e.g., by 1995, the amount of data available in just one sector, NASA space science, will reach 5 petabytes. Moreover, the amount of data, and the value of analyzing the data, are expected to increase dramatically as new satellites and sensors become available (e.g., NASA's Earth Observing System satellites). Lockheed and other companies are beginning to provide data and analysis commercially.

The Problem

A critical issue for the exploitation of collected data is the dissemination of data and value-added analyses to a diverse and widely distributed customer base. Customers must be able to use their computational environment (eventually the National Information Infrastructure) to obtain timely and complete information, without having to know the details of where the relevant data resides and how it is accessed. Customers must be able to routinely use standard, widely available (and therefore low cost) analyses, while also being able to readily create on demand highly customized analyses to make crucial decisions.

For example, a company laying an oil pipeline would want processed imagery along the pipeline route (or perhaps along several alternative pipeline routes). This imagery would have certain requirements such as image resolution, spectral band, allowable cloud obscuration, and so on. In order to be useful, this imagery usually has to be processed through various analytical techniques, e.g., registration (to precisely align different images along the pipeline route), elevation determination, feature detection, etc. The purchase of such imagery and processing is often a negotiation process: the information the customer wants may either be unavailable or prohibitively expensive. Customers will usually need to reduce costs by refining their orders based on the availability of standard or pre-existing imagery and analysis products. Thus the oil pipeline company would need active feedback during the order formation process in order to determine how some combination of existing and special order products can meet their requirements.

The diversity of user needs creates a difficult software problem: how can users easily state their needs, while the computational environment assumes the responsibility of finding (or creating) relevant information, and then delivering the results in a form that users understand?

Software Agents

A software agent is a self-contained, active software module that contains an explicit representation of its operational knowledge. This explicit representation allows agents to examine their own capabilities in order to modify their goals to meet changing needs and to take advantage of dynamic opportunities. In addition, the explicit representation allows agents to advertise their capabilities and results to other agents, thereby allowing the collection of agents to reuse each others' work.

A large-scale computational environment for data and analysis dissemination is complex and dynamic, and thus it is unrealistic to expect any human or computer program to acquire and maintain functional knowledge of even a fraction of this environment. It is also unrealistic to think that humans or computer programs will have the expertise to determine the content of an arbitrary database or the requirements and results of a new analysis routine. Therefore, agents must rely on the knowledge that other agents have about their (local) environment. The basic knowledge of a database, analysis routine, or set of user requirements is entered by the humans who define the agent in the first place, such as database administrators, algorithm implementors, or end users. This knowledge is accessed by other agents, which use it to augment and modify their own knowledge of the environment. In this way, the total sum of agent knowledge in the environment is cumulative, taking advantage of new knowledge that is constantly being added to the environment in the form of new agents or human modification of existing agents. At the same time, no agent has to have non-local knowledge about the environment: agents rely on what other agents know, augmenting their own knowledge to improve the efficiency of their ability to interact with other agents (remembering short-cuts, reliable partners, etc.).

TECHNICAL APPROACH

Under funding from NASA's technology commercialization program, we are currently building a "showcase" agent-based RTS data dissemination environment to prove the value of this technology in a real world environment. We are working closely with personnel from Lockheed's Space Systems division and Space Imaging Incorporated subsidiary to ground our effort in reality. The key technologies we are using in this effort are:

- Explicit representation of software capabilities and execution events relevant to multimedia access and analysis.
- Knowledge interchange technology to support the sharing of goals and results among agents.
- Reactive planning technology to enable agents to change their behavior in response to changes in the environment.
- User interface technology to facilitate the specification of agent tasks by a variety of end users.

Explicit Representation of Capabilities and Results

There has been considerable recent research activity directed toward the creation of explicit representations of the capabilities and interests of computer tools. Lockheed has participated in this research, primarily in the representation of engineering knowledge and the capabilities and requirements of engineering tools [1]. We are extending this research to the area of data access and exploitation software, which brings some important new features and challenges. For example, databases are usually structured in terms of abstractions that provide a starting point for the explicit representation; but conventional database abstractions leave out much information that must be supplied in the knowledge base.

Knowledge Interchange Technology for Agent Interaction

Government agencies, telephone and other communication companies are developing the network infrastructure that is making efficient large-scale dissemination of data and derivative products cost effective. A key part of this infrastructure is knowledge interchange technology that allows distributed heterogeneous software components to take full advantage of the communication enabled by the new bitways. The knowledge-sharing infrastructure includes a common knowledge representation language, domain ontologies, standard agent/tool interaction protocols, and a facilitation services such as consumer/producer matchmaking [1,2]. Database and analysis tools "plug in" to this infrastructure via wrappers. Wrappers provide an interface that translates between internal tool representations and the shared language and protocols of the infrastructure. Lockheed is a major participant in the creation of the knowledge-sharing infrastructure and wrapper technology. This technology forms the foundation of our agent-based data dissemination environment.

Reactive Planning for Dynamic Behavior Modification

A key tenet of our approach is that agents must be able to examine the capabilities and results of other agents to achieve their goals. In order to actually use this knowledge, agents must act opportunistically, modifying their goals to make use of the partial results and ongoing pursuits of other agents. For example, agents must be able to dynamically reformulate their action plan if they receive a message that another agent has already achieved one of the intended results of their actions. Reactive planning technology enables agents to dynamically change their plans and behavior in response to relevant changes in their environment [3].

User Interface Technology Facilitating Agent Task Specification

We are utilizing advanced user interface technology to ensure that all types of end users will be capable of using our agent-based RTS data dissemination system. Our interface technology hides the complexity of the underlying system by allowing users to interact with the system via high-level, forms-based graphical user interfaces that use standard terminology from the remote sensing domain.

STATUS

We are about halfway through our initial contract with NASA to demonstrate the use of software agent technology in addressing the RTS data dissemination problem.

Progress to Date

To date we have completed a working agent-based prototype for Space Imaging's customer service center (CSC) and representative data sources that it will access. The CSC is the software interface between customers and remote terrestrial sensing products (data and analyses that meet the customer's needs).

Figure 1 illustrates the architecture of our current customer service center prototype. The system demonstrates access to a variety of data sources: archives of images from specific satellites (Landsat, Spot, and Lockheed's own Space Imaging Incorporated (SII) satellite); a database of low resolution preview images, or "chips"; and the SII satellite itself, which can be tasked to produce new images, and thus act as an active data source. Reflective of the real world environment, these data sources are distributed and heterogeneous (implemented using different database management systems and different data representations).

The user interacts with the CSC system via a high-level graphical user interface (GUI). The GUI includes several features to simplify the order specification process. First, it allows the user to specify the desired imagery's geographic region location by drawing it directly on a scalable world map. Second, it allows the user to specify constraints on other image attributes (such as resolution and image acquisition date) via forms-based templates that use generic RTS domain terms and values rather than database-specific ones. Third, the system recommends settings for different attributes based on the application domain selected by the user (e.g., one meter resolution imagery for property assessment applications).

The central element of the system is the Data Broker agent, which serves as the intermediary between the customer and the data sources. The Data Broker receives formal descriptions of the desired imagery characterized by location, resolution, acquisition date, etc. from the GUI. It is responsible for matching data requests to a set of specific data sources capable of providing such data. The Data Broker is aware of the capabilities and input requirements of data sources because they have been advertised. Data sources come on line when their wrapper agents advertise their capabilities to the other agents in the environment, including the Data Broker. The Data Broker is thus able to transform incoming data requests into "targeted data requests" based on the known capabilities and requirements of all available data sources.

Wrapper agents for the individual data sources receive these targeted data requests, and are responsible for returning metadata for each of the images they have meeting the user's requirements. To do so, a wrapper translates the request from the common interagent language into the wrapped data source's query language, queries the data source, and translates the results back into the interagent language.

Lastly, the Data Broker is responsible for collecting and pruning the wrappers' results in order to create a coherent composite result set. Pruning is necessary when the different data sources provide overlapping results. It can be a task of considerable sophistication, since it can require making tradeoffs on different data characteristics (which is better, less cloud obscuration or higher resolution?). Currently, the Data Broker supports only a single pruning option: to remove older images in the result set that are completely overlapped by newer ones.

Future Work

The CSC prototype shown in Figure 1 is implemented and is end-to-end operational. However, only the Spot and SII Archives data sources have been wrapped and put on-line to date. In the remainder of this year we will be wrapping the other data sources, including the SII satellite tasking module, which will require reactivity to collection scheduling changes induced by bad weather, crisis tasking requests, and order tasking conflicts. We will also be illustrating result sharing among agents, such as between multiple Data Broker agents, and the use and management of a dynamic collection of persistent agents representing customer orders [4].

REFERENCES

[1] McGuire, J. G. et. al., 1993. SHADE: Technology for Knowledge-Based Collaborative Engineering. *Concurrent Engineering: Research and Applications* 1(3).

[2] Simoudis, E., 1992. Integrating Distributed Expertise. *International Journal of Intelligent and Cooperative Information Systems* 1(3-4).

[3] Kaelbling, L.P.; and Rosenschein, S.J., 1990. Action and Planning in Embedded Agents. *Robotics and Autonomous Systems* 6(1).

[4] Toomey, C.N.; and Johnson, R.W., 1994. Software Agents for Automating Multiple-Tool Tasks. In *Working Notes of the 1994 AAAI Spring Symposium on Software Agents.* Palo Alto, Calif.: American Assoc. for Artificial Intelligence.

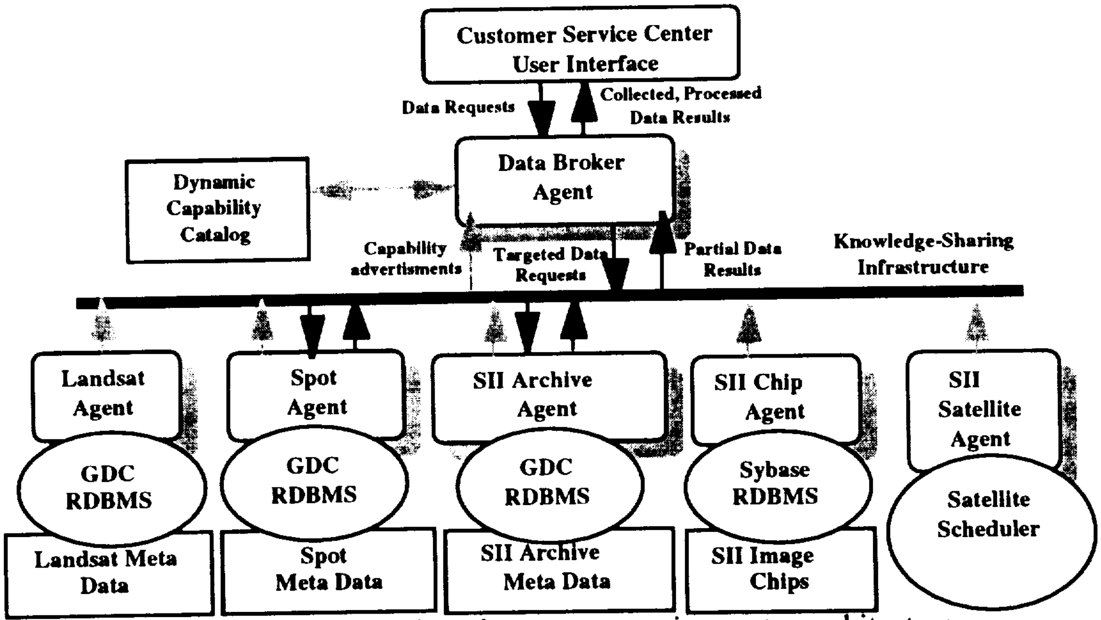


Figure 1. The agent-based customer service center architecture.

N95- 23676

Commercialization of JPL Virtual Reality Calibration and Redundant Manipulator Control Technologies

Won S. Kim
Homayoun Seraji
Paolo Fiorini
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
(818)354-5047
fax: (818)393-5007
wonsoo@telerobotics.jpl.nasa.gov

Robert Brown
Brian Christensen
Chris Beale
Deneb Robotics, Inc.
3285 Lapeer Road West
P. O. Box 214687
Auburn Hills, MI 48321-4687
(810)377-6900
fax: (810) 377-8125
bob@deneb.com

James Karlen
Paul Eismann
Robotics Research Corp.
P.O. Box 206
Amelia, OH 45102
(513)831-9570
fax: (513)381-5802

KEY WORDS AND PHRASES

Robotics, virtual reality, calibration, redundant robot arm control.

INTRODUCTION

Within NASA's recent thrust for industrial collaboration, JPL (Jet Propulsion Laboratory) has recently established two technology cooperation agreements in the robotics area: one on virtual reality (VR) calibration with Deneb Robotics, Inc., and the other on redundant manipulator control with Robotics Research Corporation (RRC). These technology transfer cooperation tasks will enable both Deneb and RRC to commercialize enhanced versions of their products that will greatly benefit both space and terrestrial telerobotic applications.

COMMERCIALIZATION OF JPL VIRTUAL REALITY CALIBRATION TECHNOLOGY

JPL recently developed a virtual reality (VR) calibration technique that enables reliable and accurate matching of a graphically simulated virtual environment in 3-D geometry and perspective with actual video camera views [1], [2]. This technique enables high-fidelity preview/predictive displays with calibrated graphic overlay on live video for telerobotic servicing applications. Its effectiveness was successfully demonstrated in a recent JPL (Jet Propulsion Laboratory)/NASA-GSFC (Goddard Space Flight Center) ORU (Orbital Replacement Unit) changeout remote servicing task. The current JPL VR calibration is a two-step procedure: camera calibration followed by object localization. Key new features of this JPL VR calibration technique include: 1) an operator-interactive method adopted to obtain reliable correspondence data, 2) a robot arm itself used as a calibration fixture for camera calibration, eliminating a cumbersome procedure of using external calibration fixtures, 3) the object localization procedure added after the camera calibration to obtain graphic overlay of both the robot arm and the object(s) on live video enabling effective use of the computer-generated trajectory mode in addition to the teleoperation mode, 4) a projection-based linear least-squares algorithm extended to handle multiple camera views for object localization, and 5) nonlinear least-squares algorithms combined with linear ones employed for both camera calibration and object localization. Details of the algorithms and their software listings [3] were prepared as part of this JPL-Industry cooperative task.

An example of a calibrated graphic overlay after the virtual reality calibration for the JPL/NASA-GSFC remote servicing demonstration is shown in Figure 1. The positioning alignment accuracy achieved in inserting a tool into the ORU hole using 4 camera views was 0.51 *cm* on the average with a 1.07 *cm* maximum error at 95% confidence level. After matching 3-D graphics models of a virtual environment with actual camera views through the above virtual reality calibration technique, the operator can now perform a telerobotic servicing task with preview/predictive displays having calibrated graphics overlay on live video. Preview/predictive displays allow the operator to generate the simulated robot arm trajectory in preview and then to visually monitor and verify the actual remote robot arm motion with confidence, and thus provide

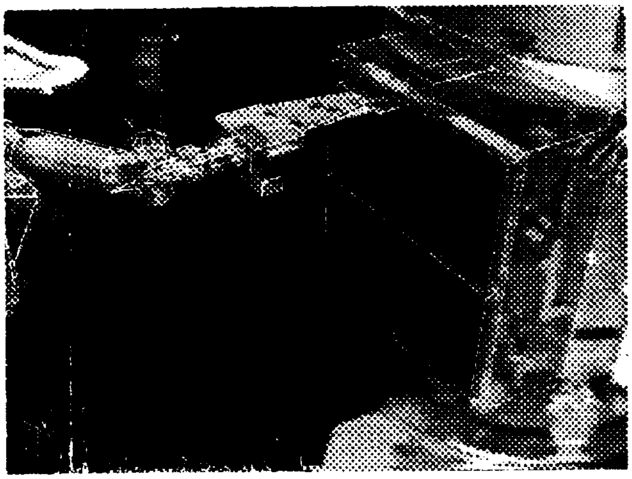


Figure 1: Overlay of calibrated 3-D graphic models (wire-frames with semi-transparent surfaces) on live video for telerobotic satellite servicing.

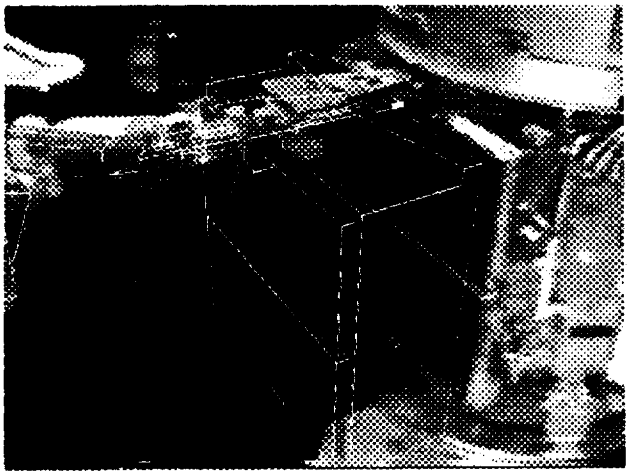


Figure 2: A snapshot of a preview/predictive display during the performance of the ORU extraction in the JPL/GSFC ORU changeout demonstration task.

effective visual prediction/verification to the operator and enhance safety and reliability in remote servicing operations regardless of communication time delay. Figure 2 shows a snapshot of a preview/predictive display during the performance of the JPL/GSFC demonstration.

Approach

We have taken the following approach in our JPL-Industry cooperative Deneb Commercialization Task: 1) JPL transfers the VR calibration software technology to Deneb, 2) Deneb, cooperating with JPL, inserts this software technology into its commercial product TELEGRIP as the video overlay/VR calibration option for marketing, and 3) in return, NASA utilizes this enhancement of a commercially supported product for NASA applications.

The virtual reality calibration option implemented on TELEGRIP will be an important element to build a state-of-the art VR interface in telerobotic applications with preview/predictive displays. Thus, the enhanced Deneb product can be effectively used in both space and terrestrial telerobotics applications, providing 1) immediate benefits to NASA for ground-controlled telerobotic servicing in space, 2) immediate benefits to the national DOE (Department of Energy) labs working on the disposal and remediation of nuclear waste, and 3) foreseeable potential applications in automotive manufacturing, medical telerobotic surgery, telerobotic construction, and maintenance robots.

Implementation on TELEGRIP

The JPL virtual reality calibration option is currently being implemented on Deneb's TELEGRIP [4] which is an open architecture based upon Dynamic Shared Objects (DSO's). DSO's provide many benefits when compared with other strategies for incorporating user-defined modules with a centralized kernel, including 1) speed of development, 2) access to all internal functions and data, including the entire geometric database, 3) flexibility in development, and 4) minimizing platform dependence. A key important feature provided by this TELEGRIP open architecture is that it allows developers/users to add their own virtual reality calibration algorithms and video overlay methods, if necessary.

Both one-window and two-window graphics/video displays are planned to be supported for VR calibration. Under the one-window calibration strategy, the TELEGRIP graphics display is divided into two separate vertically arranged NTSC-size (National Television Systems Committee standard) viewports. One viewport contains the live video image of the work environment, while the other displays the equivalent 3D graphical model. Upon completion of the camera calibration and object localization phases, the graphics-overlaid video image will be available to display in one of the viewports or to display on a separate NTSC monitor. The two-window approach relies upon two external NTSC-size GL (Graphics Library) or GLX (Graphics Library in X environments) windows with one window containing the live video image and the other the 3D graphic display. This enables users to relocate the windows in a manner desirable for their particular application. Upon completion of the camera calibration and object localization phases, a graphics-overlaid video image is available to display in any window, including the TELEGRIP window, or to display on a separate NTSC screen.

The TELEGRIP video overlay implementation is based upon an application programmer interface (API) layer which insulates the overlay developer from the specifics of video hardware, thus enabling support over a wide range of video products. Support is currently planned for the SGI (Silicon Graphics, Inc.) VideoLab, Galileo, Indigo2, Indy, and Serius video boards encompassing the entire range of current SGI computing hardware from the Indy to the Onyx. Graphic models can be overlaid in wire-frame or in solid-shaded polygonal rendering, with varying levels of transparency to produce different visual effects.

COMMERCIALIZATION OF JPL REDUNDANT MANIPULATOR CONTROL TECHNOLOGY

Theoretical and experimental investigations have demonstrated that dexterous manipulation tasks can be carried out only by redundant, force-controlled robotic manipulators that possess flexibility and versatility comparable to the human arm. For research in this area, the Robotics Laboratory at JPL acquired in 1989 two redundant 7-DOF (degree-of-freedom) manipulators made by Robotics Research Corporation (RRC) of Ohio, the leading manufacturer of this type of manipulators since the mid 1980's.

At the time of purchase, neither the application domain nor the required redundant control laws for such advanced manipulators was fully developed. JPL research has contributed to both areas by identifying tasks in which redundancy is essential and by developing an underlying control methodology for such manipulators.

RRC has recently expanded and enhanced its product line by introducing a second-generation version of its manipulator that provides improved mechanical performance and employs a unique low-level control system in which all servo electronics are mounted in the arm. It is now logical to begin integrating RRC's state-of-the-art servomechanism technology with JPL's advanced high-level control developments, and to prepare this new robot technology for commercial applications.

Under funding from NASA, the first phase of such a commercialization activity began in FY'94, with the transfer to RRC of an algorithm for redundant arm control developed at JPL[5-9] and widely used in the robotics community. This algorithm, known as Configuration Control, combines the specification of a set of constraint tasks with the end-effector prescribed trajectory to provide a highly efficient and powerful redundant arm control strategy.

Background

During the course of the past two years, RRC has developed a unique servo control architecture for its manipulator arms which greatly reduces the need for expensive external power and computing electronics and replaces the costly internal arm wiring harness with a "fly-by-wire" data/power bus communication system. Miniature DSP (Digital Signal Processor)-based servo control modules, containing all computing and power electronics, are collocated with the joint actuators in the manipulator arm joints. The parameters for the individual joint controllers are downloaded by a master computer via a high-speed communication link. Since the remotely-located master computer is free from the burden of servo power and computing electronics, high-level control functions can now be practically transferred to a general-purpose workstation or personal computer with significant cost savings. This new high-level RRC controller is designated the Next Generation Controller (RRC/NGC).

In the area of redundant arm control, JPL has developed a class of motion control algorithms for redundant manipulators called Configuration Control (CC),[5-9]. In this approach, the user can specify task-dependent constraints for the redundant manipulator which have the effect of utilizing the robot redundancy and allowing efficient end-effector trajectory control. Since this approach was implemented originally on RRC manipulators and the resulting algorithms were extensively tested in several experiments, it is felt that this technology is mature enough to be transferred to industry and incorporated into RRC's new product line (see Figure 3).

The RRC/NGC system under development will be highly compatible with the kind of centralized high-level control embedded in the CC approach. The master computer used in the NGC system is a standard workstation, and it is well suited to run the CC algorithms. Furthermore the use of a workstation (or of a PC) as a master computer enables RRC to make use of enhanced graphic capabilities to provide the user with a sophisticated interface for motion planning and control.

Approach

In order to ensure that the technology transfer proceeds smoothly, the following steps have been planned:

1. Duplicate the hardware and software environment of the RRC/NGC at JPL and test it with

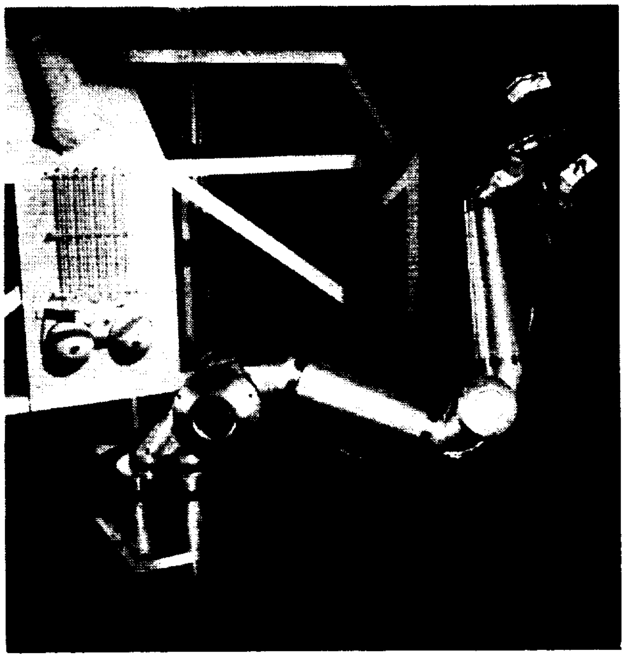


Figure 3: 7-DOF Robotics Research arm.

the RRC manipulators in the JPL Robotics Laboratory.

2. Modify JPL Configuration Control algorithms to make them compatible with the NGC environment, implement and test the algorithms on the master computer adopted in the NGC system and with the current RRC manipulators in the JPL Robotics Laboratory.

3. Integrate the tested algorithms with the new RRC manipulators using the Next Generation Controller.

Technology Transfer Issues

A technology transfer task of this type requires the same steps as to transform a laboratory prototype into a commercial product. Once the functionality of the prototype, the CC algorithms in this case, has been established and verified, then the development efforts must focus on issues such as compatibility with the rest of the system, price/performance trade-off, documentation, maintainability, and so on.

The decision was made by RRC to implement as much as possible of their software in object-oriented format, and use an IBM-compatible personal computer as the master controller. From the JPL side, it was necessary to re-engineer some existing software to eliminate the dependency of the code on data structures related to the rest of the JPL system, and to port the programs to an operating system compatible with the IBM-PC that RRC has selected as its NGC platform. In the interest of compatibility with existing RRC software, as well as to minimize overall system cost, the real-time operating system selected is the Intel iRMX running under Windows, which can execute RRC's existing code as well as the new JPL Configuration Control software modules.

The technology transfer is currently proceeding smoothly and most of the necessary programs have already been converted to a stand-alone configuration. We will be ready to integrate this software with the PC-based real-time system and test it with the RRC redundant manipulators in the JPL Robotics Laboratory later this year.

ACKNOWLEDGMENT

This work was performed partly at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Kim, W. S., Schenker, P. S., Bejczy, A. K., Leake, S., and Ollendorf, S., 1993. "An Advanced Operator Interface Design with Preview/Predictive Displays for Ground-Controlled Space Telerobotic Servicing," SPIE Conference 2057: *Telemanipulator Technology and Space Telerobotics*, 96-107, Boston, MA.

[2] Kim, W. S. and A. K. Bejczy, 1993. "Demonstration of a High-Fidelity Predictive/Preview Display Technique for Telerobotic Servicing in Space," *IEEE Trans. on Robotics and Automation*, 9(5), 698-702.

[3] Kim, W. S., 1994. *Virtual Reality Calibration: Algorithms and Software Listings with an application to Preview/Predictive Displays for Telerobotic Servicing*, JPL D-11593 (Internal Document), Jet Propulsion Laboratory, California Institute of Technology.

[4] Deneb Robotics, Inc., 1994. *TELEGRIP Access Reference Manual.*

[5] Seraji, H., 1989. "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Transactions on Robotics and Automation*, 5(4), 472-490.

[6] Seraji, H. and R. Colbaugh, 1990. "Improved Configuration Control for Redundant Robots", *Journal of Robotic Systems*, 7(6), 897-928.

[7] Colbaugh, R., H. Seraji, and K. Glass, 1989. "Obstacle Avoidance for Redundant Robots Using Configuration Control", *Journal of Robotic Systems*, 6(6), 721-744.

[8] Seraji, H., 1992. "Task-Based Configuration Control of Redundant Robots", *Journal of Robotic Systems*, 9(3), 411-451.

[9] Seraji, H.,M. Long, and T. Lee, 1993. "Motion Control of 7-DOF Arms: The Configuration Control Approach", *IEEE Transactions on Robotics and Automation*, 9(2), 125-139.

SESSION EA

Engineering Assistant Systems

Utilizing Inheritance in Requirements Engineering

N95-23677

Hermann Kaindl
Siemens AG Österreich, PSE
Geusaugasse 17, A - 1030 Vienna, Austria
Tel: +43-1-71600-288 Fax: +43-1-71600-323
E-Mail: kaih@siemens.co.at

KEY WORDS AND PHRASES

Hypertext, inheritance, requirements engineering, semiformal representation.

INTRODUCTION

Specifying the requirements of a new system to be built is one of the most important parts of the life cycle of any project. In the field called *requirements engineering* many approaches have been proposed [1]. However, few methods and tools have been available for practical use. In fact, for the early phase of defining the requirements, nearly no support is available.

While from a theoretical point of view it would be desirable to have *formal* representations of requirements, in practice unstructured natural language is often used *informally*. Our approach attempts to bridge the gap between these extremes in providing *semiformal hypertext* representations. Therefore, our approach and the tool supporting it are named RETH (Requirements Engineering Through Hypertext). Actually, RETH uses a combination of various technologies, including *object-oriented approaches* and *artificial intelligence* (in particular *frames*). We do not attempt to exclude or replace formal representations, but try to complement and to provide means for gradually developing them.

The scope of this paper is the utilization of *inheritance* for *requirements specification*, i.e., the tasks of analyzing and modeling the domain, as well as forming and defining requirements.

Among others, RETH has been applied in the CERN (Conseil Européen pour la Rechereche Nucléaire) *Cortex* project. While it would be impossible to explain this project in detail here, it should be sufficient to know that it deals with a generic distributed control system. Since this project is not finished yet, it is difficult to state its size precisely. In order to give an idea, its final goal is to substitute the many existing similar control systems at CERN by this generic approach. Currently, RETH is also tested using real-world requirements for the *Pastel Mission Planning System* at ESOC in Darmstadt.

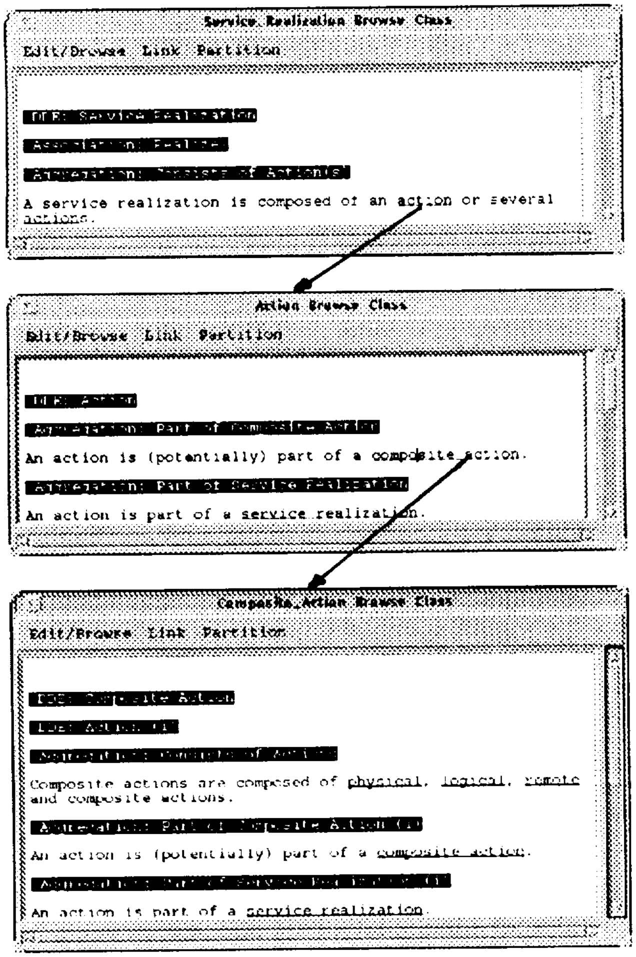
First, we outline how hypertext is integrated into a frame system in our approach. Moreover, we demonstrate the usefulness of inheritance as performed by the tool RETH. We then summarize our experiences of utilizing inheritance in the Cortex project. Lastly, we relate RETH to existing work.

HYPERTEXT INTEGRATED INTO A FRAME SYSTEM

A hypertext node is represented as a frame in our approach. (The original notion of a *frame* was coined by Minsky [2], but the frame systems implemented the original ideas only partially. In the context of this paper, a frame can be viewed as a data structure that combines data stored in *slots*.) According to the differences between object-oriented languages and frame systems as discussed in [3, 4], we selected the frame system of PROKAPPA as the basis of our tool RETH.

Our approach of integrating hypertext into a frame system is similar to the one described and used by Kaindl and Snaprud [5, 6] for *knowledge acquisition* in the course of building knowledge-based (expert) systems. One distinctive feature lets the user define disjoint *partitions* of nodes that together cover the whole node. Such a partition of a hypertext node is comparable to a slot of a frame. The idea is to support the user in partitioning the textual content in a machine recognizable form, serving as an additional means of introducing more formality.

In order to make the example below understandable, we shortly sketch the hypertext user



A service realization is composed of an action or several actions.

An action is (potentially) part of a composite action.

An action is part of a service realization.

Composite actions are composed of physical, logical, remote and composite actions.

An action is (potentially) part of a composite action.

An action is part of a service realization.

Figure 1: RETH windows showing object representation and inheritance.

interface of RETH (see Fig. 1 showing a screen dump). The presentation level handles hypertext links as follows: if the underlined string representing the link is clicked with the mouse, the window of the target node is displayed by the tool. The arrows in the figure are drawn to indicate the effect of following links on the screen. The windows shown in the figure actually poped up one at a time.

In contrast, the display of *partitions* of hypertext nodes is implemented in our tool like *expand buttons* (cf. the hypertext system *Guide* [7]). When the name of a partition (inverted in the display) is clicked, the content is expanded or shrunk (implemented as a toggle). E.g., in the window at the top of Fig. 1 the partition DDE: Service Realization is currently shrunk, while Aggregation: Consists of Action(s) is expanded. In contrast to many hypertext systems, our approach lets users mix browsing and editing of nodes, though one node can either be edited or browsed at one point in time.

INHERITANCE IN REQUIREMENTS SPECIFICATION

Due to lack of space, we cannot describe here the details of using RETH for domain analysis and modeling, and for the formation and definition of requirements. The key ideas are to represent requirements as objects, and to organize these objects as well as the objects of the domain model in a taxonomy. Within this taxonomy, inheritance can be used in several ways (see below). Hypertext links are used to interlink the hypertext nodes representing the objects. For a detailed description, the interested reader is referred to [8].

Due to our tight integration of hypertext in a frame system, inheritance can be used already in the semiformal representation. There is a notable difference between frame systems and object-oriented languages relevant for our approach: in contrast to the latter, the former also support inheritance of *values* [3, 4]. Since classes (of the domain model as well as of requirements) are described in hypertext nodes, and since these are represented as frames, the text contained in them is inherited.

Together with the concept of partitions of nodes, inheritance supports *templates*, e.g., for requirements to be filled in. Whenever a node for a requirement is created as an instance of a class of requirements, the appropriate *structure* is already given initially through inheriting a template. Inherited partitions in the (requirements) instances provide for the representation of information on requirements such as their source, reason and priority.

Detailed information about requirements is especially important for large projects, but without sufficient tool support it is often omitted. Since all the instances inherit all the respective partitions, providing such information cannot be forgotten, and the user of the system just has to fill in the text.

When requirements are organized in *classes*, all the requirements of a specific class can have a special attribute in common — represented as a partition. Moreover, whole classes of requirements (defined by the user) can have the same value (text) of an attribute, and this value can be defined *once* in the description of the class. The subclasses and instances inherit this value, but inherited information can also be overridden.

An important point is that inheritance allows one to define special attributes (including a value or not) *once* in the definition of the class, without the necessity to copy. Even more important is the possibility of re-inheriting changed values.

In contrast to most current OOA tools, RETH implements OOA inheritance already in the semiformal hypertext representation (see also Fig. 1).

EXPERIENCE WITH RETH IN THE CORTEX PROJECT

According to our experience in the real-world project Cortex, all the features of our method and its supporting tool were useful to some extent. In fact, some of them were worked out in detail in the course of this application. Due to lack of space, we will only focus here on the utilization of inheritance.

The templates of requirements depending on their class helped to point out missing information. Actually, much of it was known by the people involved, but we found it important to get it written down.

Moreover, we would like to point out specifically the usefulness of domain-specific requirements *classes*, and the use of *inheritance* within the corresponding taxonomy. They allowed the explicit ordering of the requirements according to the classification principle. While this is of course not a new principle for ordering requirements, our approach and the tool provide inheritance. Therefore, it was possible and very useful to specify information such as priorities *once* for whole classes. When the priority of a class of requirements changes, it is only necessary to specify this once — in the corresponding partition of the node representing this class. The nodes representing requirements subclasses and instances of this class re-inherit this changed value.

Another interesting example of the use of inheritance that we came across during the work on Cortex is illustrated in Figs. 1 and 2 (in the notation of [9]). An Action is *part of* a Service_Realization. Since a Composite_Action, e.g., *is an* Action, it is also *part of* a Service_Realization. This inference has to be drawn by the viewer of the O-O diagram but is made automatically via inheritance in RETH. In the bottom window of Fig. 1, the inherited partition Aggregation: Part of Service Realization (i) shows this. Moreover, inheritance points to the fact that a Composite_Action is (potentially) also *part of* a Composite_Action (see the inherited partition Aggregation: Part of Composite Action (i) in the bottom window of Fig. 1). Especially this kind of inference may be difficult for people not so familiar with recursive structures in O-O diagrams. Of course, the diagram has its advantages, too. Therefore, both forms of representation are complementary in our view.

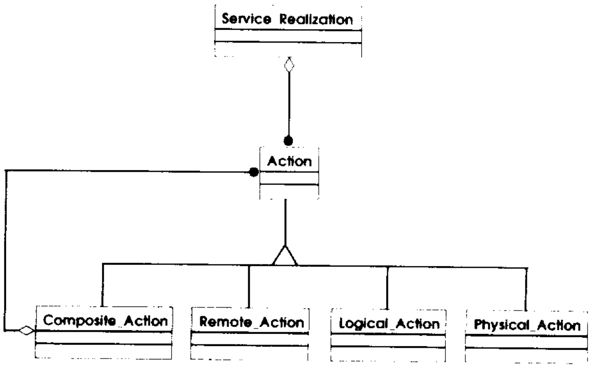


Figure 2: An object model diagram.

RELATED WORK

Due to lack of space we cannot give here a comprehensive overview of all the proposed approaches to requirements engineering. Especially for the traditional ones, the interested reader is referred to [1]. Recent OOA approaches (for an overview see [10]) challenge the traditional ones. RETH heavily builds on object-oriented ideas. However, most of today's OOA methods still ignore early development phases where important clarifications have to be made. It may even be argued that they are designed for a different phase. RETH specifically focuses on the early phase, and we propose to combine RETH with object-oriented analysis approaches.

The method by Jacobson *et al.* [11] and the tool supporting it (*Objectory*) bear some similarity to our approach. However, it does not apply object-oriented principles to the organization of the requirements, and consequently inheritance cannot be utilized (e.g., for templates).

Since RETH's internal representation is based on frames, it may be interesting to compare it with other approaches to requirements engineering using *artificial intelligence* (AI) technology. GIST [12] is an important early approach. RML [13] emphasizes the use of *knowledge representation* techniques of AI and domain modeling. Telos [14] is a derivative of RML. RA [15] shares with RETH the focus on a transition between informal and formal representations. The approach of ARIES [16] is quite similar to RA in being very knowledge-intensive. KBRA [17] utilizes hypertext ideas internally. While RETH's user interface for structuring text appears to be more developed, some of KBRA's features of machine support could be very useful in RETH. However, KBRA lacks several important features of RETH.

CONCLUSION

In summary, our tool-supported method named RETH supports several activities in the course of requirements specification. Our approach of organizing the hypertext according to *object-oriented* principles has several advantages. Representing requirements as objects helps when structuring them via *classification*. *Inheritance* is provided by our tool already in the early phase of requirements specification, which helps to avoid redundant representation of information. In particular, it provides users automatically with templates of the internal structure of requirements, that depends on the kind of requirement. This way, the users are guided to fill in important information like the reason and priority of each requirement. While RETH is not intended to substitute useful existing techniques emphasizing more formal representations, it can be combined with them.

Since the advantages of such an approach to requirements engineering cannot be fully utilized without more elaborate *traceability* of the requirements, we also investigate how to best link requirements objects with *design* objects.

The usefulness of RETH to space projects is currently assessed using real-world requirements for the *Pastel Mission Planning System* at ESOC in Darmstadt. While it is too early for a final statement at the time of this writing, the preliminary results are encouraging. Since RETH is very general in terms of application areas, we could not find any reason why the application to space projects should be a problem.

ACKNOWLEDGMENTS

Stefan Kramer and Stefan Korner did very important work in building the tool. Peter Tippold gave useful comments on earlier drafts of this paper, and his willingness for in-depth discussion of practical issues in requirements engineering is highly appreciated. Also Holger Ziegeler participated in earlier discussions. Moreover, the partial funding of this work by ESA and the ITF (Innovation- und Technologiefonds) is acknowledged. Finally, we would like to thank CERN for giving us the opportunity to apply our new approach in one of their real-world projects.

References

[1] A. M. Davis. *Software Requirements: Objects, Functions, and States.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, New York, 1975.

[3] H. Kaindl. Object-oriented approaches in software engineering and artificial intelligence. *Journal of Object-Oriented Programming*, 6(8):38–45, January 1994.

[4] H. Kaindl and S. Korner. Object-oriented languages supplemented with features of frame systems. In *Proceedings of the OOPSLA '92 Workshop on Object-Oriented Languages: The Next Generation*, Vancouver, Canada, October 1992.

[5] M. Snaprud and H. Kaindl. Knowledge acquisition using hypertext. *Expert Systems with Applications: An International Journal*, 5(3/4):369–375, 1992.

[6] M. Snaprud and H. Kaindl. Types and inheritance in hypertext. *International Journal of Human-Computer Studies (IJHCS)*, 1994. To appear.

[7] P. J. Brown. Do we need maps to navigate round hypertext documents? *Electronic Publishing—Origination, Dissemination, and Design*, 2(2):91–100, July 1989.

[8] H. Kaindl. The missing link in requirements engineering. *ACM Software Engineering Notes (SEN)*, 18(2):30–39, 1993.

[9] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design.* Prentice Hall, Englewood Cliffs, NJ, 1991.

[10] D. de Champeaux and P. Faure. A comparative study of object-oriented analysis methods. *Journal of Object-Oriented Programming*, pages 21–33, March/April 1992.

[11] I. Jacobson, M. Christerson, P. Jonsson, and G. Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley, Reading, MA, 1992.

[12] R. Balzer. Final report on GIST. Technical report, USC/ISI, Marina del Rey, CA, 1981.

[13] A. Borgida, S. Greenspan, and J. Mylopoulos. Knowledge representation as the basis for requirements specification. *IEEE Computer*, 18(4):82–91, 1985.

[14] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: Representing knowledge about information systems. *ACM Transactions on Information Systems*, 8(4):325–362, 1990.

[15] H. B. Reubenstein and R. C. Waters. The requirements apprentice: automated assistance for requirements acquisition. *IEEE Transactions on Software Engineering*, 17(3):226–240, 1991.

[16] W. L. Johnson, M. S. Feather, and D. R. Harris. Integrating domain knowledge, requirements, and specifications. *Journal of Systems Integration*, 1:283–320, 1991.

[17] A. J. Czuchry and D. R. Harris. KBRA: a new paradigm for requirements engineering. *IEEE Expert*, pages 21–35, Winter 1988.

---

[0] PROKAPPA is a trademark of IntelliCorp, Inc.

Using Dedal to Share and Reuse Distributed Engineering Design Information

N95- 23678

Vinod Baya*, Catherine Baudin, Ade Mabogunje*, Aseem Das****
David M.Cannon*, Larry J. Leifer*
(*) Center for Design Research
Stanford University, Stanford, CA 94305
Tel: (415)-725-0160 Fax: (415)-725-8475
<baya,ade,dmcannon,leifer>@sunrise.stanford.edu

(**) RECOM Software Inc.
Artificial Intelligence Research Branch
NASA Ames Research Center, Moffett Field, CA 94305
<baudin,aseem>@ptolemy.arc.nasa.gov

KEYWORDS

Indexing, retrieval, distributed design, World Wide Web.

INTRODUCTION

The overall goal of the project is to facilitate the reuse of previous design experience for the maintenance, repair and redesign of artifacts in the electromechanical engineering domain.

An engineering team creates information in the form of meeting summaries, project memos, progress reports, engineering notes, spreadsheet calculations and CAD drawings. Design information captured in these media is difficult to reuse because the way design concepts are referred to evolve over the life of a project and because decisions, requirements and structure are interrelated but rarely explicitly linked. Based on protocol analysis of the information seeking behavior of designer's, we defined a language to describe the *content* and the *form* of design records and implemented this language in Dedal, a tool for indexing, modeling and retrieving design information [1].

We first describe the approach to indexing and retrieval in Dedal. Next we describe ongoing work in extending Dedal's capabilities to a distributed environment by integrating it with World Wide Web. This will enable members of a design team who are not co-located to share and reuse information.

BACKGROUND: INDEXING AND RETRIEVAL IN DEDAL

Dedal is a tool to help designers index, model and reuse design information. It uses an *conceptual indexing language* [3] which combines concepts from a model of the designed artifact with a vocabulary representing generic task-dependent classes of information covered by design documents such as *function, operation, alternatives.*

Design information is indexed by a set of conceptual indexing patterns. A conceptual index can be seen as a structured entity consisting of two parts: the body of the index which represents the *content* of a piece of information, and the reference part that points to a region in a document. For instance: "The inner hub holds the steel friction disks and causes them to rotate" is part of a paragraph on page 20 in the record: report-333. It can be described by two indexing patterns:

> <topic *FUNCTION* subject *INNER-HUB* level-of-detail *CONFIGURATION* medium *TEXT* in-record *REPORT-333* segment *20*>.

> <topic *RELATION* subject *INNER-HUB* and *STEEL-FRICTION-DISKS* level-of-detail *CONFIGURATION* medium *TEXT* in-record *REPORT-333* segment *20*>

The queries have the same structure as the body of an index and use the same vocabulary. A question such as: "*How does the inner hub interact with the friction disks?*" can be formulated in DEDAL as:

<get-information-about topic *RELATION* of subject *INNER-HUB* and *FRICTION-DISKS*>

An indexing fragment can refer to a segment of information of different size: a paragraph, a page, a section, a chapter or a document [1]. In addition, the indexer can define *relations* among the design concepts. This enables the system to explore relations among decisions, requirements and alternatives to extend the query when a retrieval fails.

The retrieval module takes a query from the user as input, matches it to the set of conceptual indices and returns an ordered list of answers related to the question. The retrieval proceeds in two steps. The first step is to find indices which match the query exactly. If no exact matches are found then the relations in the indexing model are used to reformulate the query and step one is repeated. The retrieval procedure and a set of retrieval heuristics are described in [1]. Following is an example of retrieval in Dedal.

Designer's question is: *Why is the maximum force in this damper design 500 lbs?*

Query to Dedal: topic: *RATIONALE* for the subject: *MAX-FORCE of DAMPER*

Dedal first tries to find an indexing pattern: <topic: *RATIONALE*, subjects: *MAX-FORCE of DAMPER*> in any media and level of detail. If no indices are found, retrieval heuristics are activated. It looks for requirements associated with quantities that influence the *MAX-FORCE of DAMPER*. In this case, the indexing model indicates that the force of the damper depends on the *current* in the *solenoid* which itself depends on the *power* of the car *battery*. The system finds a constraint on *power* of *battery* documented in page 24 of "progress report 10/90". From this Dedal returns an answer like:

> Maximum-force is a requirement on force of damper, force of damper depends on the current of the solenoid, the current of the solenoid depends on the power of the car battery, there is a requirement on power of the car battery that is documented in page 24 of progress report 10/90.

Thus far Dedal has been used on two industry scale design projects. The first project was the redesign of a continuously variable damper. Results of this study are discussed in [2]. The second project was the design of a Bioreactor. In this project, the design records were indexed during the design process. Table 1 summarizes the characteristics of these design projects.In case of the Damper and the Bioreactor projects both the design teams and the document database were co-located at a single site. With a new project called STEP, we are extending Dedal so that it can support situations where both the design teams and the design records are distributed.

USING DEDAL IN A DISTRIBUTED ENVIRONMENT

Design teams in industry like NASA are multidisciplinary and distributed geographically. Therefore for smooth progress of the design project the teams should be able to collaborate efficiently. To address this concern we are extending Dedal so that it can support a distributed scenario. In this scenario, designers who are geographically distributed are able to collaborate by indexing and retrieving sharable documents. To provide this capability we are integrating Dedal with World Wide Web (WWW) [4]. WWW is a distributed hypermedia system designed to provide access to documents distributed over different sites. It uses the HyperText Markup Language (HTML) to represent a hypertext document, and the HyperText Transfer Protocol (HTTP) to request and transmit documents over the network. WWW is accessible via a variety of browsers. We are working with Mosaic, a platform independent browser, and thus will be able to support collaboration between designers working on different platforms such as Unix, Macs and PC's. Mosaic also supports various media types and is suitable for sharing audio, video and information in other media.

Dedal's integration with Mosaic will provide designers with the following functionality:

- Accessing information at other locations.
- Making information available for team members at other locations.
- Organizing information at the local site using Dedal's indexing method.

TABLE 1. Application domains of Dedal. 'Real time' refers to whether the indexing happened during the design process or not. 'Designer indexing?' states whether the indexing was done by a member of the design team or not. In all the three cases the indexing task is done by a designer (from or outside the design team), not by a knowledge engineer as is typical in such systems.

Domain	Project duration	Platform	Capture Medium	Real time?	designer indexing?
Damper	7 mon	symbolics	vmacs	No	Yes (On Team)
Bioreactor	9 mon	unix	Maker	Yes	Yes (Outside Team)
STEP	2+ yrs	unix	Mosaic	Yes	Yes (On Team)

- Creation of an indexing model of the designed artifact.
- Maintaining vocabulary consistency among the different teams.
- Accurate retrieval of distributed design records using Dedal's retrieval engine.

Figure 1 describes the architecture of Dedal in the distributed scenario. As seen in the figure the documents reside at the local site with their indices. The indexing model defines relations among the indexing terms used by the design teams and resides at a central location, accessible and modifiable by all sites. This common model facilitates consistency in the vocabulary design teams use to describe their designs. We are starting to index and model design records from the project STEP (Satellite Test of the Equivalence Principle). We are working with two design teams, one located at Stanford University and the other at JPL (Jet Propulsion Laboratory, Pasadena) to support their collaboration and information sharing.

In the beginning the designers organize their documents by filling out a template (shown in figure 2). This template is implemented in Mosaic. It lets the designer create an index at the level of individual documents. Keywords in this form are the indexing terms that are project dependent. These keywords are related in the central indexing model of the project. As we integrate more of Dedal's functionality with Mosaic, designers will be able to index their documents at various levels of detail.

SUMMARY

Using Dedal in the continuously variable damper domain showed that Dedal accurately retrieves design records indexed using the conceptual indexing method. The experience in applying Dedal to the design of the Bioreactor showed that it is possible to index and model in real time, i.e. while keeping pace with the generation of new information, without undue burden on the designer. With STEP we are extending Dedal to a distributed scenario in which case designers themselves will index the design information they generate.

REFERENCES

[1] Baudin, Catherine; Underwood, Jody G. and Baya, Vinod, "Using Device Models to Facilitate the Retrieval of Multimedia Design Information", In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, pp 1237-1243, August 29-September 2,1993a.

[2] Baudin, Catherine; Underwood, Jody G.; Baya, Vinod and Mabogunje, Ade; "Dedal: Using Domain Concepts to Index Engineering Design Information", In *Proceedings of the 14th Annual Conference of the Cognitive Science Society*, Bloomington, Indiana, pp. 702-707, July 1992.

[3] Baya, V.; Gevins, J.; Baudin, C.; Mabogunje, A.; Toye, G. and Leifer, L.; "An Experimental Study of Design Information Reuse", In *Proceedings of the 4th International Conference on Design Theory and Methodology*, ASME, Sept. 13-16, Scottsdale, Arizona, pp. 141-147, 1992.

[4] Berners-Lee, T.J. and Cailliau, R. "World-Wide Web", *Computing in High Energy Physics 1992 Conference*, Anneey, France, 23-27 September 1992.

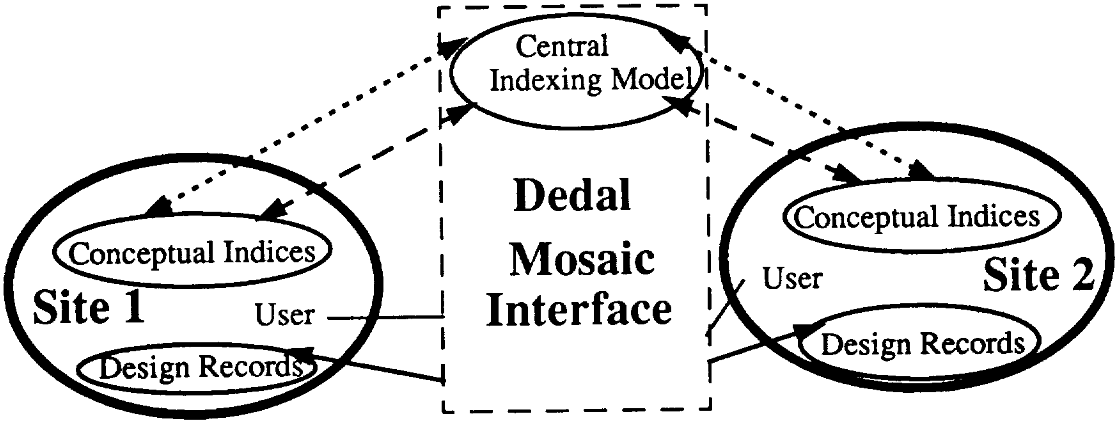
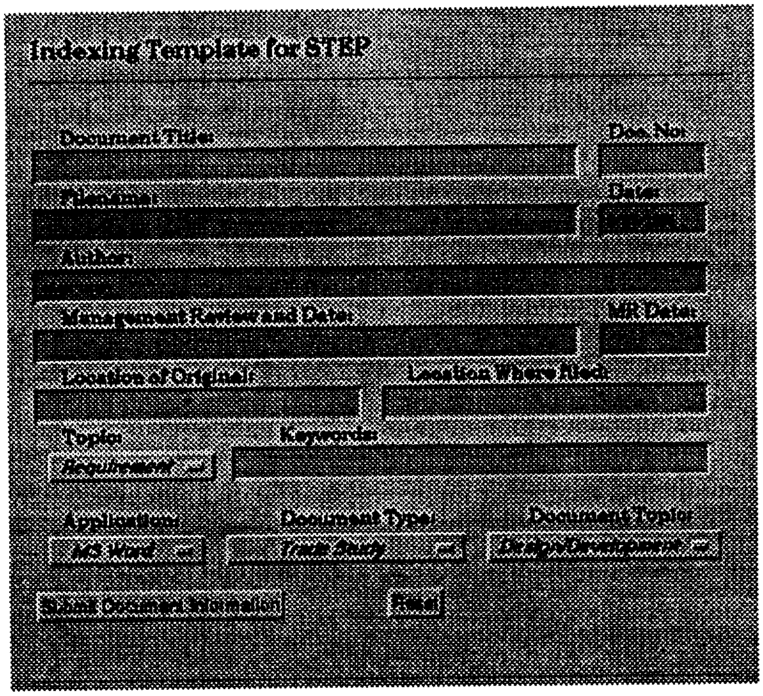


Figure 1. Architecture of Dedal in a distributed scenario. The solid arrows represent the sharing of information between designers at site 1 and 2 using the interface with Mosaic. Dashed arrows represent the creation of the index model by designers at both the sites using the local conceptual indices. Dotted arrow represents the access of the central index model by designers at both the sites for retrieval as well as creation of the index model.



Indexing Template for STEP

Document Title:
Doc. No:
Filename:
Date:
Author:
Management Review and Date:
MR Date:
Location of Original:
Location Where filed:
Topic: Requirement
Keywords:
Application: MS Word
Document Type: Trade Study
Document Topic: Design/Development

Figure 2. Template for indexing design records at the level of individual documents. This template is available as a form in Mosaic. Topics in this form are the domain independent conceptual indexing terms. Keywords are domain dependent conceptual indexing terms.

Operator Procedure Verification with a Rapidly Reconfigurable Simulator

Yumi Iwasaki*, Robert Engelmore, Gary Fehr, Richard Fikes
Adam Farquhar, Thomas Gruber

Knowledge Systems Laboratory, Department of Computer Science
Stanford University
701 Welch Road, Palo Alto, CA
94304 USA
tel. (415)723-3444 fax (415)725-5850
* iwasaki@ksl.stanford.edu

N95- 23679

KEY WORDS AND PHRASES

Automatic modeling, hybrid simulation, reconfigurable simulator, procedure verification

INTRODUCTION

Generating and testing procedures for controlling spacecraft subsystems composed of electro-mechanical and computationally realized elements has become a very difficult task. Before a spacecraft can be flown, mission controllers must envision a great variety of situations the flight crew may encounter during a mission and carefully construct procedures for operating the spacecraft in each possible situation. If, despite extensive pre-compilation of control procedures, an unforeseen situation arises during a mission, the mission controller must generate a new procedure for the flight crew in a limited amount of time. In such situations, the mission controller cannot systematically consider and test alternative procedures against models of the system being controlled, because the available simulator is too large and complex to reconfigure, run, and analyze quickly. A rapidly reconfigurable simulation environment that can execute a control procedure and show its effects on system behavior would greatly facilitate generation and testing of control procedures both before and during a mission.

There are several requirements that must be met by such a simulation system:

- **Reconfigurability** -- During a mission, the state of a component may change due to a fault or an unforeseen external event. During the design process, changes in the design of a physical system, which may occur concurrently with the design of an operating procedure, may require a modification to the procedure. For these reasons, it must be easy to change the simulation model to reflect the variety of configurations and conditions under which the spacecraft will be operated.
- **Simulation with imprecise or incomplete information** -- Exact and complete numerical data about the state of the system may not be available during design or in the presence of a fault. For example, when a leak is detected, the exact size of the leak is unlikely to be known. Therefore, the simulator must be able to predict behavior even if precise quantitative information about the state of the system is not available. If it is not possible to predict the behavior unambiguously, it should at least be able to produce a range of possible behaviors.
- **Explanation** -- When procedures produce unexpected results, it is difficult to interpret the raw simulation data, which may consist of values of hundreds of state variables in each of many states. The simulator should be able to produce a high-level, causal explanation of the simulation results, summarizing the salient information for the user and for documentation.

The How Things Work project at Stanford University has developed a system called DME (Device Modeling Environment) for modeling and simulating the behavior of electro-mechanical devices [1]. DME was designed to facilitate model formulation and behavior simulation of device behavior including both continuous and discrete phenomena. We are currently extending DME for use in testing operator procedures, and we have built a knowledge base for modeling the Reaction Control System (RCS) of the space shuttle as a testbed. We believe that DME can facilitate design of operator procedures by providing mission controllers with a simulation environment that meets all these requirements.

DME: THE RAPIDLY RECONFIGURABLE MODELING AND SIMULATION SYSTEM

DME is an evolving prototype of a "designer's associate" system, intended to support the design of electro-mechanical devices by providing effective tools for simulating and analyzing the behavior of such devices [2]. The DME system is intended as an experimental testbed and foundation on which to build new representation and reasoning capabilities. DME has already been developed to a sufficient level of maturity to provide both a demonstration vehicle and a useful experimental testbed within the project. Currently, DME provides the following capabilities:

Model formulation: DME uses the given information about the structure of a device to generate a mathematical model of its behavior. DME has knowledge of the physical phenomena in the domain, represented as *model fragments* in CML [3], a compositional modeling language developed jointly by leading members of the qualitative reasoning research community. Each model fragment describes a particular aspect of a conceptually distinct physical phenomenon in terms of the conditions under which it occurs and the consequences of its occurrence.

Given the structure of a device in terms of its components and their connections along with the conditions that hold in an initial state, DME formulates a mathematical model of the behavior of the device by composing applicable model fragments and simulates the behavior. We have also been developing techniques for automatically formulating a simulation model that embodies the abstractions, approximations, assumptions, and perspectives that are appropriate for a given analysis task [4].

Simulation: DME uses the model it generates to perform behavior simulation. When sufficient numerical information is available, simulation is carried out numerically. Otherwise, it simulates behavior qualitatively. In both cases, DME can simulate a mixture of continuous and discrete phenomena.

Explanation: On the basis of an initial device model and the behavioral predictions obtained through simulation, DME can answer a range of user queries about the structure and behavior of the modeled system [5]. An important element of the explanation approach in DME is the use of the simulator's models, rather than ad hoc "causal models" that are built specifically for explanation generation. In explaining how things work, people do use causal terminology. However, when analyzing the behavior of devices, engineers use formalisms such as logical and mathematical constraints that are not causal. DME infers causal dependencies among modeled parameters by analyzing logical and mathematical constraints.

Reasoning about functions: Understanding how a device works requires knowledge of both its intended function and its actual behavior. DME provides a representation formalism, called CFRL, for specifying intended functionality and a verification mechanism to determine whether a simulated behavior achieves an intended function [6].

USE OF DME FOR OPERATOR PROCEDURE VERIFICATION IN THE RCS

We have built a DME knowledge base for modeling the Reaction Control System (RCS) of the space shuttle, and we are extending DME to do simulation and evaluation of operator procedures. The RCS is the system of thrusters that are used to control the attitude of the space shuttle while it is in orbit. Oxygen and fuel are fed to the RCS jets from separate tanks. The thrusters do not have pumps; instead the flow is maintained by keeping the tanks pressurized with helium. Each tank has a dedicated helium supply tank to maintain pressurization.

Mission controllers have carefully constructed procedures for operating the RCS under a variety of conditions. For instance, if a leak in the RCS is detected, then two procedures are employed to secure the system and identify the location of the leak. In order to secure the system, the astronaut must close all of the RCS valves. The *RCS secure procedure* is to first close the valves nearest the thrusters and then to proceed upstream toward the helium tank until all of the valves have been closed. Once the system has been secured, the *isolation procedure* is to check the pressure in each of the segments between the closed valves. If the pressure in a particular segment is decreasing, then the leak has been isolated to that segment.

Even with procedures that seem simple, it is difficult to foresee the resulting interactions with the physical system. For instance, consider an alternative RCS secure procedure in which valves are closed in the opposite direction, starting with the main valve closest to the helium tank proceeding downstream towards the thrusters. Such a procedure is preferable for many systems -- as soon as the first (main) valve is closed, further propellant loss is prevented. In the RCS, however, this alternate procedure will result in cavitation inside the thrusters, leading to catastrophic damage.

Therefore, it is necessary to systematically test control procedures against models of the physical systems. When the execution of the procedure is simulated, the results need to be evaluated against the expected outcome of the procedure. At the time of this writing, DME has successfully formulated a behavior model of the RCS and simulated its behavior, given the specification of the RCS structure and initial conditions for the simulation. During simulation, DME allows the user to insert faults, such as leaks, or perform operator actions, such as opening and closing valves, to influence the course of behavior. As soon as any such changes are made, DME reformulates the model and continues simulating with the updated model. In this manner, DME has successfully predicted the results of the correct and incorrect valve closing sequences as described above in the presence of a leak.

We are currently extending DME in the following ways to enhance its support for procedure testing:

1) Develop the formal semantics of hybrid continuous and discrete models. This work is being carried out in collaboration with a team from the Xerox Palo Alto Research Center.
2) Extend the simulation mechanism to execute procedures automatically during simulation.
3) Expand CFRL to represent operator procedures and the intended effects of the procedures, which may not be explicit in the specification of the procedure itself.
4) Extend the verification mechanism to use the CFRL representation of operator procedures to verify whether the intended functions of a procedure are achieved in any given simulated trajectory of the system behavior.

An important type of knowledge about engineered devices is knowledge of its intended functions. Similarly, an important part of knowledge about operator procedures is knowledge of the function of the procedure, in other words, what the procedure is supposed to accomplish and how. CFRL was originally developed to represent device functions, but we believe it is also suitable for representing functions of operator procedures.

Figure 1 shows part of the proposed CFRL representation of the operator procedure to be invoked when over-pressurization of a propellant tank ($tk) is detected with both of the pressure regulators ($rega and $regb) open. Following the detection of the condition (node n0), the operator is to close the valves ($va and $vb) of both regulators (n1) and to open the thruster (n2), causing a decrease in the tank pressure (n3). When the pressure drops below 300 psi (n4), the operator is to reopen the valve of regulator A (n5). If the failure of regulator A is not detected by some other procedure (n7) within 60 seconds (n6), the operator is to conclude it is regulator B that has failed (n8).

The importance of functional knowledge extends not only to physical devices but also to *virtual devices* such as operator procedures. In the context of heterogeneous systems composed of electro-mechanical devices and control elements including digital computers and humans, operator procedures are as much a part of the system as any other physical component. It is important to evaluate the procedures under a variety of conditions, and such evaluation requires knowledge of their intended functions. We believe DME can facilitate the design of operator procedures by providing a means to explicitly represent a mission controller's intentions underlying a procedure and a useful simulation environment to evaluate whether a procedure achieves those intentions.

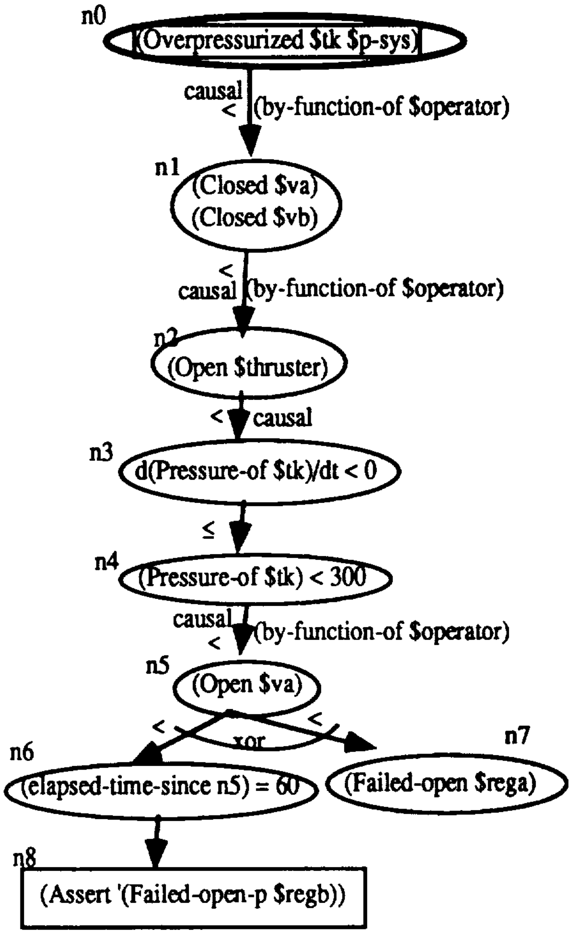


Figure 1. CFRL representation of an operator procedure

SUMMARY

In order to facilitate generation of procedures for operating complex dynamic spacecraft subsystems in a variety of expected and unexpected situations, it is essential to provide a modeling and simulation mechanism that can be quickly tailored to reflect a new configuration of the system being modeled. DME allows the user to change the system specification easily by altering the design or inserting faults to reflect a new situation. Reconfigurability of DME models comes from using compositional modeling technology. DME generates a new simulation model based on the altered specification and simulates the operator actions to predict the system behavior resulting from the actions. Such a facility will not only allow mission controllers to verify the safety of new procedures quickly, thereby avoiding unforeseen negative side effects, but also will be an essential component in a future automatic procedure generation and testing system.

ACKNOWLEDGMENT

This research was sponsored by the Advanced Research Projects Agency, ARPA Order 8607, monitored by NASA Ames Research Center under grant NAG 2-581; and by NASA Ames Research Center under grant NCC 2-537. We thank Drs. Vijay Saraswat and Daniel Bobrow of the Xerox Palo Alto Research Center for their involvement with us in the development of knowledge representation and formal semantics of hybrid systems.

REFERENCES

[1] Low, C.M. and Iwasaki, Y., *Device modelling environment: an interactive environment for modelling device behaviour*. Intelligent Systems Engineering, 1993. **1**(2): p. 115-145.
[2] Iwasaki, Y., *et al.* How Things are Intended to Work: Capturing Functional Knowledge in Device Design. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. 1993. Chambery, France.
[3] Falkenhainer, B, *et al.* CML: A Compositional Modeling Language. KSL Technical Report, January 1994.
[4] Iwasaki, Y. and Levy, A.Y. Automated Model Selection for Simulation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*. 1994. Seattle, WA. AAAI Press.
[5] Gautier, P.O. and Gruber, T.R. Generating Explanations of Device Behavior Using Compositional Modeling and Causal Ordering. In *Proceedings of the Eleventh National Conference on Artificial Intelligence.* 1993. Washington, D. C. AAAI Press/the MIT Press.
[6] Vescovi, M., *et al.* CFRL: A Language for Specifying the Causal Functionality of Engineered Devices. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*. 1993. Washington, D. C. AAAI Press/The MIT Press.

AMPHION: Specification-Based Programming for Scientific Subroutine Libraries

N95- 23680

Michael Lowry, Andrew Philpot, Thomas Pressburger, and Ian Underwood
Recom Technologies; AI Research Branch, M.S. 269-2
NASA Ames Research Center
Moffett Field, CA 94305, USA
Tel: (415) 604-3369 Fax: (415) 604-3594
lowry@ptolemy.arc.nasa.gov

Richard Waldinger and Mark Stickel
AI Center; SRI International

KEY WORDS AND PHRASES

Artificial Intelligence, knowledge-based software engineering, NAIF, software engineering, software reuse.

OVERVIEW

AMPHION is a knowledge-based software engineering (KBSE) system that guides a user in developing a diagram representing a formal problem specification. It then automatically implements a solution to this specification as a program consisting of calls to subroutines from a library. The diagram provides an intuitive domain-oriented notation for creating a specification that also facilitates reuse and modification.

AMPHION'S architecture is domain independent. AMPHION is specialized to an application domain by developing a declarative domain theory. Creating a domain theory is an iterative process that currently requires the joint expertise of domain experts and experts in automated formal methods for software development.

AMPHION has been applied to JPL's NAIF domain through a declarative domain theory that includes an axiomatization of JPL's SPICELIB subroutine library. Testing with planetary scientists demonstrates that AMPHION's interactive specification acquisition paradigm enables users to easily develop, modify, and reuse specifications after only a short tutorial. AMPHION routinely synthesizes programs consisting of dozens of SPICELIB subroutine calls from these specifications in just a few minutes.

Qualitative assessments indicate an order of magnitude productivity increase using AMPHION over manual program development. AMPHION is currently undergoing alpha testing in preparation for distribution to the NAIF community. Other NASA domains are under consideration. Future research will address the technology needed for domain experts to develop their own AMPHION domain theories with only minimal consultation from experts in formal methods.

MOTIVATION

Within the space science community, subroutine libraries are a ubiquitous form of software reuse. However, space scientists often do not make effective use of libraries. Sometimes this happens because a subroutine library is developed without following good conventional software engineering practices, resulting in inadequate documentation, untrustworthy code, and a lack of coherence in the different functions performed by the individual routines. However, even when a subroutine library is developed following the best conventional software engineering practices, users often have neither the time nor the inclination to fully familiarize themselves with it. The result is that most users lack the expertise to properly identify and assemble the routines appropriate to their problems. This represents an inherent knowledge barrier that lowers the utility of even the best-engineered software libraries: the effort to acquire the knowledge to effectively use a subroutine library is often perceived as being more than the effort to develop the code from scratch. AMPHION is an effective solution to this knowledge barrier.

The objective of AMPHION is to enable users who are familiar with the basic concepts of an application domain to program at the level of abstract domain-oriented problem specifications, rather than at the detailed level of subroutine calls. AMPHION breaks through the knowledge barrier by enabling use of a subroutine library without having to absorb all the documentation about a library, especially the plethora of implementation details such as the representation conventions for subroutine parameters.

NAIF APPLICATION

The first application domain for AMPHION is solar-system kinematics, as implemented in the SPICELIB subroutine library developed by the Navigation Ancillary Information Facility (NAIF) at JPL. SPICELIB is an extremely well-engineered library used by planetary scientists to plan and analyze the observing geometry for data collected during interplanetary missions or by space-based telescopes. A domain theory was developed that includes an abstract formalization of solar-system kinematics suitable for specifying problems, and the knowledge needed to implement solutions using SPICELIB. To date, Amphion has demonstrated the following essential capabilities for real-world KBSE:

1. Users without training in formal methods readily develop domain-oriented diagrams corresponding to formal problem specifications using Amphion's specification-acquisition tools.
2. Users can reuse, modify, and maintain previously developed specifications, thereby elevating the software life cycle from the code level to the specification level.
3. Automatic deductive program synthesis achieves acceptable performance, given an appropriately structured domain theory and moderate use of theorem-proving tactics.

Programming at the Specification Level

To enable users to program at the specification level, AMPHION consists of a specification-acquisition component to guide users in developing a formal specification, and a program synthesis component that automatically generates a program implementing a solution to the specification. Users enter specifications graphically through a menu-guided graphical user interface (GUI). Figure 1 is an example of a completed specification: it denotes the problem of predicting the solar incidence angle at the point on Jupiter closest to Galileo at a particular time. (This is the sub-spacecraft point). The specification acquisition component performs semantic checks on completed specification diagrams, and then automatically translates them to a logical form used by the program synthesis component.

The output of program synthesis for the NAIF application is a FORTRAN-77 program consisting of calls to the SPICELIB subroutine library. AMPHION generated the SOLAR program in Figure 2 from the specification in Figure 1 in 52 seconds of CPU time on a Sparc 2. In over a hundred programs generated by AMPHION for the NAIF domain to date, the CPU time has exceeded three minutes in only four cases. This is an unprecedented level of performance for the deductive synthesis approach, developed over 25 years ago [1,2]. Most of the program synthesis component is independent of the target output language. It would only take two weeks of work to adapt AMPHION for a different output language such as C or UNIX shell files.

AMPHION's specification language for the NAIF domain is at the level of abstract geometry. This specification language is part of the declarative domain theory. The vocabulary is basic Euclidean geometry (e.g., points, rays, ellipsoids, and intersections) augmented with astronomical terms (e.g., planets, spacecraft, and photons; the latter for specifying constraints used in calculating light-time correction). The specification language does not include the myriad implementation details required for correctly calling SPICELIB subroutines, such as coordinate frames, units, time systems, etc; these details are automatically deduced during program synthesis. The user only needs to define the abstract problem and the desired representation conventions for the program inputs and outputs.

AMPHION's GUI bears a superficial resemblance to data-flow oriented graphical programming environments. For example, Apple's HOOKUP application enables users to select icons from a palette that represent individual subroutines, and then connect input and output ports. However, these environments only provide an alternate notation to conventional programming languages. In contrast, AMPHION enables a radical separation between the level at which users

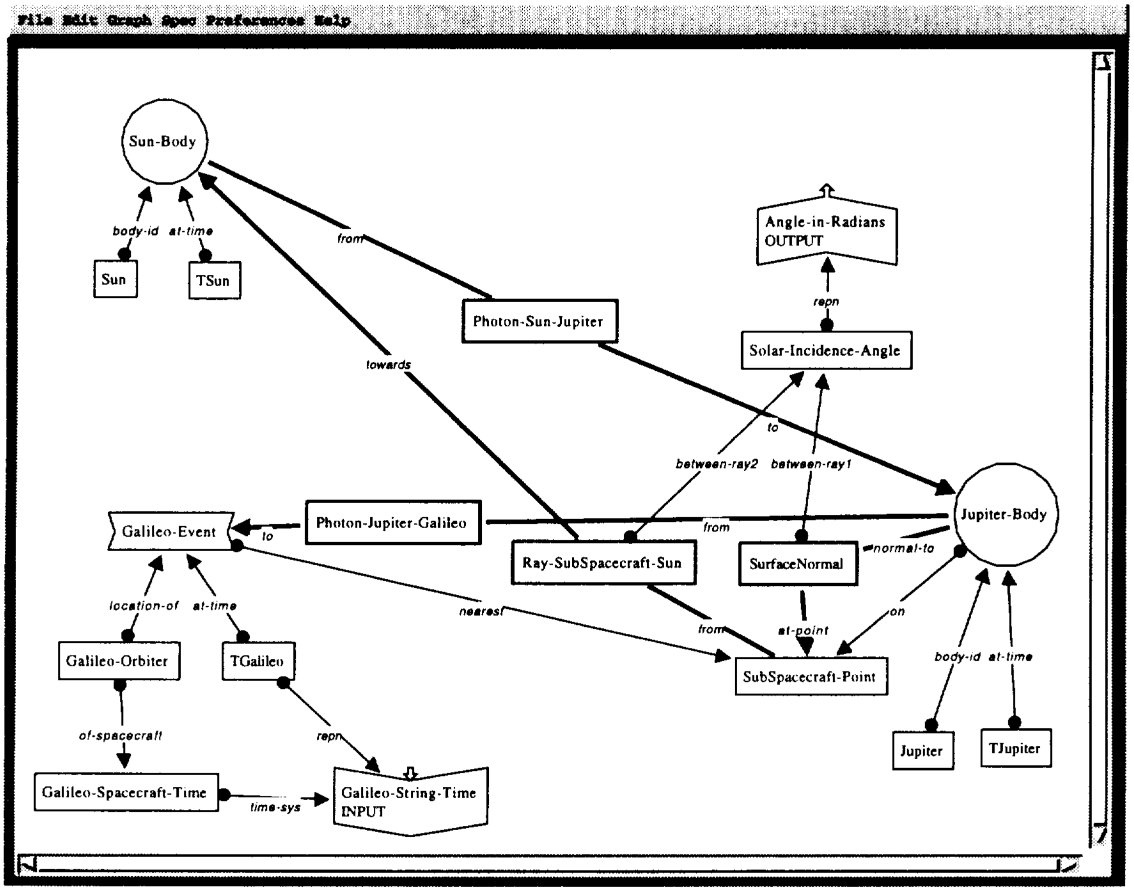


Figure 1: Diagram for solar incidence angle developed interactively with AMPHION.

```
      SUBROUTINE SOLAR ( GALILE, ANGLEI )
C     Input Parameters
      CHARACTER*(*) GALILE
C     Output Parameters
      DOUBLE PRECISION ANGLEI
C     Function Declarations
      DOUBLE PRECISION VSEP
C     Parameter Declarations
      INTEGER JUPITE
      PARAMETER (JUPITE = 599)
      INTEGER GALIL1
      PARAMETER (GALIL1 = -77)
      INTEGER SUN
      PARAMETER (SUN = 10)
C     Variable Declarations
      DOUBLE PRECISION RADJUP ( 3 )
      DOUBLE PRECISION E
      DOUBLE PRECISION PVGALI ( 6 )
      DOUBLE PRECISION LTJUGA
      DOUBLE PRECISION V1 ( 3 )
      DOUBLE PRECISION X
      DOUBLE PRECISION PVJUPI ( 6 )
      DOUBLE PRECISION LTSUJU
      DOUBLE PRECISION MJUPIT ( 3, 3 )
      DOUBLE PRECISION V2 ( 3 )
      DOUBLE PRECISION X1
      DOUBLE PRECISION DV2V1 ( 3 )
      DOUBLE PRECISION PVSUN ( 6 )
      DOUBLE PRECISION XDV2V1 ( 3 )
      DOUBLE PRECISION V ( 3 )
      DOUBLE PRECISION N ( 3 )
      DOUBLE PRECISION PN ( 3 )
      DOUBLE PRECISION DV2N ( 3 )
      DOUBLE PRECISION XDV2N ( 3 )
      DOUBLE PRECISION DXDV2V ( 3 )
      DOUBLE PRECISION XDXDV2 ( 3 )
C     Dummy Variable Declarations
      INTEGER DMY10
      DOUBLE PRECISION DMY20 ( 6 )
      DOUBLE PRECISION DMY60 ( 6 )
      DOUBLE PRECISION DMY130
      CALL BODVAR ( JUPITE, 'RADII', DMY10, RADJUP )
      CALL SCS2E ( GALIL1, GALILE, E )
      CALL SPKSSB ( GALIL1, E, 'J2000', PVGALI )
      CALL SPKEZ ( JUPITE, E, 'J2000', 'NONE', GALIL1,
                   DMY20, LTJUGA )
      CALL VEQU ( PVGALI ( 1 ), V1 )
      X = E - LTJUGA
      CALL SPKSSB ( JUPITE, X, 'J2000', PVJUPI )
      CALL SPKEZ ( SUN, X, 'J2000', 'NONE', JUPITE,
                   DMY60, LTSUJU )
      CALL BODMAT ( JUPITE, X, MJUPIT )
      CALL VEQU ( PVJUPI ( 1 ), V2 )
      X1 = X - LTSUJU
      CALL VSUB ( V1, V2, DV2V1 )
      CALL SPKSSB ( SUN, X1, 'J2000', PVSUN )
      CALL MXV ( MJUPIT, DV2V1, XDV2V1 )
      CALL VEQU ( PVSUN ( 1 ), V )
      CALL NEARPT ( XDV2V1, RADJUP ( 1 ),
                   RADJUP ( 2 ),RADJUP ( 3 ),N, DMY130)
      CALL SURFNM ( RADJUP ( 1 ), RADJUP ( 2 ),
                   RADJUP ( 3 ), N, PN )
      CALL VSUB ( N, V2, DV2N )
      CALL MTXV ( MJUPIT, DV2N, XDV2N )
      CALL VSUB ( V, XDV2N, DXDV2V )
      CALL MXV ( MJUPIT, DXDV2V, XDXDV2 )
       ANGLEI = VSEP ( XDXDV2, PN )
      RETURN
      END
```

Figure 2: SOLAR program generated by AMPHION from Figure 2.

specify problems and the level at which solutions are implemented by the program synthesis component. AMPHION's GUI provides an alternate notation to formal specifications written in mathematical logic. The notation of mathematical logic can be formidable; that is one reason that specification-based software engineering life cycles have not previously been adopted in practice.

AMPHION's GUI employs an object-oriented paradigm for interactively developing problem specifications. Conceptually, a user develops a problem specification by first defining a configuration, and then declaring a subset of the objects in a configuration to be inputs or outputs of the desired program. A configuration is a set of abstract objects and their relationships.

A user generates a configuration through the actions of adding objects, deleting objects, moving the edges between objects that define their interrelationships, and by merging objects together. Adding and deleting objects are done through menus; moving edges and merging objects are done by directly manipulating the diagram. Declaring an object to be an input or output of the desired program brings up a menu of the possible data-representation conventions: coordinate systems for locations, time systems for time, and units of measurement. These alternative representation conventions are also part of the declarative domain theory.

AMPHION's specification-acquisition component not only enables specifications to be developed from scratch, but it is also especially well suited for specification reuse and modification. The abstract graphical notation makes it much easier to identify the required modifications than it is to trace through dependencies in code. AMPHION's editing operations facilitate making the changes. Furthermore, there is no possibility of introducing bugs in the code, since AMPHION synthesizes the code from scratch for the modified specification.

FUTURE DIRECTIONS

Why the name AMPHION? AMPHION was the son of Zeus who used his magic lyre to charm the stones lying around Thebes into position to form the city's walls. The AMPHION system's expertise lies in charming subroutines into useful programs through SNARK, an advanced automatic theorem prover developed at SRI International. A tutorial introduction for this deductive approach to program synthesis can be found in [3], while more details on the use of SNARK for synthesizing programs in the NAIF domain can be found in [4]. One advantage of the deductive approach is that a synthesized program is guaranteed to be a correct implementation of a user's specification, with respect to the domain theory. This reduces the software verification problem to a one-time verification of the domain theory. The declarative nature of the domain theory simplifies verification.

Because it uses a generic architecture, described in [5], AMPHION can be applied to other domains and subroutine libraries by developing the appropriate domain theories. The methodology for developing suitable AMPHION domain theories is described in [6]. Developing the initial NAIF domain theory took three months of collaboration between a NAIF expert and experts in automated formal approaches to program synthesis. Much of the subsequent refinements to the domain theory were straightforward and could likely be done by domain experts with the appropriate tools. Future research will include developing such tools.

REFERENCES

[1] Green, C.; 1969. "Application of Theorem Proving to Problem Solving", in IJCAI-69.

[2] Waldinger, R.; and Lee, R; 1969. "PROW: A Step Toward Automatic Program Writing", in IJCAI-69.

[3] Manna, Z.; and Waldinger, R.; 1992. "Fundamentals of Deductive Program Synthesis", in IEEE Transactions on Software Engineering (18) 8.

[4] Stickel, M.; Waldinger, R.; Lowry, M.; Pressburger, T.; and Underwood, I.; 1994. "Deductive Composition of Astronomical Software from Subroutine Libraries", in CADE-12.

[5] Lowry, M.; Pressburger, T.; Philpot, A.; and Underwood, I.; 1994. "A Formal Approach to Domain-Oriented Software Design Environments", in KBSE-94.

[6] Lowry, M.; Pressburger, T.; Philpot, A.; and Underwood, I.; 1994. "AMPHION: Automatic Programming for Scientific Subroutine Libraries", in ISMIS-94.

SESSION MD

Monitoring and Diagnostics

N95- 23681

TIKON
An Intelligent Ground Operator Support System

Thomas Görlach, Gerhard Ohlendorf, Frank Plaßmeier, Uwe Brüge
DASA/ERNO, PO-Box 105909, D-28259 Bremen, Germany
Phone: +49-421-539-5502 Telefax: +49-421-539-4400
E-mail : Gerhard.Ohlendorf@erno.de

ABSTRACT

This paper presents first results of the project "Technologien für die intelligente Kontrolle von Raumfahrzeugen" (TIKON). The TIKON objective was the demonstration of feasibility and profit of the application of artificial intelligence in the space business. For that purpose a prototype system has been developed and implemented for the operation support of the Roentgen Satellite (ROSAT), a scientific spacecraft designed to perform the first all-sky survey with a high-resolution X-ray telescope and to investigate the emission of specific celestial sources. The prototype integrates a scheduler and a diagnosis tool both based on artificial intelligence techniques. The user interface is menu driven and provides synoptic displays for the visualization of the system status. The prototype is used and tested in parallel to an already existing operational system.

KEYWORDS AND PHRASES

Diagnosis, ground operations, scheduling, synoptic displays.

INTRODUCTION

The TIKON project is sponsored by the German Space Agency (DARA) and performed by DASA/ERNO with support of the German Space Operation Center (GSOC). It will be finished in December 1994. As shown in Figure 1 the TIKON system consists of three main parts: The synoptic display manager, the scheduler and the diagnosis tool.

The goal of the project is the development of a ground operator assistant system for the ROSAT satellite ground activities. Those activities consist of :

- the scheduling of a half year observation plan for X-ray stars which is constrained by user requirements, orbital aspects and contract requirements
- the scheduling of a weekly observation plan considering additional short term wishes of the users and actual orbital data
- the monitoring of ROSAT housekeeping telemetry-data for the attitude measurement and control system (AMCS) and the data handling system (DHS). This includes the detection and isolation of anomalies and failures.

The above mentioned activities are actually performed using classical operational methods which offer not very much clearness and graphical support for the operator.

TIKON provides a user friendly and convenient tool on a SUN workstation which visualizes the incoming telemetry-data on a synoptic display. The synoptic display shows the ROSAT system in different component levels and depicts finally the selected subsystem's data in a graphical form on meters and charts. In addition to that

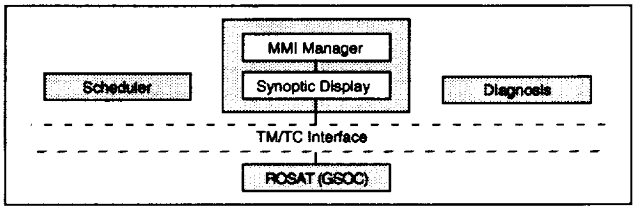


Figure 1. Main Components

46

limit violations are indicated by color changes. For the failure detection and analysis a so-called diagnoser is installed which evaluates the probability of component failures out of a combination of telemetry-data.

USER INTERFACE

The applied synoptic display is an intelligent user interface that processes ROSAT telemetry data in a graphical and user friendly way and that reacts on events by displaying the subsystem's data in question. Those events may be a limit violation or user requests. Figure 2 depicts in a simplified manner the main display of the tool representing the ROSAT subsystems at one glance.

DIAGNOSIS

An important objective of the TIKON project is the evaluation of advanced Fault Detection, Isolation and Recovery (FDIR) methods in order to identify the potentials of improved operator support in case of spacecraft malfunctions.

Application

In the frame of the TIKON project, the ROSAT AMCS has been selected as a sample application for knowledge based FDIR. Twenty knowledge bases related to ROSAT AMCS components have been defined which are used to evaluate the ROSAT Telemetry (TM) data in order to find malfunctions of these components. The FDIR system is executed as a separate process that analyses pre-processed TM data, displays diagnostic results in specific windows and also sends the diagnostic results to a synoptic display utility in order to visualize them. Whereas the synoptic display offers an easy to comprehend schematic view of the AMCS components, the FDIR windows provide more detailed information that is closer related to the diagnostic processing.

Method

The TIKON FDIR component is based on the Connection Matrix Based Expert System Tool (CONNEX) technology, which in the frame of

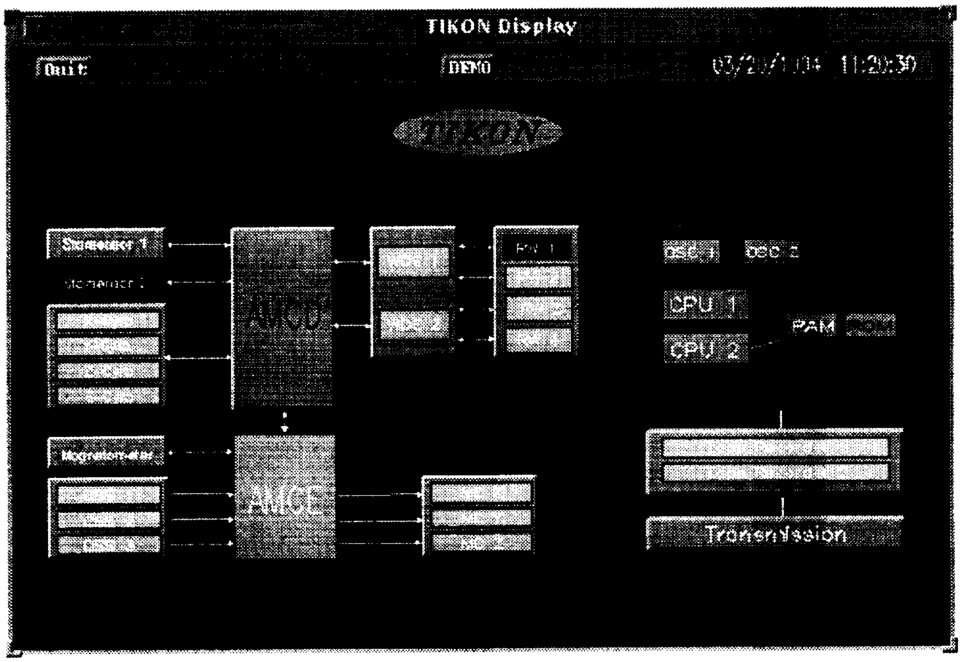


Figure 2. TIKON Synoptic Display (Main Level)

the Computer Based Payload Operation Support System (COMPASS) project has already been applied to a payload during the German D2 Spacelab mission. For the TIKON project, template knowledge bases have been added as a new feature in order to facilitate knowledge acquisition and maintenance in the presence of multiple instances of structurally similar technical systems. For example, ROSAT contains four Gyros of similar structure with similar related telemetry data. Instead of defining four distinct knowledge bases only one template knowledge base needs to be defined which is then used to instantiate four concrete knowledge bases.

Connection Matrices can be seen as extended decision tables, allowing for fault diagnosis based on approximate matches between observed exception patterns and expected exception patterns for predefined faults. Key advantages of this approach are:

- increased robustness against local deviations between expected and observed system behavior
- better ability to cope with evolving anomalies and improved early warning capability
- increased robustness against sensor failures
- improved ability to handle multiple faults

Basically, diagnosing a system for a given exception vector is performed as follows: First, the diagnoses are grouped into so called discrimination sets. Each discrimination set consists of diagnoses which are related to the same set of observed exceptions. Only those discrimination classes which are related to a set of exceptions that is not a true subset of the set of observed exceptions related to another discrimination class are considered for further processing. At least one member of each of these discrimination classes must be a valid diagnosis, since it accounts for at least one otherwise unexplained exception. The members of a particular discrimination class on the other hand are competitors, since they account for the same subset of exceptions. The selection among the members of a discrimination class is performed by computing the proximity ratio between the cardinality of the intersection between observed exceptions and the exceptions expected for the particular fault and the cardinality of the set of exceptions expected for this particular fault. Figure 3 illustrates an example of the diagnostic processing.

	A1	A2	A3	A4
E1		*		*
=> E2	*	*		
=> E3	*	*	*	
E4	*			*
=> E5			*	*
E6	*			*

Figure 3. Example of a Connection Matrix

In Figure 3 the inputs E1 to E6 denote exceptions, A1 to A4 anomalies (i.e. faults). An asterisk indicates that the anomaly in the top most row and the exception in the left most column are related, i.e. that this anomaly will cause this exception. Provided that the exceptions E2, E3 and E5 are observed, the reasoning goes as follows:

There are three discrimination classes:

C1 = {A1, A2} which accounts for E2 and E3

C2 = {A3} which accounts for E3 and E5

C3 = {A4} which accounts for E5

The elements of C3 are discarded, since the set of exceptions they account for is a true subset of those the elements of C2 account for.

A3 is selected, since it is the only anomaly that accounts for E3 and E5 simultaneously.

A2 is preferred over A1, since it has the higher proximity ratio (2/3 vs. 1/2).

A3 and A2 remain as final diagnoses, with A1 being a possible alternative to A2.

SCHEDULING

The TIKON scheduling tool is based on the Mission Activities and Resources Scheduler (MARS), a general purpose scheduling tool developed by DASA/ERNO for scheduling of space missions. A new MARS feature required in the scope of the TIKON study is an optimization scheduling strategy, which depends on user defined optimization criteria for a Schedule. MARS intends to find not only a Schedule fulfilling all hard constraints but also tries to optimize the Schedule by pre-selecting Activities according the optimization criteria before applying of the Rule system.

Objectives

The scheduling of ROSAT concentrates only on the pointing phase. During this phase typically 1800 requests for observations of different sources must be handled by the system to schedule a period of 6 months. These need to be scheduled as efficiently as possible to avoid wasting of valuable observation time.

The observations are basically constrained by:

- must be scheduled within a slot between particle belts (hard celestrial constraint)
- their visibility (hard celestrial constraint)
- observation instrument (hard operational constraint)
- time share between observations of different countries (soft operational constraint)
- Observations must be separated by a slew operation (hard operational constraint)

Thereby, two principal goals shall be achieved:

- Generation of a timeline, fulfilling for all scheduled observation requests the constraints
- This timeline shall maximize the observation time in comparison to the principal available slot duration during the pointing phase

Approach

For TIKON the following functionalities had to be added to the MARS system:

- Optimizing scheduling process
- Possible interruption of Activities

These functionalities have been added without changing the principal way of the MARS scheduling method. The advantage is that future not yet known constraints might well be handled by the generic MARS data description possibilities and scheduling functionality.

The following approach for the representation of the ROSAT scheduling problem was used:

- All observable sources are represented by MARS Resources,which have as discrete Availability Profiles the time spans where the source is visible (i.e. could be observed) or not. These Resources have the type reusable since they are handled like targets, which however can only be observed one at a time.
- An observation request for a source is represented by a MARS Activity, which basically has as Resource Request the specific Resource representing the source to be pointed at.
- The scheduling process shall schedule Activities under the following conditions:
    - All hard constraints must be fulfilled
    - Activities must not be scheduled parallel, they can be interrupted
    - The soft constraints (e.g. country share) are met as far as possible
    - The generated timeline shall approximate the optimization criteria as far as possible

For an example of a ROSAT scheduling situation see Figure 4 (next page).

Scheduling and Optimization Approach

The general MARS Scheduling can be seen as a heuristic search process, but with a certain restriction of the search space. This can inhibit to find the best solution, but allows to handle praxis relevant and therefore very complex problems.

Aim of an optimization is to find a Goal Schedule s_g which is optimal with respect to some goal function v:

$$v(s_g) = \text{Optimal !}$$

The goal function v for a TIKON Schedule is defined as the percentage of the unused observation time measured against the available observation time. Then the best Schedule would use all available observation time.

The general idea of an optimizing strategy in MARS is now the following:

Use function v as an estimation of the heuristic function which guides the search process so that the optimal search path corresponds to the optimal solution in the sense of the function v. Even if not the complete search space can be used, it is hoped that MARS will find a sub-optimal solution.

The scheduling algorithm was extended by a pre-selection module which provides the set of Activities fulfilling all hard constraints and which would optimize the so far generated Schedule with respect to v. To provide enough Activities for further processing also a certain percentage of sub-optimal candidates is taken into account. Thereafter the Rule system is applied to achieve the soft constraints.

CONCLUSION

Although the test phase has just been started and will continue until end of this year some first results are:

- improvement for operators through the hierarchical user interface which allows a quick orientation
- this interface enables also a reduction of required training periods for newcomers
- the integration of data acquisition and diagnosis as well as the presentation of diagnostic results at various levels of detail reduces the operator workload and leads to an accelerated failure diagnosis cycle
- The graphical plot facilities of the Schedule represent a new quality of user information, e.g. about possible alternatives
- The new scheduling approach achieves in first tests a utilization of 88 percent of the possible observation time while fulfilling the soft constraint with a deviation of less than 5 percent.

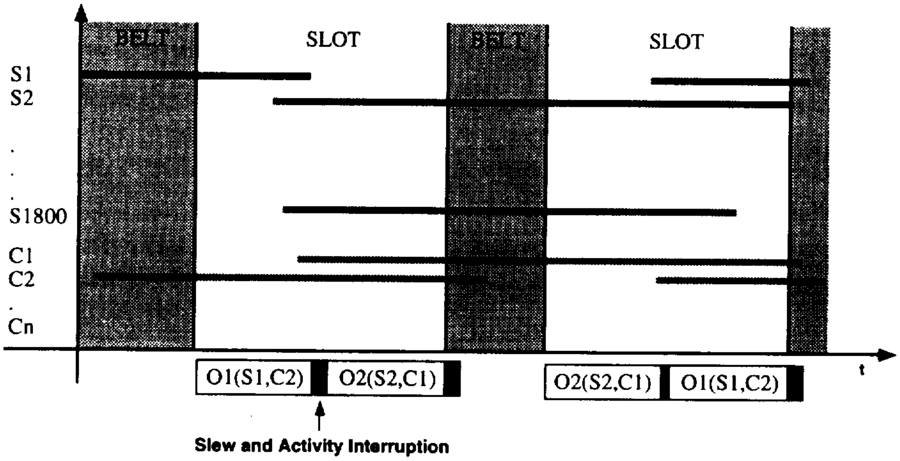


Figure 4. ROSAT Scheduling

N95- 23682

Toward an Automated Signature Recognition Toolkit for Mission Operations

T. Cleghorn (NASA/JSC) **C. Culbert** (NASA/JSC) **D. Hammen**[†](MITRE)
P. Laird[*] (NASA/ARC) **M. Macha** (NASA/JSC) **T. Moebes** (SAIC)
L. Perrine (NASA/JSC) **R. Saul** (NASA/RECOM) **R. Shelton**[†] (NASA/JSC)

KEY WORDS AND PHRASES

Event detection, pattern recognition, signature recognition, telemetry, time series.

SIGNATURE RECOGNITION

Signature recognition is the problem of identifying an event or events from its time series. The generic problem has numerous applications to science and engineering. At NASA's Johnson Space Center, for example, mission control personnel, using electronic displays and strip chart recorders, monitor telemetry data from three-phase electrical buses on the Space Shuttle and maintain records of device activation and disactivation. Since few electrical devices have sensors to indicate their actual status, changes of state are inferred from characteristic current and voltage fluctuations. Controllers recognize these events both by examining the waveform signatures and by listening to audio channels between ground and crew. Recently the authors have developed a prototype system that identifies major electrical events from the telemetry and displays them on a workstation. Eventually the system will be able to identify accurately the signatures of over fifty distinct events in real time, while contending with noise, intermittent loss of signal, overlapping events, and other complications. This system is just one of many possible signature recognition applications in Mission Control. While much of the technology underlying these applications is the same, each application has unique data characteristics, and every control position has its own interface and performance requirements. There is a need, therefore, for CASE tools that can reduce the time to implement a running signature recognition application from months to weeks or days. This paper describes our work to date and our future plans.

DEVELOPING A SIGNATURE RECOGNITION APPLICATION

A typical signature-recognition application monitors a data stream and is activated by an "event," as defined by the satisfaction of certain conditions. Data is then taken from the data stream, filtered and converted, and passed to a pattern-recognition module. The module decides to what class the event belongs and adjusts the controller's display. The event may also be captured for later offline use.

The following six steps are followed in designing and implementing a signature recognition application:

1. *Identify the users.* At Mission Control the end users (and the domain experts) are mission controllers.
2. *Acquire the data.* Training the system to identify signatures requires that one collect a set of correctly labeled signatures. Other information in the form of rules may also be required. This data is usually in short supply, either because some events occur rarely (e.g., engine failures) or because accurately labeled events are unavailable in machine-readable form. Ensuring the accuracy of the training data is, of course, critical.
3. *Design the pattern-recognition method(s).* Along with classical pattern recognition (PR) methods, more general techniques like neural networks, genetic algorithms, and

---

[*] NASA Ames Research Center, Moffett Field, CA. 94035-1000 (USA). Email: LAIRD@PTOLEMY.ARC.NASA.GOV

[†] NASA Johnson Space Flight Center, PT4, 2102 NASA Road 1, Houston, TX 77058-3696 (USA). Email: SHELTON@GOTHAMCITY.JSC.NASA.GOV, DHAMMEN@MITRE.ORG

52

decision trees are effective and easy to understand. User confidence in the PR method is very important: for our users to accept the application, they need (and want) to understand the PR method conceptually, and are unwilling to base decisions upon an inscrutable answer from a "black box."

4. *Design the user interface.* Ideally the user interface should be an integral part of the system design from inception. Since a certain amount of experimentation is needed to ascertain the best presentation, a flexible interface tool for rapid prototyping is invaluable.
5. *Engineer the system architecture.* Online data typically flow from the input line, through various filters and formatting routines, onto and off of queues, to pattern recognizers, screen displays, and archival storage. Ensuring that the system can keep pace with this flow is essential.
6. *Evaluate the results.* One must plan to monitor the accuracy and performance of the running system over time, because the environment is constantly changing and the signatures with it.

THE SIGNATURE RECOGNITION TOOLKIT CONCEPT

Our goal is to automate the above steps to the extent possible, and to place much of the specification, implementation, and maintenance tasks into the hands of the end users. Current application development environments like AVS, Khoros, Matlab, etc., are useful for prototyping but do not produce a real-time application. Naturally, however, we borrow many ideas from these existing toolkits.

The task of enlisting the users is, of course, inherently human, so automation begins with the data acquisition step. At Johnson Space Center's Mission Control, flexible subsystems are in place that distribute telemetry data to the applications. In order to apply pattern recognition to this stream, we must identify repeatable event instances in the available data that can then be subjected to pattern analysis. Data segmentation—extracting finite events from the stream—can be very subtle owing to noise and other unforeseen properties. Alternately, one can monitor the stream continuously, treating every data sample as an event; but when the sample rate is high, performance requirements will severely restrict the possible analysis.

A "Data Warehouse" (DW) tool that runs offline can capture signatures in a database, display them for for domain experts to examine and label, and later format them as input to training programs. The same tool can record rule-based knowledge from the experts and, later in the process, help with system performance monitoring (see below).

The third step (designing a PR technique) can be substantially automated, but will often entail some assistance from an expert. Any good toolbox contains multi-purpose neural network, decision tree, and genetic algorithm software, as well as more specialized techniques. But there are so many problem-specific issues—e.g., the amount and kind of generalization, measures of accuracy and confidence, tradeoffs between speed and power, noise compensation, feature extraction, training time versus recognition time, and allowance for future growth in training data and the number of classification labels—that we believe that the support of a PR engineer will be required.

Step four is greatly simplified by today's interface building tools. Connecting the interface widgets to the data stream is straightforward except for the task of ensuring that dataflow bottlenecks do not lose input data. This task may require the assistance of a software engineer. The time to accomplish this task can be mitigated if the toolbox modules are fitted with calling interfaces so that they can be "plugged into" one another without total recompilation, much like the components on an electronic breadboard.

Finally, part of ongoing performance monitoring includes the task of having the users validate the labels assigned by the system, and using the results to check that the accuracy of the system does not degrade. We find that classifiers often need to be retrained. The DW tool can archive the online events with the system-assigned labels, collect the results of user validation, calculate and report the accuracy, and automatically retrain the classifiers on the most current samples.

In summary, a signature-application toolkit will contain the following software components integrated into a uniform environment:

- A mechanism for capturing data and segmenting signature events.
- A DataWarehouse tool that saves labeled events for training and testing and formats them in various ways for output to software components. Later this same tool supports the process of monitoring the performance

and accuracy of the system over time.

- A library of PR modules that can be trained to classify events to specified accuracy and confidence levels.
- An Interface Builder so that end users can design and maintain their view of the events as they occur.
- A library of dataflow components equipped with a flexible module-to-module interface, so that the system can be assembled simply by describing the modules and their connections.

Given this, the users will still need a PR engineer to define events and evaluate the PR options, and a software engineer to assemble and debug the system.

STATUS OF THE SIGNATURE RECOGNITION TOOLKIT

This description comes mostly from our experiences constructing prototypes in two domains. Initial work has begun on a third domain, and plans are to build several more prototypes or pre-prototypes in order to converge on a toolkit specification and design.

Implemented applications.

The two implemented domains are nearly opposites. One ("EGIL") entails recognizing about fifty types of events of several seconds' duration that occur regularly during the mission. Since unseen (unlabeled) events also occur, the classifiers must include a "none-of-the-above" category—a requirement that makes the recognition task much more challenging. Additional complications occur because events can overlap in time, and noise or loss of signal can obliterate a significant part of the signature. Archival data is plentiful, but assigning labels to this data is an expensive, manual process.

The other application, Guidance, Navigation, and Control (GNC), distinguishes normal from abnormal signatures in order to help controllers decide whether the onboard guidance components are functioning normally. Events last ten minutes or more. Actual (as opposed to simulated) failures are, fortunately, extremely rare, but because of the paucity of data, defining the appropriate level of generalization from sparse training data and estimating the confidence in the classifier are difficult.

Event Detection.

Most of the time the continuous EGIL data stream contains only noise, indicating steady-state loads on the onboard devices. By experimentation, we learned that we could identify most device activations by differentiating the data stream and thresholding the result. This method usually flags events in such a way that the signatures appear at a predictable offset in the time window; thus the pattern recognition modules do not need to resolve translational ambiguities. Another kind of translational ambiguity is removed by subtracting an average initial load value from the samples passed to the pattern recognition modules. The pattern recognizers, therefore, see only the load associated with the device that triggers the event, without the quiescent (DC) load due to other devices on the same bus. One other critical piece of information extracted by the event detector is which of the three phases on the electrical bus are active. This information separates the signature classes into single-phase and multi-phase classes, making subsequent discrimination easier.

Data Management.

When managing our training data became a major headache, we built a DW tool using an off-the-shelf indexed-file component (GDBM) and an interpretive X-Windows-based graphical interface (TCL/TK). The DW runs on Unix workstations, supports data visualization, classification, and formatting, and is soon to be extended for use with post-flight analysis.

System Architecture.

The two applications are running on several flavors of Unix workstations and interact with the controllers by means of an X Windows/Motif interface. All original code is written in C. Whereas quite a few software modules are applicable to more than one application, they may be used in different contexts. For example, filters to remove bursty noise spikes prior to processing the data stream are used in both the EGIL and GNC applications, but they are not invoked by the same modules nor are they invoked in quite the same way. In order to reuse such modules in multiple applications, we developed an efficient "plug-in" interface to replace hard-coded connections between modules.

Each module (data acquisition, spike filter, FFT, event detector, etc.) is written to conform to a plug-in interface. Plug-in services include initialization, termination, data distribution, and timing. When a module is provided with data via the data distribution interface, it operates on that data and then can request that the plug-in controller pass output products to the module's recipients. The connections between processors and recipients are made separately from the modules in a dataflow module. The dataflow modules are presently hand-coded in C; future versions of the toolbox, however, will provide the ability to graphically select and connect modules.

Pattern Recognition.

We have experimented with a variety of pattern-recognition algorithms in order to build a library of PR modules. The NETS package (developed by the Software Technology Branch at JSC [1] has been successful for building feed-forward neural network classifiers. *Ad hoc* network architectures have also been used with success, notably a basis-function network combined with principle-components projection that strongly localizes the set of active function nodes [4]. Our experiences, positive and negative, with network classifiers are in concurrence with those documented by others, e.g., [3].

We have also implemented a more conventional statistical classifier that first extracts features from the events and then applies a Bayesian discriminant calculated from these feature values. Since feature extraction is usually a tricky, manual process, we worried about how feature-based classifiers might be used in an automated environment. In response we developed a method for automating the feature-extraction process based on a genetic algorithm. The features constructed by the algorithm can be used with any classifier method, including networks and decision trees [2]. With the addition of Fourier and wavelet transforms, nearest-neighbor and local-linear models, our repository of pattern classification techniques is growing rapidly.

User Interface and Configuration Builders.

Currently each application interacts with the users via an X-Windows/Motif interface. Work remains to be done on a user-definable interface builder tool and a system configuration tool, but a consensus is developing on what such an interface should include. For example, the Mission-Control venue requires that the flight controllers have a very high confidence in the correctness of the application's outputs. The user interface bolsters this confidence by making available on the display both the signature waveform and the system classifications. Controllers can, therefore, correct an occasional incorrect diagnosis and at the same time develop confidence in the accuracy of the system.

SUMMARY AND FUTURE PLANS

The results of our work to date on the Automated Signature Recognition Toolkit present a number of avenues for future work. One important direction is to continue development of specific user applications which contain the core pattern recognition tool set. As designed, multiple end-user applications should be easily created from a common system architecture, revolving around plug-in pattern recognition modules. Each end-user application will utilize pattern recognition techniques tailored to the signals or patterns for that particular console domain. New console areas will be added on a regular basis until all Mission Control Center positions with relevant data have been evaluated.

Another important direction for this work is to provide a well defined, categorized database of patterns for evaluation and testing of various algorithms. In the process of preparing the existing tools and evaluating their performance during Shuttle missions, we have gathered and classified a large amount of real-world data that is available offline for testing and comparing classification algorithms.

Finally, future challenges include the integration of expert rules with statistical pattern analysis and utilizing regularities in the temporal sequence of signature events.

References

[1] Paul T. Baffes, Robert O. Shelton, and Todd A. Phillips. NETS user's guide (version 3.0). Technical Report JSC-23366, Lyndon B. Johnson Space Center, Houston, TX, 1991.

[2] Philip Laird and Ronald Saul. Automated feature extraction for supervised learning. In *Proceedings of the IEEE Conference on Evolutionary Computation*, New York, 1994. IEEE Press.

[3] Justin D. Paola and Robert A. Schowengerdt. A review and analysis of neural networks for classification of remotely sensed multispectral imagery. Technical Report 93.05, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffett Field, CA, 1993.

[4] Robert O. Shelton. Pattern recognition tool kit for strip chart signature analysis. In *Proceedings of the First Department of Energy workshop on Applications of Neural Networks in Materials Science*, 1994.

N95- 23683

Attention Focusing and Anomaly Detection in Systems Monitoring

Richard J. Doyle
Artificial Intelligence Group
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109-8099
Tel: (818) 306-6149, Fax: (818) 306-6912
rdoyle@aig.jpl.nasa.gov

KEY WORDS: monitoring, anomaly detection

Abstract

Any attempt to introduce automation into the monitoring of complex physical systems must start from a robust anomaly detection capability. This task is far from straightforward, for a single definition of what constitutes an anomaly is difficult to come by. In addition, to make the monitoring process efficient, and to avoid the potential for information overload on human operators, attention focusing must also be addressed. When an anomaly occurs, more often than not several sensors are affected, and the partially redundant information they provide can be confusing, particularly in a crisis situation where a response is needed quickly.

The focus of this paper is a new technique for attention focusing. The technique involves reasoning about the distance between two frequency distributions, and is used to detect both anomalous system parameters and "broken" causal dependencies. These two forms of information together isolate the locus of anomalous behavior in the system being monitored.

1 Introduction

Mission Operations personnel at NASA have the task of determining, from moment to moment, whether a space platform is exhibiting behavior which is in any way anomalous, which could disrupt the operation of the platform, and in the worst case, could represent a loss of ability to achieve mission goals. A traditional technique for assisting mission operators in space-platform health analysis is the establishment of alarm thresholds for sensors, typically indexed by operating mode, which summarize which ranges of sensor values imply the existence of anomalies. Another established technique for anomaly detection is the comparison of predicted values from a simulation to actual values received in telemetry. However, experienced mission operators reason about more than just alarm threshold crossings and discrepancies between predicted and actual sensor values: they may ask whether a sensor is behaving differently than it has in the past, or whether a single behavior is resulting in—the particular bane of operators—a rapidly developing alarm sequence.

Our approach to introducing automation into real-time systems monitoring is based on two observations: 1) mission operators employ multiple methods for recognizing anomalies, and 2) mission operators do not and should not interpret all sensor data all of the time. We seek an approach for determining from moment to moment which of the available sensor data is most informative about the presence of anomalies occurring within a system. The work reported here extends the anomaly detection capability in the SELMON monitoring system [2, 3] by adding an attention focusing capability. This work complements other work within NASA on empirical and model-based methods for fault diagnosis of aerospace platforms [4, 5].

2 Background: The SELMON Approach

Abnormal behavior is always defined as some kind of departure from normal behavior. Unfortunately, there appears to be no single, crisp definition of "normal" behavior. In the traditional monitoring technique of limit-sensing, normal behavior is predefined by nominal value ranges for sensors. A fundamental limitation of this approach is the lack of sensitivity to context. In the other traditional monitoring technique of discrepancy detection, normal behavior is obtained by simulating a model of the system being monitored. This approach, while avoiding the insensitivity to context of the limit-sensing approach, has its own limitations. The approach is only as good as the system model. It can be difficult to distinguish genuine anomalies from errors in the model.

Noting the limitations of the existing monitoring techniques, we have developed an approach to monitoring which is designed to make the anomaly detection process more robust, i.e., to reduce the number of undetected anomalies. Towards this end, we introduce *multiple* anomaly models, each employing a different notion of "normal" behavior.

2.1 Anomaly Detection Methods

In this section, we briefly describe some of the methods that we use to determine when a sensor is reporting anomalous behavior. These measures use knowledge about each individual sensor, without knowledge of any relations among sensors.

Surprise

An appealing way to assess whether current behavior is anomalous or not is via comparison to past behavior. This is the essence of the *surprise* measure. It is designed to highlight a sensor which behaves other than it has historically. Specifically, *surprise* uses the historical frequency distribution for the sensor in two ways: To determine the likelihood of the given current value of the sensor (*unusualness*), and to examine the relative likelihoods of different values of the sensor (*informativeness*). It is those sensors which display unlikely values when other values of the sensor are more likely which get a high *surprise* score. *Surprise* is not high if the only reason a sensor's value is unlikely is that there are many possible values for the sensor, all equally unlikely.

Alarm Anticipation

The *alarm anticipation* measure in SELMON performs a simple form of trend analysis to decide whether or not a sensor is expected to be in alarm in the future. A straightforward curve fit is used to project when the sensor will next cross an alarm threshold, in either direction. A high score means the sensor will soon enter alarm or will remain there. A low score means the sensor will remain in the nominal range or emerge from alarm soon.

Value Change

A change in the value of a sensor may be indicative of an anomaly. In order to better assess such an event, the *value change* measure in SELMON compares a given value change to historical value changes seen on that sensor. The score reported is based on the proportion of previous value changes which were less than the given value change. It is maximum when the given value change is the greatest value change seen to date on that sensor. It is minimum when no value change has occurred in that sensor.

Space limitations preclude describing additional SELMON anomaly measures which reason about individual sensors and about system interactions through the use of a causal model.

2.2 Previous Results

In order to assess whether SELMON increased the robustness of the anomaly detection process, we performed the following experiment: We compared SELMON performance to the performance of the traditional limit-sensing technique in selecting critical sensor subsets specified by a Space Station Environmental Control and Life Support System (ECLSS) domain expert, sensors seen by that expert as useful in understanding episodes of anomalous behavior in actual historical data from ECLSS testbed operations.

The experiment asked the following specific question: How often did SELMON place a "critical" sensor in the top half of its sensor ordering, based on the anomaly detection measures?

The performance of a random sensor selection algorithm would be expected to be about 50%; any particular sensor would appear in the top half of the sensor ordering about half the time. Limit-sensing detected the anomalies 76.3% of the time. SELMON detected the anomalies 95.1% of the time.

These results show SELMON performing considerably better than the traditional practice of limit-sensing. They lend credibility to our premise that the most effective monitoring system is one which incorporates several models of anomalous behavior. Our aim is to offer a more complete, robust set of techniques for anomaly detection to make human operators more effective, or to provide the basis for an automated monitoring capability.

The following is a specific example of the value added of SELMON. During an episode in which the ECLSS pre-heater failed, system pressure (which normally oscillates within a known range) became stable. This "abnormally normal" behavior is not detected by traditional monitoring methods because the system pressure remains firmly in the nominal range, where limit-sensing fails to trigger. Furthermore, the fluctuating behavior of the sensor is not modeled; the predicted value is an averaged stable value which fails to trigger discrepancy detection.

3 Attention Focusing

A robust anomaly detection capability provides the core for monitoring, but only when this capability is combined with attention focusing does monitoring become both robust *and* efficient. Otherwise, the potential problems of information overload and too many false alarms may defeat the utility of the monitoring system.

Although many anomalies can be detected by applying anomaly models to the behavior reported at individual sensors, monitoring also requires reasoning about interactions occurring in a system and detecting anomalies in behavior reported by several sensors.

The attention focusing technique developed here uses two sources of information: historical data describing nominal system behavior, and causal information describing which pairs of sensors are constrained to be correlated, due to the presence of a dependency. The intuition is that the origin and extent of an anomaly can be determined if the misbehaving system parameters *and* the misbehaving causal dependencies can be identified.

3.1 Two Additional Measures

While SELMON runs, it computes incremental frequency distributions for all sensors being monitored. These frequency distributions can be saved as a method for capturing behavior from any episode of interest. Of particular interest are historical distributions which correspond to nominal system behavior.

To identify an anomalous sensor, we apply a distance measure, defined below, to the frequency distribution which represents recent behavior to the historical frequency distribution representing nominal behavior. We call the measure simply *distance*. To identify a "broken" causal dependency, we first apply the same distance measure to the historical frequency distributions for the cause sensor and the effect sensor. This reference distance is a weak representation of the correlation that exists between the values of the two sensors due to the causal dependency. This reference distance is then compared to the distance between the frequency distributions based on recent data of the same cause sensor and effect sensor. The difference between the reference distance and the recent distance is the measure of the "brokenness" of the causal dependency. We call this measure *causal distance*.

3.2 Some Definitions

Define a distribution D as the vector d_i such that

$$\forall i, 0 \leq d_i \leq 1$$

and

$$\sum_{i=0}^{n-1} d_i = 1$$

For a sensor S, we assume that the range of values for the sensor has been partitioned into n contiguous subranges which exhaust this range. We construct a frequency distribution as a vector D_S of length n, where the value of d_i is the frequency with which S has displayed a value in the ith subrange.

We define two special types of frequency distribution. Let F be the random, or flat distribution where $\forall i, d_i = \frac{1}{n}$. Let S_i be the set of "spike" distributions where $d_i = 1$ and $\forall j \neq i, d_j = 0$.

If our aim was only to compare different frequency distributions of the same sensor, we could use a distance measure which required the number of partitions, or bins, in the two distributions to be equal, and the range of values covered by the distributions to be the same. However, since our aim is to be able to compare the frequency distributions of different sensors, these conditions must be relaxed.

3.3 The Distance Measure

The distance measure is computed by projecting the two distributions into the two-dimensional space $[f, s]$ in polar coordinates and taking the euclidian distance between the projections.

Define the "flatness" component $f(D)$ of a distribution as follows:

$$\sum_{i=0}^{n-1} \frac{1}{2} \left| \frac{1}{n} - d_i \right|$$

This is simply the sum of the bin-by-bin differences between the given distribution and F. Note that $0 \leq f(D) \leq 1$. Also, $f(S_i) \to 1$ as $n \to \infty$.

Define the "spikeness" component $s(D)$ of a distribution as:

$$\sum_{i=0}^{n-1} \phi \frac{i}{n-1} d_i$$

This is simply the centroid value calculation for the distribution. The weighting factor ϕ will be explained in a moment. Once again, $0 \leq s(D) \leq 1$.

Now take $[f, s]$ to be polar coordinates $[r, \theta]$. This maps F to the origin and the S_i to points along an arc on the unit circle. See Figure 1.

Note that we take $\phi = \frac{\pi}{3}$. This choice of ϕ guarantees that $\Delta(S_0, S_{n-1}) = \Delta(F, S_0) = \Delta(F, S_{n-1}) = 1$, and all other distances in the region which is the range of Δ are by inspection ≤ 1.

Insensitivity to the number of bins in the two distributions and the range of values encoded in the distributions is provided by the $[f, s]$ projection function, which abstracts away from these properties of the distributions.

Additional details on desired properties of the distance measure and how they are satisfied by the function Δ may be found in [1].

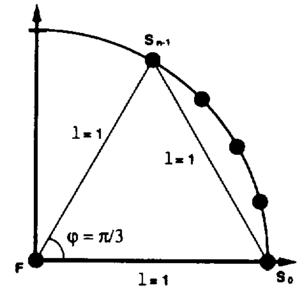


Figure 1: The function $\Delta(D_1, D_2)$.

3.4 Results

In this section, we report on the results of applying the distribution distance measure to the task of focusing attention in monitoring. The distribution distance measure is used to identify misbehaving nodes (*distance*) and arcs (*causal distance*) in the causal graph of the system being monitored, or equivalently, detect and isolate the extent of anomalies in the system being monitored.

Figure 2 shows a causal graph for a portion of the Forward Reactive Control System (FRCS) of the Space Shuttle. SELMON was run on seven episodes describing nominal behavior of the FRCS. The frequency distributions collected during these runs were merged. Reference distances were computed for sensors participating in causal dependencies.

SELMON was then run on 13 different fault episodes, representing faults such as leaks, sensor failures and regulator failures. Due to space limitations, only one of these episodes will be examined here; results were similar for all episodes. In each fault episode, and for each sensor, the distribution distance measure was applied to the incremental frequency distribution collected during the episode and the historical frequency distribution from the merged nominal episodes. These distances were a measure of the "brokenness" of nodes in the causal graph; i.e., instantiations of the *distance* measure.

New distances were computed between the distributions corresponding to sensors participating in causal dependencies. The differences between the new distances and the reference distances for the dependencies were a measure of the "brokenness" of arcs in the causal graph; i.e., instantiations of the *causal distance* measure.

The episode of interest involves a leak affecting the first and second manifolds (jets) on the oxidizer side of the FRCS. The pressures at these two manifolds drop to vapor pressure. The dependency between these pressures and the pressure in the propellant tank is severed because the valve between the propellant tank and the manifolds is closed. Thus there are two anomalous system parameters (the manifold pressures) and two anomalous mechanisms (the agreement between the propellant and manifold pressures when the valve is open).

The *distance* and *causal distance* measures computed for nodes and arcs in the FRCS causal graph reflect this faulty behavior. See Figure 3. (To visualize how the distribution distance measure circumscribes the extent of anomalies, the coloring of nodes and the width of arcs in the figure are correlated with the magnitudes of the associated *distance* and *causal distance* scores, respectively.) The apparent anomaly at the third manifold is due to a known flaw in the training simulator which generated the data. The explanation for the

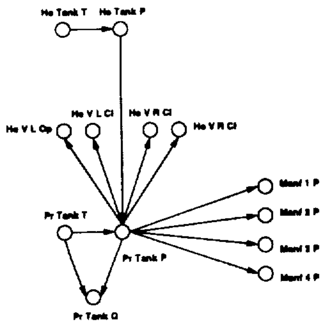


Figure 2: Causal Graph for the Forward Reactive Control System (FRCS) of the Space Shuttle.

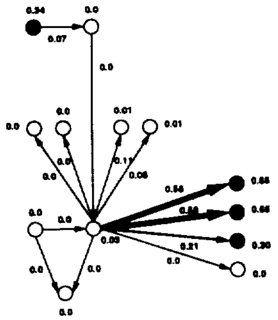


Figure 3: A leak fault.

apparent helium tank temperature anomaly is more interesting: in response to the leak, the valve between the propellant tank and the manifolds closes. The closed system now has a smaller volume, and since the pressure remains the same, temperature must rise according to the ideal gas law. SELMON flags this behavior as anomalous, *even though the relevant causal dependency was not available in the model.* In this case, SELMON helped debug an incomplete model. This helium tank temperature behavior was present in the data for all six leak episodes.

4 Towards Applications

The approach described in this paper has usability advantages over other forms of model-based reasoning. The overhead involved in constructing the causal and behavioral models of the system is minimal. The behavioral model is derived directly from actual data; no off-line modeling is required. The causal model is of the simplest form, describing only the existence of dependencies. For the Shuttle RCS, a 198-node causal graph was constructed in a single one-and-one-half-hour session between the author and the domain expert.

SELMON is being applied at the NASA Johnson Space Center as a monitoring tool for Space Shuttle Operations and Space Station Operations. Early applications include the one for the propulsion (PROP) flight control discipline reported on here, and ones for the thermal (EECOM) and mechanical (MMACS) flight control disciplines. An operational SELMON prototype is available for evaluation by all flight control disciplines, only requiring that a list of sensors "owned" by that discipline be provided.

At the Jet Propulsion Laboratory, we are looking at the problem of onboard downlink determination for the Pluto Fast Flyby project, now in its early design phase. The spacecraft will have limited communications capacity and it will not be possible to transmit all onboard-collected sensor data. Only four hours of coverage from the Deep Space Network will be available per week. The challenge is to devise a method for constructing a suitable summary of a week's worth of sensor data guaranteed to report on any anomalies which occurred. The anomaly detection and attention focusing capabilities of SELMON may be well-suited to this task.

5 Summary

We have described the properties and performance of a distance measure used to identify misbehavior at sensor locations and across mechanisms in a system being monitored. The technique enables the locus of an anomaly to be determined. This attention focusing capability is combined with a previously reported anomaly detection capability in a robust, efficient and informative monitoring system, which is being applied in mission operations at NASA.

6 Acknowledgements

The members of the SELMON team are Len Charest, Dennis DeCoste, Nicolas Rouquette and Jay Wyatt. Harry Porta provided valuable mathematical and counterexample insights during the development of the distance measure. Matt Barry and Charles Robertson provided valuable discussion. Matt Barry also served invaluably as domain expert for the Shuttle FRCS.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] R. Doyle, "A Distance Measure for Anomaly Detection and Attention Focusing in Systems Monitoring," *8th International Workshop on Qualitative Reasoning about Physical Systems*, Nara, Japan, June 1994.

[2] R. Doyle, L. Charest, Jr., N. Rouquette, J. Wyatt, and C. Robertson, "Causal Modeling and Event-driven Simulation for Monitoring of Continuous Systems," *Computers in Aerospace 9*, American Institute of Aeronautics and Astronautics, San Diego, California, October 1993.

[3] R. Doyle, S. Chien, U. Fayyad, and J. Wyatt, "Focused Real-time Systems Monitoring Based on Multiple Anomaly Models," *7th International Workshop on Qualitative Reasoning about Physical Systems*, Eastsound, Washington, May 1993.

[4] T. Hill, W. Morris, and C. Robertson, "Implementing a Real-time Reasoning System for Robust Diagnosis," *6th Annual Workshop on Space Operations Applications and Research*, Houston, Texas, August 1992.

[5] J. Muratore, T. Heindel, T. Murphy, A. Rasmussen, and R. McFarland, "Space Shuttle Telemetry Monitoring by Expert Systems in Mission Control," in *Innovative Applications of Artificial Intelligence*, H. Schorr and A. Rappaport (eds.), AAAI Press, 1989.

Predictability in Spacecraft Propulsion System Anomaly Detection Using Intelligent Neuro-Fuzzy Systems

S. Gulati

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California, USA

THIS PAPER WAS NOT SUBMITTED.

AN EXPERT SYSTEM FOR DIAGNOSING ANOMALIES OF SPACECRAFT

Michael Lauriente, NASA Goddard Space Flight Center, Code 400
Greenbelt, Maryland 20771

Rick Durand and Al Vampola, University Research Foundation
Greenbelt, Maryland 20771

Harry C. Koons and David Gorney, The Aerospace Corporation, P.O. Box 92957
Los Angeles, CA 90009

Telephone (301)286-5690, Fax (301)286-1742
E-Mail Address: lauriente@envnet.gsfc.nasa.gov

N95- 23684

KEY WORDS AND PRASES

Artificial Intelligence, rule-based expert system, spacecraft anomalies, space environment interactions

INTRODUCTION

Although the analysis of anomalous behavior of satellites is difficult because it is a very complex process, it is important to be able to make an accurate assessment in a timely manner when the anomaly is observed. Spacecraft operators may have to take corrective action or to "safe" the spacecraft, space-environment forecasters may have to assess the environmental situation and issue warnings and alerts regarding hazardous conditions, and scientists and engineers may want to gain knowledge for future designs to mitigate the problems. Anomalies can be hardware problems, software errors, environmentally induced, or even the cause of workmanship. Spacecraft anomalies attributable to electrostatic discharges have been known to cause command errors. A goal is to develop an automated system based on this concept to reduce the number of personnel required to operate large programs or missions such as Hubble Space Telescope (HST) and Mission to Planet Earth (MTPE). Although expert systems to detect anomalous behavior of satellites during operations are established, diagnosis of the anomaly is a complex procedure and is a new development.

DESCRIPTION

The tool that is being proposed is a rule-based online expert system for diagnosing in-flight spacecraft anomalies that has the future of simplifying the complex task of analyzing spacecraft anomalies. It uses heuristics in addition to algorithms which allow approximate reasoning and inference and has the ability to attack problems not rigidly defined. The expert system provides scientists with needed risk analysis and confidence not found in the usual programs. The system currently runs on an IBM RISC 6000 at Goddard Space Flight Center (GSFC). The inference engine used is NASA's C Language Integrated Production System (CLIPS).[1,2] A window implementation makes it a more effective tool.

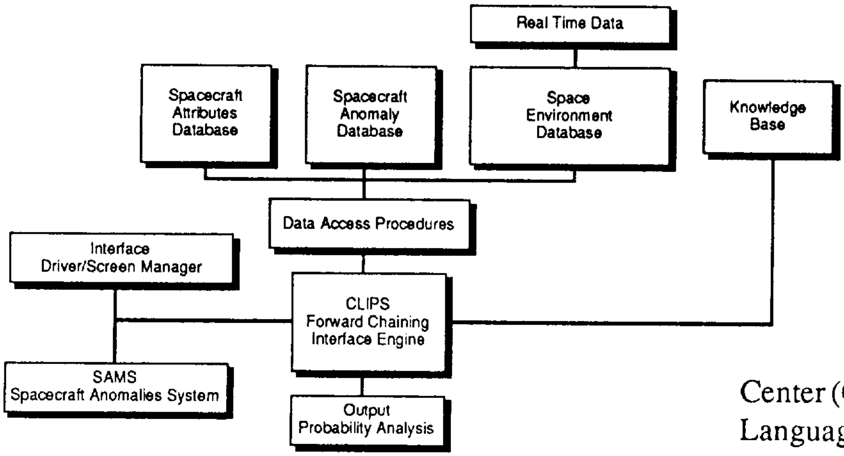


Figure 1. Expert System Architecture.

The architecture of the system is shown in Figure 1. The real time link shown is an option available to

62

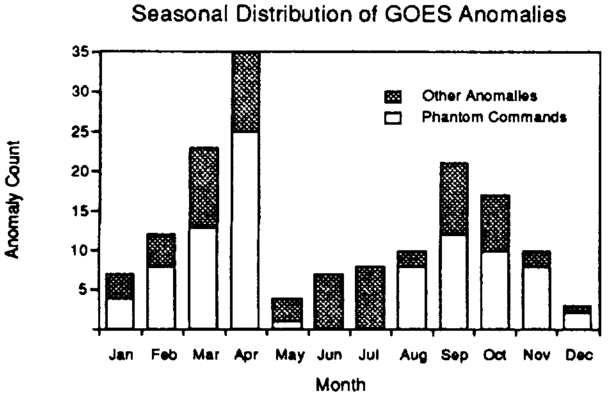


Figure 2. Seasonal distribution of GOES anomalies

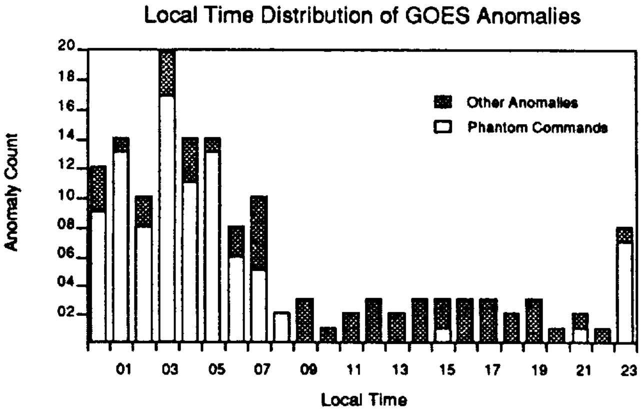


Figure 3. Local-time distribution of GOES anomalies

collect quasi-real time satellite broadcasts by NOAA's Space Environments Services Center (SESC) in Boulder, Colorado. Also available is an option to link up with the interactive space modeling facilities at EnviroNET.[3] The interface driver shown provides a graphing capability. An example is the seasonal distribution of all the GOES spacecraft anomalies as shown in Figure 2, plotted from data in the Spacecraft Anomaly Database using IDL™ graphics.[4] This file was provided by NOAA's National Geophysical Data Center (NGDC). Phantom command anomalies show a bimodal distribution by season. The other anomalies do not. As the phantom commands have been correlated to substorms, it follows that phantom commands also exhibit a seasonal trend.

Figure 3 is a plot of the local time-observed anomalies for the GOES spacecraft. The clustering of phantom commands shows the extent of the particle injection and the subsequent discharging due to high surface potentials.

The block shown as SAMS in Figure 1 was developed by NGDC as a utility to provide a full range of functions for managing, displaying and analyzing data, including functions to examine single anomalies or sets of anomalies for environmental relations. Histograms of local time and seasonal occurrence frequency provided by this utility can reveal distinct patterns for spacecraft which are susceptible to static charge buildup and electrostatic discharge.

Over 300 events are in the database going back to 1971. The contributions to this database were made by cooperation on a world-wide scale and 80 per cent of the spacecraft are at geosynchronous orbit. The four databases shown represent different techniques for storing

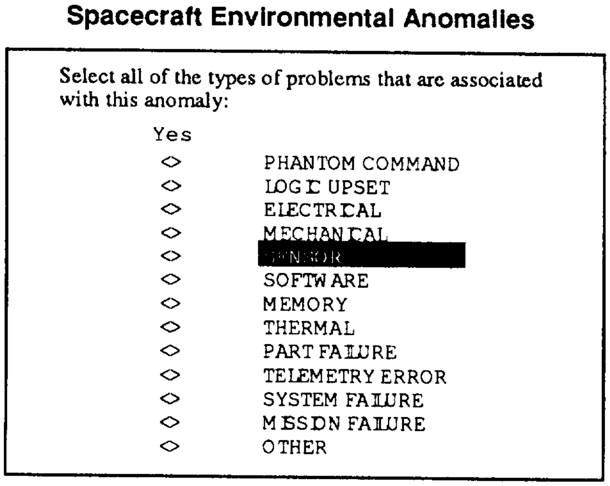
RULE 201
SUBJECT : : BULK CHARGING-RULES
DESCRIPTION : : (recurs when fluence high)
If 1) the recurrence of the anomaly, and
2) the recurrence is of HIGH_PENETRATING-FLUX, and
3) 1) the seven-day accumulated fluence of penetrating electrons is HIGH, or
2) the seven-day accumulated fluence of penetrating electrons is VERY_HIGH,
Then there is suggestive evidence (60%) that the cause of the anomaly is BULK_CHARGING

IF : : (RECURRENCE AND PERIODICITY = OF_HIGH- PENETRATING_FLUXAND
(ACCUM_FLUEN = HIGH OR ACCUM_FLUEN = VERY_HIGH))
THEN : : (CAUSE = BULK_CHARGING CF 60)

RULE 110
SUBJECT : : TOTAL_DOSE-RULES
DESCRIPTION : : (Local time recurrence rules out total radiation dose.)
If 1) the recurrence of the anomaly, and
2) the recurrence of an anomaly in a specific local-time sector,
Then it is definite (100%) that the cause of the anomaly is not TOTAL_DOSE.

IF : : (RECURRENCE AND LT_RECUR
THEN : : (CAUSE ! = TOTAL_DOSE)

Figure 4. Rule Format



Spacecraft Environmental Anomalies

Select all of the types of problems that are associated with this anomaly:

Yes
- ◇ PHANTOM COMMAND
- ◇ LOGIC UPSET
- ◇ ELECTRICAL
- ◇ MECHANICAL
- ◇ SENSOR
- ◇ SOFTWARE
- ◇ MEMORY
- ◇ THERMAL
- ◇ PART FAILURE
- ◇ TELEMETRY ERROR
- ◇ SYSTEM FAILURE
- ◇ MISSION FAILURE
- ◇ OTHER

Figure 5. Expert system query screen

and accessing data. The architecture of the system was designed to emulate the way the user normally looks at data to diagnose anomalies. The expert system not only consolidates expertise in a uniform, objective, and logical way, but it also offers "smart" ways of accessing various databases which are transparent to the user. Then by applying various rules in its knowledge base, the system is queried, as appropriate, to arrive at a conclusion. The current development of the system is able to attribute the causes of satellite anomalies to one of several possible categories, including surface charging, bulk charging, single event upsets (SEUs), and total radiation dose. The architecture is such that other causes could be added if a satisfactory rule base were developed. The rule base includes the expert system rules that will be "fired" under control of the inference engine. The rules are entered in a defined "if-then" format as shown in Figure 4. The user interface links to databases which include past environmental data, satellite data and known anomalies.

The knowledge base consists of over 200 rules and provides links to historical and environmental databases. Unlike its algorithmic predecessors, it can be flexible in the way it attacks complex problems. The system output was verified by referring to historical case studies and historical databases.

The anomaly database is an ASCII file provided by the NGDC which contains information on approximately 300 historical anomalies. Figure 5 is a listing of the types of problems considered for anomalous behavior. The attributes database is an ASCII file for launch and orbital information on satellites as shown in Figure 6 is an abbreviated format. The actual listing has 35 satellites.

The environment database is an ASCII text file of the historical record of the geophysical parameter known as Kp, the planetary magnetic index, used to estimate the severity of magnetic storms within the Earth's magnetosphere. The solar flare database is an ASCII data file on the date and time-of-occurrence of X-class solar x-rays These files are accessed by a C-language interface between the expert system and the ASCII file.

The Attributes Database is an ASCII file for launch and orbital information on satellites. It is possible to anticipate anomalies based on particular orbits. These probable causes have been summarized for classes of orbits in the tutorial paper on spacecraft anomalies.[5] These probable causes are also covered by rules and facts in the Knowledge Base.

FUTURE WORK

The graphical outputs of the Anomaly Database were used as illustrations merely to make the point that these fact resources are readily accessed. They lend to the tool an advantage for analyzing and interpreting large data sets. The development of the engine or driver is considered adequate for the task. The fact base and

NAME	INCLINATION	APOGEE	PERIGEE	LONGITUDE	LAUNCH DATE
SAMPEX	82°	520 km	670 km	-1°	07/03/92
UARS	57.0	579	573	-1	09/15/91
TOMS	82.6	1203	1185	-1	08/15/91
TDRS 5	0.0	35805	35774	-1	08/02/91
TDRS 1	1.4	35798	35785	-1	04/05/83

Figure 6: Launch and orbital information for satellites contained in the database

knowledge base, on the other hand, need to be expanded. The correlation of cause and effects of solar terrestrial effects is a young science. Enough evidence has been collected by NOAA's NGDC that these environmental effects need to be considered as serious.[6] Workshops and special publications that update our knowledge on these environmental interactions should be used as resources for the knowledge base. New frames are also needed. Orbital debris has been recognized as a threat and algorithms exist that are easily accommodated by the expert system. Scintillation related to noisy telemetry links and commanding errors are also candidates to be considered ionospheric.[7] The facility has been improved by the speed of the IBM RISC 6000, and with the use of X Windows, the system will also be enhanced. The Spacecraft Attributes Database does not presently contain information on electrical parts which is certainly an area that needs pursuing.

A new initiative under study is a spin-off expert system for diagnosing anomalies during the early phase of the spacecraft life. The present operation depends on a contingency manual for guidance when anomalies occur. This expert system is an ideal candidate to host a "lessons learned" archive to improve on the facilities now available to ground operators.

Wilkinson has found a solar cycle dependence for SEUs on the Tracking and Data Relay Satellite (TDRS-1),which are caused by cosmic rays.[6] Anomalously high rates of SEUs were correlated with solar flares. We are now collecting SEU data from the Total Ozone Mapping Spectrometer (TOMS) and the Solar, Anomalous, and Magnetospheric Particle Explorer (SAMPEX) satellites in a cooperative effort with NOAA. According to NOAA's Joe Allen, every satellite is a potential monitor of the space environment.[8] By continuing to study the SEUs of spacecraft in different orbits, we hope to get a better understanding of the anomalous behavior of spacecraft for incorporation into the rules of the expert system.

The incorporation of real-time or near real-time data would permit a much more efficient resolution of the causes of satellite anomalies. For this to be achieved, major interagency cooperation will be needed. A long range goal is to reduce the number of personnel needed to monitor and control the large NASA missions. The present development can be incorporated as a baseline for subsystem for ground operators. The implementation of this concept will hold great promise for reducing cost of operations throughout NASA.

ACKNOWLEDGMENTS

We have been fortunate enough to receive outstanding support through a cooperative agreement with the University Research Foundation by Chris O'Callaghan, John Ruby, Tom Wehrung, Mark Rolincik, and Peter Messore, who are students. We are indebted to the staff of NASA Johnson Space Center's AI Laboratory for their cooperation in the use of CLIPS. The technical discussions with Dan Wilkinson and Dr. Joe H. Allen at NOAA were very helpful.

REFERENCES

[1] Giarratano, J. C. (1989, May). "Artificial Intelligence Section," CLIPS User's Guide, Version 4.3 of CLIPS, Lyndon B. Johnson Space Center.

[2] Rolinick, Mark; Lauriente, Michael; Koons, Harry, C.; Gorney, David; "An On-Line Expert System For Diagnosing Environmentally Induced Spacecraft Anomalies Using Clips," NASA Conference Publication 3187, SOAR '92, Vol.1.

[3] Lauriente, Michael, "An On-Line Spacecraft Environment Interactions Information System, J. Spacecraft and Rockets, 31, No. 2, Mar- Apr 1994.

[4] Wilkinson, Daniel C., "National Oceanic and Atmospheric Administration's Spacecraft Anomaly Database and Examples of Solar Activity Affecting Spacecraft," J. Spacecraft and Rockets, 31. No. 2, Mar-Apr 1994.

[5] Vampola, A.L. "Environmentally Included Spacecraft Anomaly Analysis," Spacecraft and Rockets, 31, No. 2, Mar-Apr 1994

[6] Wilkinson, Daniel C., TDRS-1 Single Event Upsets And The Effect Of The Space Environment," IEEE Transactions on Nuclear Science, Vol. 38, No. 6, Dec., 1991.

[7] Koons, H. C., and Gorney, D. J., "Spacecraft Environmental Anomalies Expert System," J. Spacecraft and Rockets, 31. No.2, Mar-Apr 1994.

[8] Allen, Joe, Private Communication, May 1989

Distributed Intelligence for Ground/Space Systems

N95-23685

Mads Aarup &
Klaus Heje Munch
CRI Space
Denmark
ph: +45 4582 2100
fax: +45 4581 3217
maa@nov.cri.dk
khm@spd.cri.dk

Joachim Fuchs
ESA/ESTEC/WGS
The Netherlands
ph +31 1719 8 5296
fax: +31 1719 8 5419
joachim@wgs.estec.esa.nl

Ralf Hartmann
Dornier
Germany
ph: +49 7545 8 9374
fax: +49 7545 8 4411
hartma@spacediv.dofn.de

Tim Baud
Cray Systems
United Kingdom
ph: +44 272 277854
fax: +44 272 290917
baud@craysys.co.uk

INTRODUCTION

DI is short for Distributed Intelligence for Ground/Space Systems and the DI Study is one in a series of ESA projects concerned with the development of new concepts and architectures for future autonomous spacecraft systems. The kick-off of DI was in January 1994 and the planned duration is three years. The total budget is 600,000 ESA Accounting Units corresponding to approximately $720,000.

Problem Definition

The background of DI is the desire to design future ground/space systems with a higher degree of autonomy than seen in today's missions. The aim of introducing autonomy in spacecraft systems is to:

- lift the role of the spacecraft operators from routine work and basic trouble-shooting to supervision,
- ease access to and increase availability of spacecraft resources,
- carry out basic mission planning for users,
- enable missions which have not yet been feasible due to eg. propagation delays, insufficient ground station coverage etc,
- possibly reduce mission cost.

Project Description

The study serves to identify the feasibility of using state-of-the-art technologies in the area of planning, scheduling, fault detection using model-based diagnosis and knowledge processing to obtain a higher level of autonomy in ground/space systems.

A demonstration of these technologies will be developed in the form of a prototype to run in a laboratory environment for the purpose of evaluating future ground/space system designs, and to experiment with the distribution of functionalities of the autonomous architecture between the ground and space segment. DI will use the ERS-1 earth observation mission as the reference mission for the study.

Consortium

The DI Study is carried out for the System Simulation Section of ESA's Technology Center ESTEC by a consortium, led by CRI, and backed by Cray Systems and Dornier.

CRI has a background in the development of ground control systems, planning/scheduling and simulation, combined with spacecraft operations support in the area of flight dynamics. CRI has applied knowledge-based techniques for ESA/ESTEC and ESA/ESOC to mission planning, flight operations, and failure detection, diagnosis and repair. CRI is head of an industrial Consortium developing the Ørsted Scientific Micro Satellite, with direct responsibility for AIV and mission planning, space and ground segment and operations. Ørsted will be launched by a Delta Launcher early 1996.

Cray Systems has developed simulators for most ESA missions, including ERS-1. Also, Cray has substantial experience in the development of control centers and mission planning. Cray has been a main player in the development of the ERS-1 Control Center, and has designed and implemented the operational ERS-1 mission planning system for ESA's Operations Center ESOC.

Dornier was prime contractor for the ERS-1 industrial consortium, and has played a lead role in numerous other spacecrafts, providing solid spacecraft and ground system engineering experience. Dornier offers extensive experience in the development of flight operations plans, in addition to knowledge-based planning.

REFERENCE MISSION

A suitable reference mission for verification of a distributed knowledge-based ground/space architecture providing autonomy should involve a complex spacecraft in an orbit that is either partly without ground contact or so distant that significant delays are inevitable. A natural choice is to select ERS-1 as the reference mission since:

- ERS-1 is equipped with several scientific instruments with many operational constraints, implying very complex mission planning,
- ERS-1 is in a low polar orbit causing it to be out of ground contact during prolonged periods of time,
- operational experience has been gained, making it possible to qualify advantages of autonomy and AI.

Furthermore, the ERS-1 systems engineering expertise and the ERS-1 simulator is available in the DI consortium.

APPROACH

The DI study is divided into two phases. In phase I, we have taken the rather provocative liberty to simply consider the ground and space segment as one combined system. This allows focusing on the essential user requirements on the overall system and on the interaction of the various modules of the system. In the phase I mock-up, the following software will be reused:

- The goal-oriented planning module of Dornier's TINA planner,
- The Optimum-AIV scheduling kernel that CRI previously extended with ERS-1-like subsystem models for the GMPT prototype,
- Cray Systems' operational ERS-1 simulator (for simulating all aspects of the spacecraft behavior),

Furthermore, several ideas from the faults diagnosis and constraints generation module of CRI's EOA (Expert Operator's Associate) may be re-used for the fault diagnosis and repair part of the mock-up.

In phase II, the focus will be concentrated on the distribution aspects of the ground and space segments taking into account issues of distributed artificial intelligence. The development of the distributed phase II prototype will further improve the integrated software tools of the phase I mock-up enabling the evaluation and demonstration of benefits.

ARCHITECTURE

The phase I architecture is based on a hierarchical, object oriented approach providing basis for re-use of existing software modules and ease of final distribution of functionality between the ground and the space segment in phase II.

An overview of the architecture is shown in Figure 1.

Selected data/knowledge structures and modules shown in the architecture are briefly described in the following.

Data/Knowledge Structures

<u>User Requests</u> describe either experiments or spacecraft maintenance operations, and

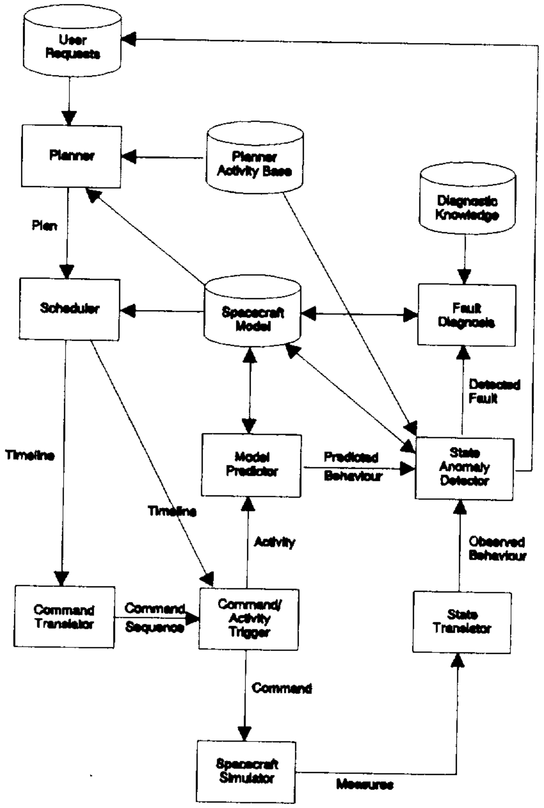


Figure 1: Functional Architecture

are defined by a number of attributes e.g. instrument to use, execution time, orbit position, priority, etc. The formulation of a user request does not require knowledge of the low-level activities necessary to accomplish the request.

Planner Activity Base contains definitions of low level activities to be used for achieving user requests. An activity is defined by:

- preconditions necessary to start the activity,
- resources necessary to carry out the activity (used during scheduling), and
- changes which the activity applies compared to its initial state, e.g. concerning resource availabilities or auxiliary constraints.

Spacecraft Model contains various types of information about the spacecraft used for:

- the prediction of spacecraft behavior,
- the comparison between predicted and observed behavior of the spacecraft (and thereby the fault detection), and
- the diagnosis of a detected fault, e.g. an unexpected component state change or a change of available resources.

The model includes static knowledge about the structure and behavior of the spacecraft and its subsystems, and dynamic knowledge about the current state of the spacecraft. The static knowledge facilitates the reasoning about behavior of the spacecraft as a response to activities, and the generation of diagnosis hypotheses on defective components based on discrepancies in predicted and observed behavior. The dynamic knowledge which is maintained by the model predictor includes such information as resource availabilities (electrical power, data storage capacity, etc.), and descriptions of all anomalies identified by the fault diagnosis module. The model is an abstraction of the spacecraft and the corresponding spacecraft model used in the ERS-1 simulator. It will consist of a subset of the real spacecraft such that it is self-contained with little or no reliance on un-modelled functions. Furthermore, the reasoning about the behavior for the spacecraft will be on the level of activities/predicted behavior rather than the lower command/measures level of the spacecraft simulator.

Diagnostic Knowledge contains an abstraction of relevant experience from satellite designers, manufacturers and operators used for diagnosing faults. This knowledge, expressed as a number of heuristics, can be used either for postulating a priori diagnosis hypotheses or for focussing a systematic model-based diagnosis.

Modules

Planner defines a plan for achieving a number of user requests, i.e. selects and arranges a number of low-level activities defined in the planner activity base such that the execution of the activities will achieve the requests. The planner must take into account the actual state of the spacecraft model. Replanning is invoked if either the user requests are changed or the spacecraft model is updated as a result of fault diagnosis. The planning process is goal-driven based on backward chaining with backtracking.

Scheduler produces a timeline of the activities generated by the planner. The timeline defines the starting time and duration of all activities. The scheduler is initiated each time a new plan has been generated or some resource availability has changed due to a failure. It interfaces the spacecraft model for retrieving constraints used in the scheduling process, e.g.:

- resource constraints on requests made by the activities,
- temporal constraints on predefined fuzzy times due to orbit position or target visibility and to the duration of activities,
- system state constraints on configuration and platform maintenance.

Model Predictor generates expected behavior of the spacecraft based on the spacecraft model as a response to commands. The model predictor applies forward chaining for reasoning about the behavior. It updates the changing states and modes of the subsystems in the model.

State Anomaly Detector (or fault detector) identifies faults based on:

- the observed behavior being an abstraction of the measures derived from the spacecraft simulator,
- the predicted behavior derived from the spacecraft model by the model predictor,
- the definition of activities in the Planner Activity Base for verifying post-conditions associated to activities,
- constraints defined in the spacecraft model some of which depend on the actual state of the spacecraft subsystems.

The fault detection enables the autonomous system to detect such faults as:

- hardware or software errors where the predicted behavior of the spacecraft is inconsistent with the observed behavior,
- errors where the current state of the spacecraft is inconsistent with verification parameters or constraints defined in the model, e.g. due to a wrong time-tag in a manually up-linked command sequence.

Having detected a fault, the fault detection triggers the fault diagnosis module.

Fault Diagnosis generates hypotheses explaining a detected fault. The most important method to be applied for fault diagnosis is model-based diagnosis using the spacecraft model for generating hypotheses about abnormal subsystems or components explaining the fault.

The result of the fault diagnosis is an update of the spacecraft model in case the analysis derived an anomaly, e.g. that a spacecraft status or constraint have changed in an unforeseen manner or that a spacecraft resource has changed in an unexpected way. In the former situation, the fault diagnosis module reinvokes the planner as such problems require an update of the logical sequence of activities to be carried out for recovery. In the latter situation, the scheduler is reinvoked for recovery.

CONCLUSION

The current status as of June 1994 is that a Draft User Requirements Document for the phase I prototype has been produced and the ERS-1 mission demonstration scenarios have been described. The prototype mock-up development has just begun with a clarification of the general MMI strategy.

N95- 23686

Learning Time Series for Intelligent Monitoring *

Stefanos Manganaris **Doug Fisher**

Dept. of Computer Science
Vanderbilt University
Box 1679, Station B
Nashville, TN 37235, U.S.A.
Tel. (615)343-4111, Fax (615)343-8006
E-mail: `stefanos@vuse.vanderbilt.edu`

KEY WORDS AND PHRASES

Bayesian learning, minimum description length, monitoring, time series data.

ABSTRACT

We address the problem of classifying time series according to their morphological features in the time domain. In a supervised machine-learning framework, we induce a classification procedure from a set of preclassified examples. For each class, we infer a model that captures its morphological features, using Bayesian model induction and the minimum message length approach to assign priors. In the performance task, we classify a time series in one of the learned classes when there is enough evidence to support that decision. Time series with sufficiently novel features, belonging to classes not present in the training set, are recognized as such. We report results from experiments in a monitoring domain of interest to NASA.

INTRODUCTION

Performance improvement in classification tasks has been a traditional area of machine learning. The objects to be classified are usually described by time-invariant attribute values. Our research is motivated by applications in temporal and sequential domains. In such domains, an object's properties often vary with time; objects are described by a *time series of values* for each attribute.

This paper focuses on learning to classify time series based on the morphological features of their behavior over time (i.e., the shape of their plots). We study univariate time series, where each object is described by one time-varying attribute. The term signature will be used synonymously with the term univariate time series.

INDUCTION OF CLASS MODELS AND CLASSIFICATION

A set of preclassified signatures (the training examples) are presented to the learner simultaneously. Given that signatures in the same class share morphological characteristics, we design a learner that infers *class models*, represented by functions of time, that capture them. Functions in the space we consider can be decomposed into a set of polynomials and intervals, with one polynomial per interval. For example, Figure 1 shows a signature and the class model induced from it. We use a Bayesian model induction technique to find

---

*This research has been supported by a grant from NASA Ames (NAG 2-834).

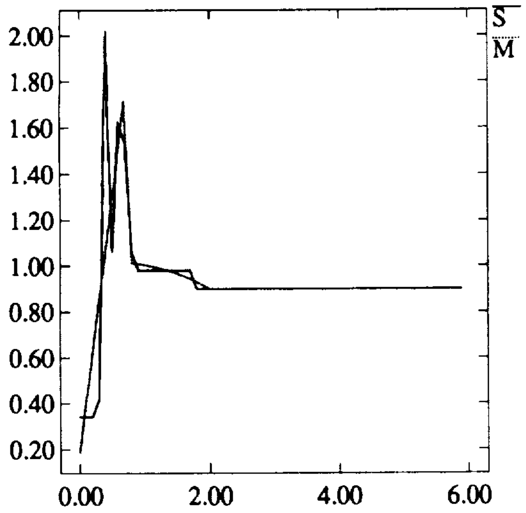


Figure 1: A signature (S) and the class model induced from it (M).

the function best supported by the training data [1]. For each class we search for the model M with maximum posterior probability in light of prior information I and training data D.

$$P(M|D,I) = P(M|I)\,\frac{P(D|M,I)}{P(D|I)} \qquad (1)$$

To assign priors, $P(M|I)$, we use the minimum message length approach [5, 6]. The negative logarithm of the prior probability of a model, $-\log_2 P(M|I)$, is equal to the theoretical minimum length of a message that describes M in light of prior information I. Similar techniques have been used for surface reconstruction in computer vision [3], and for learning engineering models to support design [4], among other applications.

Class models are parameterized, thus the search for the best model extends in the space of parameters. We use the parameters in [3] and an additional precision parameter. Each class model has a partitioning of the time domain into a sequence of intervals. For a given interval we search through all possible families of parameterized models; we use polynomials of up to degree two, but, the method can be easily generalized. To facilitate probabilistic predictions, we assume a Gaussian noise model and independence of sampling errors. We also assume that the variance of the noise distribution is constant over an interval. For each interval we estimate the coefficients of the polynomial and the variance of the noise that maximize the posterior probability of the model.

After training, given a signature, S, and a set of class models, the goal is to find the model most likely to be correct for the signature in light of the prior knowledge. We treat this as a hypothesis testing problem: for each class, C, we compute the *evidence*, $e(C|D,I)$, that S is an object of the class C [2]:

$$e(C|D,I) = 10\log_{10}\left[\frac{P(C|D,I)}{P(\overline{C}|D,I)}\right] \qquad (2)$$

The probability that S belongs in a class other than C, $P(\overline{C}|D,I)$, is computed from the posterior probabilities of all other classes and from the posterior probability of a special "novel" class. The likelihood of the "novel" class is set to zero when any of the known classes has a non-negligible likelihood. When all known classes have low likelihoods, its likelihood is computed so that it tends to one as the maximum likelihood among the known classes tends to zero. The prior of the "novel" class is set to an arbitrary low value. Under normal circumstances, the "novel" class plays no role in the computation of evidence, because of its very low posterior. Only when all known classes have low posterior probabilities, does the "novel" class become a viable alternative.

A MONITORING APPLICATION

The Electrical Generation and Integrated Loading (EGIL) controllers at NASA monitor telemetry data from the Shuttle to detect various events that take place onboard. Typically, an event is the onset or termination of operation of an electrical device on a power bus. Each event has a signature with a set of distinguished morphological characteristics, based on which the controllers identify them. There are over two hundred different events of interest, making their accurate identification a challenging task.

Signatures are extracted from the telemetry stream whenever a change in one of the currents is detected that exceeds a preset threshold. All signatures have the same duration (6 sec. after the triggering change), and their baselines are normalized by subtracting a suitable DC value.

We have designed a set of experiments to demonstrate the feasibility of automating the classification of EGIL signatures using CALCHAS, a Bayesian induction system for time series data. Here we focus on the effect of training in classification performance. We use the percentage of correctly classified instances as our dependent measure of learning. In our experiments there are ten classes of signatures for ten different events; the average number of signatures per class is about 65. Our current implementation only handles univariate time series. There are many three-dimensional signatures in the EGIL domain; in these cases we ignore two of the phases.

In each run, we train CALCHAS on an equal number of randomly selected signatures from each class. We then evaluate its performance on the remaining signatures. We vary the amount of training by using different training set sizes. The results with training sizes of one and eight are summarized in the *confusion matrix* shown in Table 1. Each entry of the table shows the percentage of test signatures, in the class labeling the row, that were classified by CALCHAS to the class labeling the column. The top row for each class was obtained after training CALCHAS with one signature per class; the bottom row was obtained with training sizes of eight. All percentages are averaged over twenty runs; the standard deviations are shown. For example, with a training set of eight signatures, an average of 74% of the WCS test signatures were correctly classified as WCS, and 1% and 25% were incorrectly classified as RCR and NOVEL, respectively. In general, the matrix diagonal indicates the percentage of correct classifications. Entries corresponding to UN1 and UN3 are for signatures whose actual class was unknown.

Table 1 indicates that increased training results in higher classification accuracies. A notable exception seems to be the GAL class, where training with eight signatures results in significantly lower accuracy than training with one signature. We suspect that GAL is an example of a disjunctive concept: there is more than one pattern of morphological features describing signatures in the class. CALCHAS is currently unable to handle disjunctive concepts; training on multiple patterns for a class results in a confused class model and thus lower classification accuracy.

Beyond the practical advantages of automatic vs. manual monitoring, a Bayesian learning approach offers the following technical advantages. It provides a principled way of discerning the distinguishing features of a signature from measurement noise; it mitigates the problem of overfitting. CALCHAS provides an estimate of the confidence in each classification. When more than one classification is supported by roughly the same evidence, we can recognize this fact and report it, as opposed to making an arbitrary classification. Similarly, we can report when no classification is supported with significant evidence. Signatures with sufficiently novel features, belonging to classes not present in the training set, are recognized as such and are classified as NOVEL; potentially costly classification mistakes are avoided.

Table 1: Classification of EGIL signatures (assumed univariate—see text).

CLASS		PHO	VAC	AWCS	H2O	CAB	PRP	WCS	TPS	RCR	GAL	NOVEL
PHO	1	40±29			1±4				2±7		57±29	
	8	96±5									4±5	
VAC	1		68±32									32±32
	8		93±2									7±2
AWCS	1			92±22	5±22							3±1
	8			96±2								4±2
H2O	1	2±9			98±9							
	8				100±0							
CAB	1					79±17						22±17
	8					90±16						10±16
PRP	1						98±4				2±4	
	8						98±2				2±2	
WCS	1							52±28		1±0		47±28
	8							74±4		1±0		25±4
TPS	1	7±14							76±17	3±5	15±11	
	8	8±7							85±8			7±7
RCR	1						2±0			97±1		
	8						3±0			97±0		
GAL	1	2±1									98±0	
	8	22±40									78±40	
UN1	1	46±10			13±2		12±2		3±2	2±1	22±9	2±0
	8	55±4			12±1		12±3		1±1	3±1	15±7	2±0
UN3	1	9±5			20±4		30±4		8±4	4±1	9±3	20±0
	8	18±2			15±1		29±2		11±2	4±1	3±2	20±0

ACKNOWLEDGMENTS

We would like to thank Phil Laird, Robert Shelton, Ron Saul, Wayne Iba, and Deepak Kulkarni for providing the data we used in the experiments and information on the domain.

REFERENCES

[1] Peter Cheeseman. On finding the most probable model. In J. Shrager and P. Langley, editors, *Computational Models of Discovery and Theory Formation*, chapter 3, pages 73–95. Morgan Kaufmann, 1990.

[2] E. T. Jaynes. Probability theory: The logic of science. Advance fragments of this book are made available by the author. Contact etj@wuphys.wustl.edu, 1993.

[3] Edwin P.D. Pednault. Some experiments in applying inductive inference principles to surface reconstruction. In *Proc. of the Eleventh International Joint Conf. on Artificial Intelligence*, pages 1603–1609, 1989.

[4] R. Bharat Rao and Stephen C-Y. Lu. Learning engineering models with the minimum description length principle. In *Proc. of the Tenth National Conf. on Artificial Intelligence*, pages 717–722, 1992.

[5] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

[6] C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *The Journal of the Royal Statistical Society*, 49(3):252–265, 1987.

N95- 23687

An Operations and Command System for the Extreme Ultraviolet Explorer

Nicola Muscettola **David J. Korsmeyer**
Recom NASA
NASA Ames Research Center
Moffett Field, CA 94035
Tel. (415) 604-4744
Fax: (415) 604-3594
e-mail: mus@ptolemy.arc.nasa.gov

Eric C. Olson **Gary Wong**
Center for EUV Astrophysics
University of California at Berkeley
Berkeley, CA 94720
Tel. (510) 643-5652
Fax: (510) 643-5660
e-mail: ericco@cea.berkeley.edu

KEY WORDS AND PHRASES

Artificial intelligence, automation, diagnosis, monitoring, scheduling.

INTRODUCTION

About 40% of the budget of a scientific spacecraft mission is usually consumed by Mission Operations & Data Analysis (MO&DA) with MO driving these costs. In the current practice, MO is separated from spacecraft design and comes in focus relatively late in the mission life cycle. As a result, spacecraft may be designed that are very difficult to operate. NASA centers have extensive MO expertise but often lessons learned in one mission are not exploited for other parallel or future missions. A significant reduction of MO costs is essential to ensure a continuing and growing access to space for the scientific community.

We are addressing some of these issues with a highly automated payload operations and command system for an existing mission, the Extreme Ultraviolet Explorer (EUVE). EUVE is currently operated jointly by the Goddard Space Flight Center (GSFC), responsible for spacecraft operations, and the Center for Extreme Ultraviolet Astrophysics (CEA) of the University of California, Berkeley, which controls the telescopes and scientific instruments aboard the satellite. The new automated system is being developed by a team including personnel from the NASA Ames Research Center (ARC), the Jet Propulsion Laboratory (JPL) and the Center for EUV Astrophysics (CEA).

An important goal of the project is to provide AI-based technology that can be easily operated by nonspecialists in AI. For example, CEA personnel are experienced with the specific EUVE scheduling problem but not with general scheduling methodologies. Since a dedicated AI expert cannot be supported, it is difficult for them to extend and customize their current scheduling tool within a coherent framework. This situation is typical of the smaller NASA satellites programs.

Another important goal is the reusability of the techniques for other missions. Models of the EUVE spacecraft need to be built both for planning/scheduling and for monitoring. In both cases, our modeling tools allow the assembly of a spacecraft model from separate sub-models of the various spacecraft subsystems. These sub-models are reusable; therefore, building mission operations systems for another small satellite mission will require choosing pre-existing modules, re-parametrizing them with respect to the actual satellite telemetry information, and reassembling them in a new model. We are stressing multi-mission support during the tool's development process. The planning and scheduling tools are also being evaluated by science planning and spacecraft sequencing teams for the Cassini Saturn orbiter's mission.

We briefly describe the EUVE mission and indicate why it is particularly suitable for the task. Then we briefly outline our current work in mission planning/scheduling and spacecraft and instrument health monitoring.

THE EUVE MISSION

NASA's EUVE was launched on June 7, 1992. The satellite's mission included three phases. The first phase was a six month long all-sky survey for sources of extreme ultra-violet (EUV) radiation. This phase was completed in January 1993 and resulted in the detection of more than 400 sources of EUV emission. The second phase occurred simultaneously with the all-sky survey. This involved a deep-survey and spectroscopy of much fainter EUV sources in a narrow band of the sky along the ecliptic. The third phase EUVE mission began on January 21, 1993 and is still underway. During this phase, Guest Observers from around the world are using spectrometers and photometers to investigate EUV sources found during the all-sky and deep surveys. The expected output of this phase are spectroscopic and imaging data for over 100 targets per year.

The nominal completion date for the mission is the end of 1995. However, this does not depend on lack of scientific interest nor on an expected deterioration of the excellent health of the spacecraft. MO activities at GSFC and CEA are very labor intensive and, therefore, costly. For this reason, it is not expected that NASA will be willing to continue supporting the EUVE MO after 1995. Some options are being considered in order to lengthen EUVE's contribution to the astrophysical community [1]. These include: (1) the reduction of operations from 3 to 1 shifts a day as soon as possible and the redirection of savings into developing more automated operations, and (2) transferring complete spacecraft operation to CEA using a robust workstation-based operation system.

The set of tools developed by our project will provide payload health management, real-time science data analysis, trending and classification, and science command planning and scheduling for the extreme ultraviolet telescopes. ARC will also provide advanced data systems support for the ground network and control stations. This enhancement of the EUVE science operations center (ESOC) will make the previous options viable.

From the point of view of the ARC and JPL team, EUVE is an ideal demonstration testbed for various information science and AI technologies. Perhaps the most favorable characteristic is that the spacecraft is currently in flight with a good historical database of operations. Spacecraft systems, constraints, and operational procedures are known. This makes spacecraft modeling easier than for missions still in the design phase. Also, CEA already has experience with the use of AI-based tools for science planning. This experience can be leveraged to facilitate the transition to the new generation of tools that ARC and JPL will provide. Another important aspect is the fact that EUVE is structurally simpler than other more ambitious spacecraft (e.g., HST, Cassini). Therefore it will be easier to apply automation of spacecraft sequencing, monitoring and diagnosis, and data systems management. The experiences gathered with EUVE will build confidence for an aggressive automation of more complex missions.

SPACECRAFT SEQUENCING

To continue operation after 1995 CEA will need to take greater responsibility for the spacecraft command sequencing and uplink process. ARC will support this transition with an integrated planning and scheduling system. Such a system will allow; (1) simplification of sequence validation, since at any stage the system will guarantee satisfaction of spacecraft constraints on the base of a detailed, internal model of the spacecraft; (2) generation of schedules with higher science output, since it will be possible to take advantage of detailed knowledge of spacecraft constraints even in preliminary stages of science planning.

Currently, planning and scheduling are done at EUVE through a mixture of manual procedures, utilities and programs developed around SPIKE [4]. SPIKE is an AI-based scheduling tool originally developed at the Space Telescope Science Institute (STScI) for long-term scheduling of the Hubble Space Telescope. SPIKE is being successfully used in operation for HST and has been applied to other space telescopes. Experience in the use of SPIKE for EUVE operations suggest some features missing in SPIKE but essential in a more useful automated tool. The main problem in using SPIKE has been the difficulty in integrating spacecraft ephemeris calculations into the basic scheduling engine. This is not surprising given SPIKE's original focus on long-term scheduling. For such a task coarse approximations are sufficient (e.g., a fixed percentage of orbit time available for observation over the entire scheduling horizon). However, EUVE's task is eminently *short-term;* and coarse approximations become too inaccurate to be useful (e.g., an accurate calculation of exposure time requires knowledge of exact South Atlantic Anomaly (SAA) traversal times for each orbit). In the cases when a modification of the SPIKE constraints are possible, the work involved requires either mapping new constraints onto the heuristics used by SPIKE or modifying SPIKE's own inference engine. Both the previous tasks require personnel with qualifications that are outside of the reach of a more cost-effective, small satellite MO organization. In the case of CEA, this has led to complement SPIKE *from outside*, with extensive preprocessing routines and a mostly manual observation scheduling process. The mismatch between the long-term scheduling philosophy and the needs of short-term scheduling are likely to become even more severe when CEA will take over the spacecraft command sequencing task.

The scheduling system that ARC is developing is based on HSTS [8], a planning an scheduling framework originally aimed at HST's short-term scheduling problem; in that domain HSTS has demonstrated the ability to build schedules that take into account most of the detailed spacecraft constraints and that can be easily transformed in executable spacecraft command sequences. A major effort has been put into providing easily usable constraint modeling facilities; these will allow a mission sequencing expert to easily express spacecraft constraints even without a deep understanding of the functioning of the underlying scheduling engine. Given the similarity of constraints across spacecraft domains and the modularity of the HSTS modeling framework, it will be easy to reuse model components across several missions. Currently, the multi-mission emphasis is being pursued by providing HSTS's domain modeling language to science planners for the Cassini mission in order for them to model constraints in their domains. As the number and types of constraints in a model increases, it is likely that a single schedule building philosophy (e.g., SPIKE's min-conflicts) will not be sufficient for the task. HSTS will provide an underlying modeling and temporal data base capabilities on which a suitable EUVE scheduler will be assembled from a number of possible scheduling and planning methodologies [8, 9, 3, 6, 2, 7]. Easy schedule visualization and manipulation is an important factor in order to complement and adjust the automatic scheduler's decisions to the needs and wants of EUVE's sequencing operators; we are developing such system in collaboration with Heuristicrats Inc. using DTS's scheduling interface toolkit.

PAYLOAD HEALTH MONITORING

A major area of interest to the ARC, JPL and CEA is the automated monitoring and diagnosis of system failures of both the ground and flight systems of the EUVE. The previous and current work on the Augmented Monitoring and Diagnosis Application (AMDA) system [10] for the Control Center Complex at NASA Johnson Space Center can be applied to the EUVE monitoring and diagnosis. The EUVE spacecraft and EUV instrument controllers face a number of problems in monitoring normal operations, diagnosing potential problems, and developing work-around procedures. These problems include determining the initial failure point, determining degraded operation modes, diagnosing the faults, and providing a range of diagnostic hypotheses. Currently, determining and diagnosing faults is a laborious, time consuming process which is highly dependent upon the expert knowledge of a few people. The research and development effort in the area of automated monitoring and diagnosis will be focused on assisting mission controllers to overcome these problems. The architecture of this system includes fault management techniques which utilize digraph failure models as well as model-based diagnosis and expert systems.

Automated fault diagnosis of the EUVE flight and ground systems requires utilization of modeling techniques that will allow inexpensive and quick diagnosis. The automation of much of the tedious systems analysis performed by the current flight controllers and an overview of the system status will help to reduce the operational requirements for the EUVE. This is especially important during low data gathering swing shifts and should eventually allow the elimination of the two swing shifts, with the automated diagnosis and warning system acting as the primary monitoring agent during those times. This 3-to-1 shift reduction effort was the focus of the ARC/CEA collaboration for the spring and summer of 1994.

The first element of that effort is developing the ESOC software version that actively monitors and detects system anomalies and pages off-duty support personnel based upon the severity of the anomaly. ARC and CEA have developed a new version of the ESOC software for the payload mission operators. This system, called EWORKS/EPAGE is developed in the commercially available software RTworks from Talarian, Inc. and the Sun NetManager. EWORKS performs the payload health monitoring and anomaly detection functions for the EUV telescopes onboard the platform. Initially five subsystems are being monitored for each of the seven telescope detectors. The general health, power, thermal control, high voltage, and command echoes. This first step is to be completed on August 31, 1994.

On September 1, 1994, the second step will begin, a simulated single shift operation. The EWORKS software will be frozen and put into operation for a two month trial period. During this time the ESOC personnel will continue 24 hour shifts. At the end of this period the decision for reduction from three to one shift of operations will be made based upon the feedback from GSFC and the ESOC mission operators. Pending approval the transition to single shift operations is scheduled for November 1, 1994.

ARC will develop system engineering models from the designs and operational parameters of the EUVE spacecraft and instrument components [5]. To develop the EUVE spacecraft systems model, the spacecraft system parameters such as mass, size, operational constraints, avionics, power, communications, thermal system, and instrument systems need to be modeled as separate subsystems. In order to successfully develop each of the subsystem models, we must perform a top-level analysis to adequately parametrize and understand them. The models will be integrated into a complete representational model of the EUVE spacecraft and verified against the operational data. The objective of the small satellite system model is the development of a model which identifies and quantifies the key system characteristics necessary for failure diagnosis and fault tracing. High-fidelity modeling and attention to actual system design are necessary for the model to be used to evaluate the performance of EUVE systems and to develop robust monitoring and diagnosis systems.

Bibliography

[1] Douglas, F.; Bruegman, O., 1994. *Study of Low Cost Operations Concepts for EUVE Mission Operations*, report to Dr. Ronald S. Polidan, Advanced Small Payloads Study, (Internal Document), NASA Goddard Space Flight Center.

[2] Drummond, M.; Bresina, J.; and Swanson, K., 1994. *Just-In-Case Scheduling*, Technical Report, Artificial Intelligence Research Branch, NASA Ames Research Center.

[3] Hansson, O.; and Mayer, A., 1994. DTS: A Decision Theoretic Scheduler for Space Telescope Applications, in *Intelligent Scheduling*, edited by Fox, M. and Zweben, M., Morgan Kauffman.

[4] Johnston, M.D., 1990. SPIKE: AI Scheduling for NASA's Hubble Space Telescope, *Proceedings of the 6th Conference on Artificial Intelligence Applications*, IEEE Computer Society Press.

[5] Korsmeyer, D. J., 1992. A Cislunar Guidance Methodology and Model for Low-Thrust Trajectory Generation. AAS 91-433, H. Jacob (Ed.), *Astrodynamics 1991, Advances in the Astronautical Sciences, Vol. 76,* Univelt Publishers, San Diego, CA.

[6] Lansky, A., 1993.*Localized Planning With Diversified Plan Construction Methods*, Technical Report FIA-93-17, Artificial Intelligence Research Branch, NASA Ames Research Center.

[7] Minton, S.; Johnston, M.D.; Philips, A.B.; and Laird, P., 1992. Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems, *Artificial Intelligence*, 58.

[8] Muscettola, N., 1994. HSTS: Integrating Planning and Scheduling, in *Intelligent Scheduling*, edited by Fox, M. and Zweben, M., Morgan Kauffman.

[9] Muscettola, N., 1993. Scheduling by Iterative Partition of Bottleneck Conflicts, in *Proceedings of the 9th Conference on Artificial Intelligence for Applications*, IEEE Computer Society Press.

[10] Patterson-Hine, F. A., et. al., 1993. DMS Augmented Monitoring and Diagnosis Application (DMS AMDA) Prototype, *Proceedings of the AIAA Computing in Aerospace 9,* San Diego, CA

N95- 23688

Performance Results of Cooperating Expert Systems in a Distributed Real-time Monitoring System

U. M. Schwuttke, J. R. Veregge, and A. G. Quan
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109 U.S.A.
Tel: 818-354-1414 Fax: 818-393-6004
E-mail: ums@puente.jpl.nasa.gov

KEY WORDS AND PHRASES

Automation, distributed systems, expert systems, monitoring and diagnosis, real time.

INTRODUCTION

There are numerous definitions for real-time systems, the most stringent of which involve guaranteeing correct system response within a domain-dependent or situationally defined period of time. For applications such as diagnosis, in which the time required to produce a solution can be non-deterministic, this requirement poses a unique set of challenges in dynamic modification of solution strategy that conforms with maximum possible latencies. However, another definition of real time is relevant in the case of monitoring systems where failure to supply a response in the proper (and often infinitesimal) amount of time allowed does *not* make the solution less useful (or, in the extreme example of a monitoring system responsible for detecting and deflecting enemy missiles, completely irrelevant). This more casual definition involves responding to data at the same rate at which it is produced, and is more appropriate for monitoring applications with softer real-time constraints, such as interplanetary exploration, which results in massive quantities of data transmitted at the speed of light for a number of hours before it even reaches the monitoring system.

The latter definition of real time has been applied to the MARVEL system-[1]-for automated monitoring and diagnosis of spacecraft telemetry. An early version of this system has been in continuous operational use since it was first deployed in 1989 for the Voyager encounter with Neptune. This system remained under incremental development until 1991 and has been under routine maintenance in operations since then, while continuing to serve as an artificial intelligence (AI) testbed in the laboratory. A second-generation Galileo application has been on-line for only one year and is still under active development. The second-generation system builds on experience gained with the earlier embedded diagnosis systems to achieve an order of magnitude increase in processing capability.

The system architecture has been designed to facilitate concurrent and cooperative processing by multiple diagnostic expert systems in a hierarchical organization. The diagnostic modules adhere to concepts of data-driven reasoning, constrained but complete nonoverlapping domains, metaknowledge of global consequences of anomalous data, hierarchical reporting of problems that extend beyond a single domain, and shared responsibility for problems that overlap domains. The system enables efficient diagnosis of complex system failures in real-time environments with high data volumes and moderate failure rates, as indicated by extensive performance measurements.

COOPERATING DIAGNOSIS SYSTEMS IN A DISTRIBUTED ARCHITECTURE

The need for robust mechanisms of cooperation among real-time diagnostic modules has

---

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The authors wish to acknowledge strong support from JPL's Voyager and Galileo Projects, the Multimission Operations Support Office, and Director's Discretionary Fund.

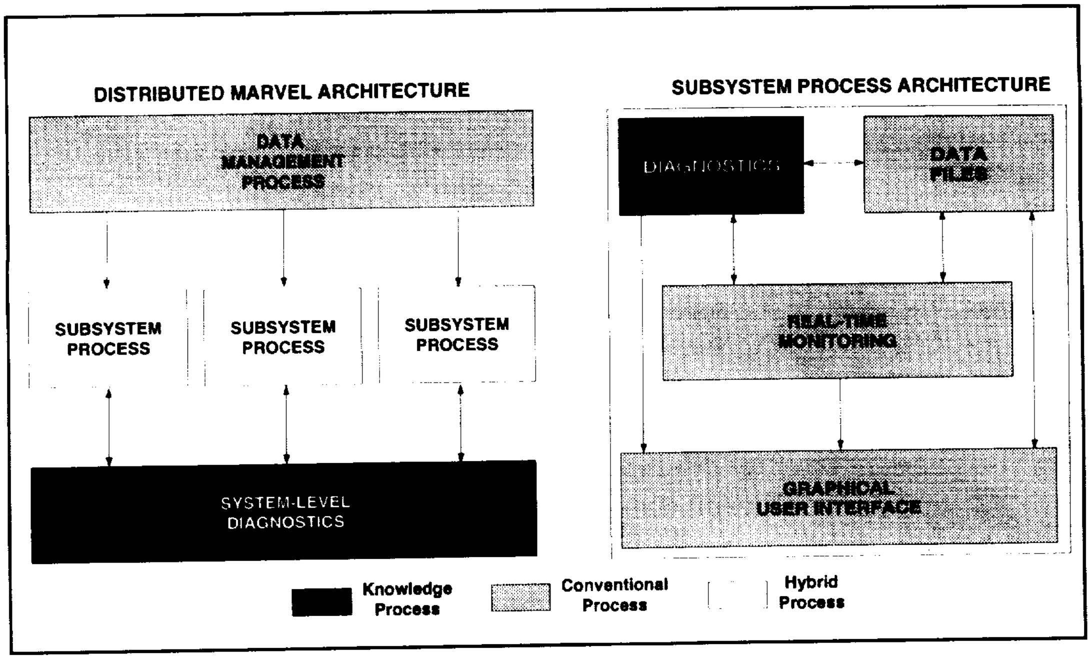


Figure 1. The distributed architecture on the left can currently be configured to run on one to four UNIX workstations. The hybrid subsystem processes on the left are composed of conventional and knowledge processes, as shown in the figure on the right. Knowledge processes are used only when a reasoning capability is explicitly required.

been an important driver of the system architecture. The notion of joint responsibility-[2]-as an alternative to the more conventional notion of agents acting in self-interest-[3], [4]-has been amended with modular problem decomposition and data-driven reasoning in order to minimize the need for communication between agents. The various modules in the distributed architecture of Figure 1 are allocated among a configuration of UNIX workstations. The data management module receives data from a source (in the case of our current application, the data is spacecraft telemetry received from the Jet Propulsion Laboratory's (JPL) ground data system) and allocates it to the appropriate subsystem monitor based on identification of data type. (Our system is partitioned according to the structure of the spacecraft, with one subsystem monitor for every spacecraft subsystem monitored by MARVEL, including command, flight data, attitude and articulation control, and telecommunications; propulsion, thermal, and power have not been addressed.)

Each of the subsystem monitors provides algorithmic functions such as validation of telemetry, detection of anomalies, trend analysis, and automatic reporting. These functions, while not in themselves of interest in AI or computer science research, are vital components of a real-world diagnostic system. In addition, each subsystem process can provide diagnosis of failures based on anomalous data and recommendation of corrective actions. The latter two functions are provided by knowledge-based modules that are embedded within each of the individual subsystem monitors. The remaining modules include the graphical user interface and display processes for each of the subsystem monitors, and the system-level diagnostic agent for handling failures that manifest themselves across multiple subsystems (and therefore cannot be completely analyzed by any one subsystem alone). Detailed reasoning examples that illustrate cooperation among diagnosis modules are presented elsewhere-[5].

EXPERT SYSTEM CHARACTERISTICS

Rule-based diagnostic modules are embedded in efficient algorithmic code. The algorithmic code performs all functions that do not explicitly require reasoning capability, so that the use of the less efficient reasoning modules is limited to those functions for which it is essential.

Forward-chaining demons are used to represent domain knowledge. Reasoning is activated by the appearance of data that requires diagnosis. The initial determination that diagnosis is required is made by algorithmic monitoring code, which detects potential anomalies algorithmically and passes the anomalous data to an appropriate diagnostician. In the absence of anomalous data within its domain, a diagnostic system is idle.

Each diagnostic system is responsible for a small, clearly partitionable domain of expertise. Partitioning is governed by the natural decomposition of the system being diagnosed. This helps overcome disadvantages associated with rule-based systems for which, typically, implementation can be intractable, execution is nondeterministic and relatively slow, and verification can be difficult. Small, modular knowledge bases enable developers to handle more easily definable subproblems. Smaller knowledge bases execute more efficiently, because less time is spent in search. Finally, smaller knowledge bases are easier to verify.

Each diagnostician has sufficient knowledge to be fully accountable for diagnoses within its area and has no knowledge of other domains. This requires that accountability for locally detectable failures must be local. However, the participation of more than one diagnostic system is required when symptoms manifest themselves in more than one domain. Each diagnostic system has the necessary metaknowledge to identify symptoms of failures that could possibly extend beyond its domain. Metaknowledge is contained in a set of rules in each knowledge base, and is associated with the occurrence of events whose analysis may require the cooperation of other agents.

An expert forwards all known information pertaining to failures beyond its domain to another agent at the next higher level in the hierarchy. The underlying approach on forwarded messages is conservative; it is up to the agent receiving the information to determine whether a fault requiring a diagnostic message and an alarm has occurred, or whether the anomalous data has some other explanation. When necessary, metaknowledge is used to direct messages to the relevant agent(s) in order to complete the final analysis of the anomalous data and provide diagnosis of any associated failures.

EXPERIMENTAL RESULTS

The distributed architecture described in this paper has been applied to two generations of real-time monitoring systems. The Galileo system, currently under development, does not yet include on-line modules for diagnosis. The Voyager system, completed in 1991, contains four diagnostic expert systems (developed using a commercial shell) in a two-level hierarchy.

Conventional monitoring modules for four of the spacecraft subsystems were completed: the flight data subsystem, the computer command subsystem, the attitude and articulation control subsystem, and the telecom subsystem. Three of the expert systems are embedded in conventional modules that provide data access/manipulation and monitoring in addition to providing graphical user interfaces and other subsystem-specific automation. The system-level diagnostician is not embedded within another module.

The computer command subsystem (CCS) expert contains on the order of 150 rules, focuses on a relatively broad domain analysis, and is invoked very frequently (for almost every parameter). The attitude and articulation control subsystem (AACS) expert contains approximately 100 rules, and focuses on a more narrow domain of analysis. It is invoked infrequently. The telecom expert system contains on the order of twenty-five rules and is invoked continuously (for every parameter). The flight data subsystem (FDS) module does not contain an expert system.

Experimental evaluation on a network of workstations (Sun Microsystem Sparc LXs running Solaris 2.2) involved a series of tests to determine the maximum number of data parameters that could be processed per module per second (a subsystem module includes both the conventional and knowledge-based components, as shown in Figure 1). The primary purpose of this evaluation was to learn about the performance of the expert systems and apply our insights to future development on the Galileo application. This evaluation was not motivated by a need to improve the performance of the Voyager system, as current data rates are considerably slower than during the planetary encounters and are easily handled by the existing software configuration.

The results are shown in Figure 2. The baseline performance was below expectation, with FDS, CCS, AACS, and Telecom processing 26, 3, 24, and 428 parameters per second respectively, or 481 total parameters per second processed by the entire system. Performance profiling revealed that file input/output (I/O) and the graphical user interfaces (GUIs) rather than the

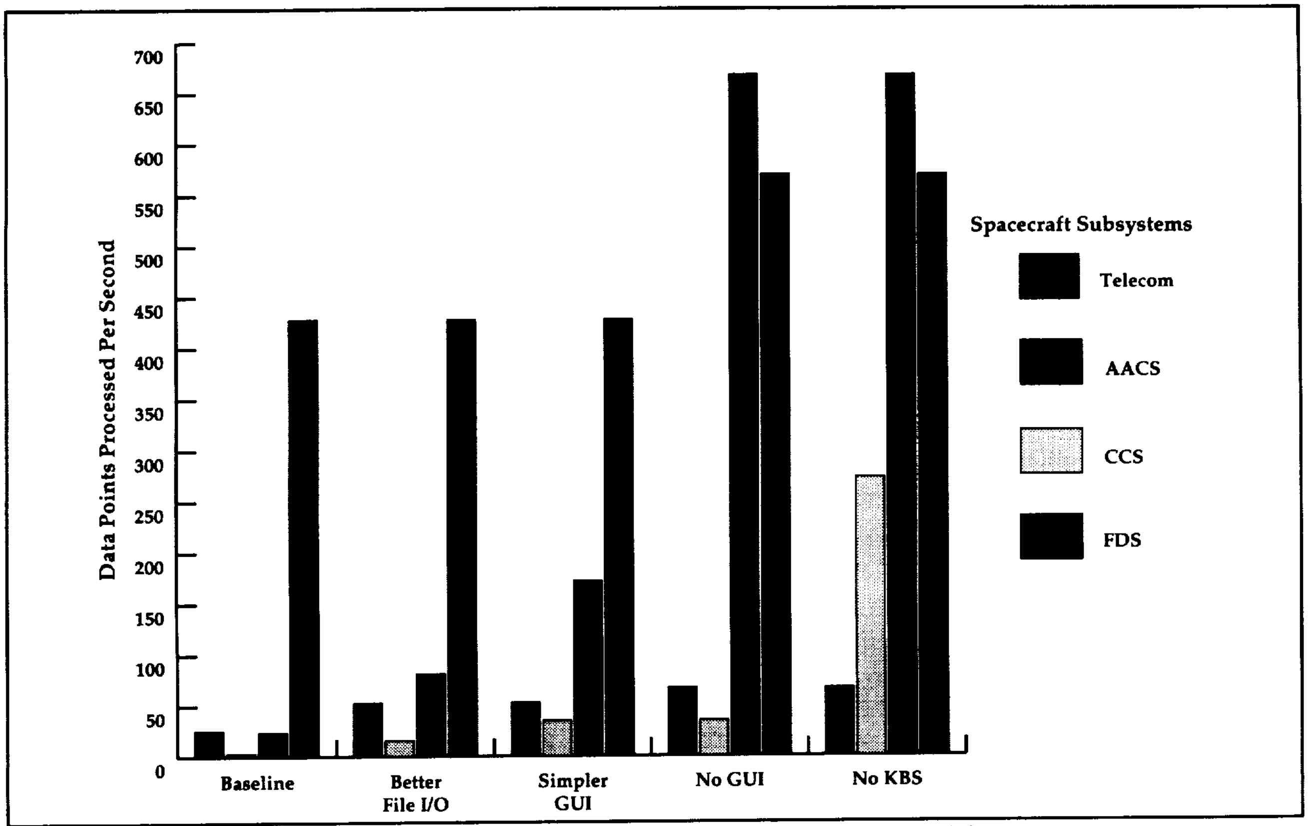


Figure 2. Performance results for each of the subsystem modules.

diagnostic modules were primary performance bottlenecks.

With regard to these bottlenecks, the four modules can be categorized as follows: FDS and CDS have moderately complex GUIs, and perform significant file I/O. AACS has the most complex GUI and performs very little file I/O, because the input files read by this subsystem are sufficiently small that they are read entirely into memory upon system initialization. Telecom has a simple GUI and performs no file I/O.

Optimizing file I/O where possible improved performance to 53, 16, 81, and 428 parameters per second. (This is the only improvement discussed in this section that was carried forward to the operational system.) Simplifying the graphical user interface by eliminating real-time scrolling windows (known to be computationally inefficient in MOTIF user interfaces; considered desirable by end-users and thus included in the FDS, CCS, and AACS modules of the operational system) further improved performance to 53, 35, 172, and 428 parameters per second. Eliminating the graphical user interface entirely resulted in further performance increases to 67, 35, 646, and 570 parameters per second. Finally, eliminating the expert systems yielded performance of 67, 273, 668, and 570 parameters per second.

These results made it possible to gain a number of new insights with regard to our system. The biggest surprise was the high performance of the telecom module. The combination of the small knowledge base and the simple user interface enables processing of 428 parameters per second. Elimination of both the GUI and the expert system only results in a further performance improvement on the order of 25 percent, indicating that no substantial penalty is associated with the significant enhancement to functionality provided by these two components of the module. The next generation of the system will benefit from this result, in that frequently performed analysis that requires the use of an expert system will be implemented with a number of small, cooperating modules rather than one larger module. This in itself is not unexpected; it is the magnitude of the benefit that was surprising. Further performance improvement could likely be gained with a more efficient expert system shell. This will be investigated, although we do not currently expect more than an additional order of magnitude improvement.

C-2

The AACS expert system is larger by a factor of four, and slower, in the worst case, by over two orders of magnitude. This can be explained by a significantly larger search space and greater depth in each search. Performance could likely be improved with a faster reasoning shell and by modularization of the knowledge base. However, the diagnostic component of this module is invoked sufficiently rarely (often less than once per hour) that this is not an important bottleneck. In the case of this type of module, it is preferable to simplify the GUI, which continues to impose considerable resource overhead.

The CCS expert system is large and is invoked regularly as part of ongoing trend analysis in that subsystem module. Elimination of the expert system results in an additional order of magnitude increase in performance, providing further indication that a large knowledge base is inappropriate for frequently invoked real-time diagnosis. The CCS knowledge base is characterized by breadth rather than depth. As a result, it would be both beneficial (and straight-forward) to reduce it to three or more component modules without imposing significant overhead from resulting interprocess communication. (If this were implemented, the CCS module would still be I/O bound, as it reads from a number of very large files.)

As a result of these insights, the Galileo implementation takes a more efficient approach to file I/O. It also tends to be more efficient in its graphical user interface, in that it does not include some of the higher overhead user interface widgets. Such changes impact functionality, requiring a certain amount of negotiation with end users (who are typically willing to compromise in favor of performance). In addition, the Galileo system makes greater use of the distributed architecture with more than one module per subsystem, and more than one diagnostic component per module.

CONCLUSION

The MARVEL distributed architecture demonstrates the successful implementation of multiple cooperating agents in a complex real-time diagnostic system. We have designed an architecture that facilitates concurrent and cooperative processing by multiple agents in a hierarchical organization. These agents adhere to the concepts of data-driven embedded diagnosis, constrained but complete nonoverlapping domains, metaknowledge of global consequences of anomalous data, hierarchical reporting of problems that extend beyond an agent's domain, and shared responsibility for problems that overlap domains.

The MARVEL architecture is simple and well suited for real-time telemetry analysis. Conventional processing is used wherever possible in order to facilitate performance. The knowledge-based agents are embedded within the algorithmic code, and are invoked only when necessary for diagnostic reasoning. Distribution of telemetry monitoring and diagnostic processes across workstations provides significant improvement in performance. These qualities allow for efficient real-time diagnosis of anomalies occurring in a complex application.

Maximum modularization of frequently invoked reasoning modules will enable significant performance improvements in the next generation system.

REFERENCES

[1] Schwuttke, U. M.; Quan, A. G.; Angelino, R.; Childs, C. L.; Veregge, J. R.; Yeung, R.; and Rivera, M. B., 1992. MARVEL: A Distributed Real-time Monitoring and Analysis Application. In *Innovative Applications of Artificial Intelligence 4*, MIT Press.

[2] Jennings, N. R.; and Mamdani, E. H., 1992. Using Joint Responsibility to Coordinate Collaborative Problem Solving in Dynamic Environments. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 269-275. San Jose, California.

[3] Durfee, E. H., 1988. Cooperation through Communication in a Distributed Problem Solving Network. In *Distributed Artificial Intelligence*, Vol. 2. Pitman Publishing.

[4] Cohen, P. R.; Hart, D. M.; and Howe, A. E., 1990. Addressing Real-time Constraints in the Design of Autonomous Agents. *COINS Technical Report 90-06*. University of Massachusetts at Amherst.

[5] Schwuttke, U. M.; and Quan, A. G., 1993. Enhancing Performance of Cooperating Agents in Real-time Diagnostic Systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 332-337, Chambery, France.

SESSION RM

Rovers and Robot Motion Planning

Path Planning for Planetary Rover Using Extended Elevation Map

N95-23689

Ichiro Nakatani, Takashi Kubota, and Tetsuo Yoshimitsu

The Institute of Space and Astronautical Science

3-1-1, Yoshinodai, Sagamihara-shi,
Kanagawa-ken 229, JAPAN
Tel. +81-427-51-3911, Fax. +81-427-51-3988
E-MAIL: nakatani@net202.newslan.isas.ac.jp

ABSTRACT : This paper describes a path planning method for planetary rover to search for a path on planetary surface. Planetary rover is required to travel safely over a long distance for many days in unfamiliar terrain. Hence it is very important how planetary rover processes sensory information to understand the environment and to make decisions. As a new data structure for a map information, an extended elevation map(EEM) has been newly introduced, which includes the effect of the size of the rover. The proposed path planning can be conducted in such a way as if the rover were a point while the size of the rover is automatically taken into account. The validity of the proposed method is verified by computer simulations.

KEY WORDS AND PHRASES

Path planning, planetary rover, elevation map.

INTRODUCTION

In recent years many researchers have extensively studied and developed mobile robots (planetary rovers) for unmanned surface exploration of planets[1][2][3]. Planetary rover is required to travel safely over a long distance for many days in unknown terrain. Due to the communication delay between the earth and the rover, round trip propagation time, and bandwidth limitation, autonomous capability of rover is essential.

One of the important functions for a planetary rover is to plan a path from a start point to a goal without hitting obstacles. Path searching in a structured world with polygonal obstacles has received considerable attention as part of the general problem of robot motion planning, and various algorithms have been proposed [4][5][6][7]. However, there are few outdoor guidance systems that can create a path plan in such unknown and unstructured environment as planetary surface[8][9]. There have also been proposed only few practical path planning methods that consider the size of the robot.

This paper describes a path planning method for planetary rover to search for a path on planetary surface. The model of a rover is introduced to consider the size of planetary rover. This model can be easily modified into any other rover architecture. A planetary rover makes an elevation map by observing the environment. The conventional elevation map was based on the implicit assumption that a rover can be described as a point[10]. We have newly introduced an extended elevation map, which includes the effect of the size of the rover. By using an extended elevation map, path planning can be conducted in such a way as if the rover were a point while the size of the rover is automatically taken into account. The difference of the height in accordance with the different rover orientation is also taken into consideration. The proposed path planning algorithm is based on grid search method.

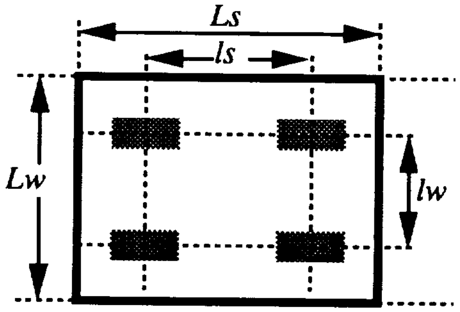
This paper is structured as follows. In 2nd Section, modeling of the planetary rover is discussed. Then a method to make an extended elevation map is explained in 3rd Section. In 4th Section, a path planning algorithm based on extended elevation map is proposed. Computer simulations are given in 5th Section. Final Section is for discussion, conclusion, and future work of the research.

86

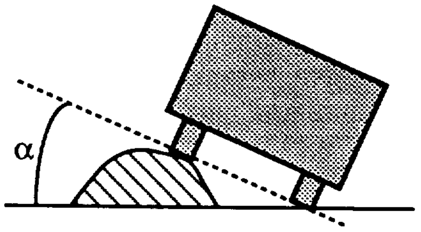
ROVER GEOMETRY

The performance of a rover moving in unstructured environment depends upon the geometry, such as suspension, size, the number of wheels etc. To consider the size of a planetary rover, a model of the rover is introduced as shown in Fig.1. Rover geometry is expressed by three parameters, roll angle criterion, pitch angle criterion, and height criterion.

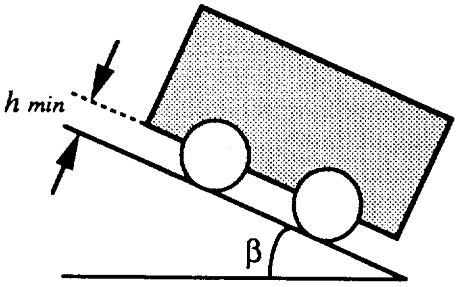
The maximum roll angle $\alpha_{\max}$ is corresponds to the capability to clear obstacles. The maximum pitch angle $\beta_{\max}$ means the maximum angle of inclination for a rover to go over. The height $h_{\min}$ means the minimum distance between the body of a rover and the ground to avoid hitting the ground. Though this model shows the case for a four wheel rover, it is easy to adapt such a modeling to any other rover with different geometry.



(a) rover geometry



(b) roll angle



(c) pitch angle

Figure 1. Model of a rover

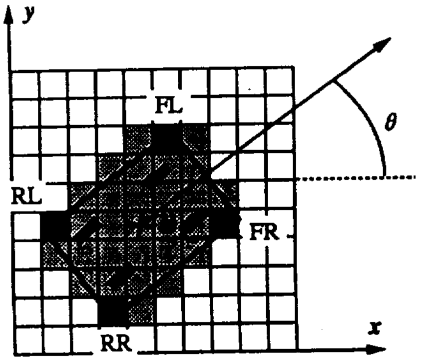
EXTENDED ELEVATION MAP

Map Data Structure

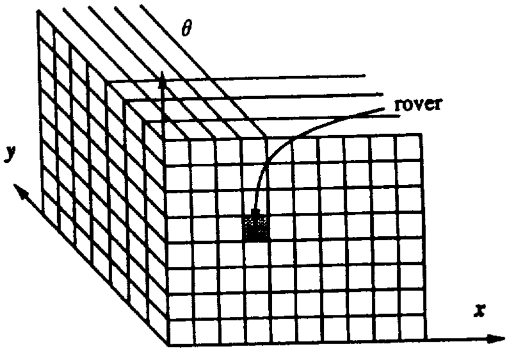
The planetary rover plans a path based on an elevation map which consists of many square grids. Each grid $G(x,y)$ has the height information z of that point (x,y).

$$z = h(G) = h(x,y) \tag{1}$$

An elevation map shows the terrain data in front of the rover detected by a sensor. The conventional elevation map is based on the implicit assumption that a rover can be described as a point. Here a new map concept is proposed to include the effect of the size of the rover as shown in Fig.2. A rover actually occupies some grids on the elevation map. So virtual map with the information on the position and the attitude of a rover is proposed. The authors call this map an Extended Elevation Map (EEM). The information about the traversability is added to each grid on EEM. By using EEM, path planning can be conducted in such a way as if the rover were a point while the size of the rover is automatically taken into account.



(a) occupied grids(elevation map)



(b) extended elevation map

Figure 2. Extended elevation map

Judgment of Traversable Area

Traversable area means the area where a rover can stay stably. Now suppose that the position of four wheels of a rover are FR, FL, RR, and RL respectively. And then the position of the center of gravity and the orientation (azimuth angle) of a rover on EEM is to be expressed by (x,y,θ) respectively. The stability conditions for a rover roll angle and pitch angles are,

$$\frac{|h(\mathrm{FR})-h(\mathrm{FL})|}{l_w} \leq \tan(\alpha_{\max}) \quad (2)$$

$$\frac{|h(\mathrm{RR})-h(\mathrm{RL})|}{l_w} \leq \tan(\alpha_{\max}) \quad (3)$$

$$\frac{|h(\mathrm{FR})-h(\mathrm{RR})|}{l_s} \leq \tan(\beta_{\max}) \quad (4)$$

$$\frac{|h(\mathrm{FL})-h(\mathrm{RL})|}{l_s} \leq \tan(\beta_{\max}) \quad (5).$$

The condition for the rover body to avoid hitting the surface of the ground is

$$h(G_i) < P(G_i) \quad (\forall i=1,\cdots,N) \quad (6),$$

where P denotes the plane constructed by the contact points of the wheels with the ground. N is the number of grids occupied by the rover. If all the conditions from (2) through (6) are satisfied, that area is defined as traversable one. Otherwise such a area means non-traversable one.

The height H of a rover on EEM is expressed as follows.

$$H(x,y,\theta) = \frac{1}{4}\{h(\mathrm{FR})+h(\mathrm{FL})+h(\mathrm{RR})+h(\mathrm{RL})\} \quad (7).$$

PATH PLANNING

Extended grids on EEM are searched in the proposed path planning algorithm. Here suppose that the rover can move in eight kinds of directions $(\theta=\frac{\pi}{8}\cdot j \quad (j=0,\cdots,7))$. The following two action patterns for the rover are selected.

Action_1 : move to the neighboring grid without turning
Action_2 : turn to a different direction at the same place

In the case of a 2-1/2 dimensional environment like the surface of Mars, simple distance does not provide a correct required traverse time since the slope of the terrain can drastically affect the time. A cost function is required for estimating the time and power of motion over a 2-1/2 dimensional terrain. So the path from a start point to a goal is determined in such a way as the following cost function be minimized. The cost function E consists of two energy functions, the motion energy E_{hor} for horizontal movement and the potential energy E_{ver} for vertical movement.

$$E = E_{\mathrm{hor}} + E_{\mathrm{ver}} \quad (8)$$

where

$$E_{\mathrm{hor}} = \mathrm{K}_1\cdot\sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \quad (9),$$

(in case of Action_1)

$$E_{\mathrm{hor}} = \mathrm{K}_1\cdot\frac{\pi}{8}\sqrt{l_w{}^2+l_s{}^2} \quad (10),$$

(in case of Action_2)

$$E_{\mathrm{ver}} = \mathrm{K}_2\cdot|H(x_1,y_1,\theta_1)-H(x_2,y_2,\theta_2)| \quad (11),$$

K_1, K_2 : constant.

SIMULATION RESULTS

In order to investigate the validity of the proposed method, path planning in a terrain environment shown in Fig.3 is simulated. Table 1 shows the parameters for this simulation. Traversable areas and untraversable areas are obtained as shown in Fig.4. White regions mean traversable areas, and black regions mean non-traverse areas. Gray regions show the areas where a rover can stay stably or not depending upon the orientation of the rover. Figure 5 shows that a reasonable path can be planned by the proposed algorithm. Calculation time is about 3.0[s] (CPU: SPARC IU/FPU 40MHz).

Table 1. Parameters for a rover

map size	9.0 × 9.0 [m]
grid size	0.3 [m]
width of rover	1.1 [m]
length of rover	1.3 [m]
wheel base	1.1 [m]
distance between right wheel and left wheel	1.3 [m]
start point	(4 , 4)
goal point	(25 , 25)
K_1 , K_2	10

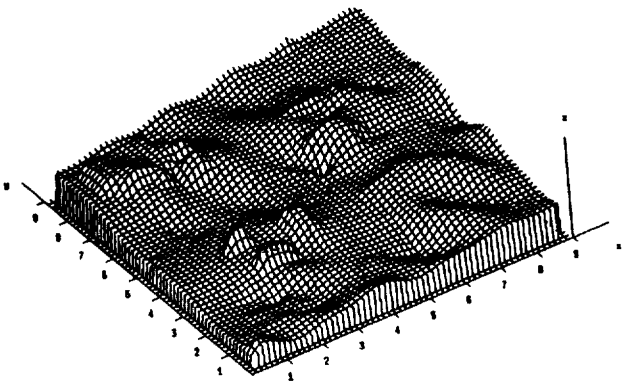


Figure 3. A terrain map

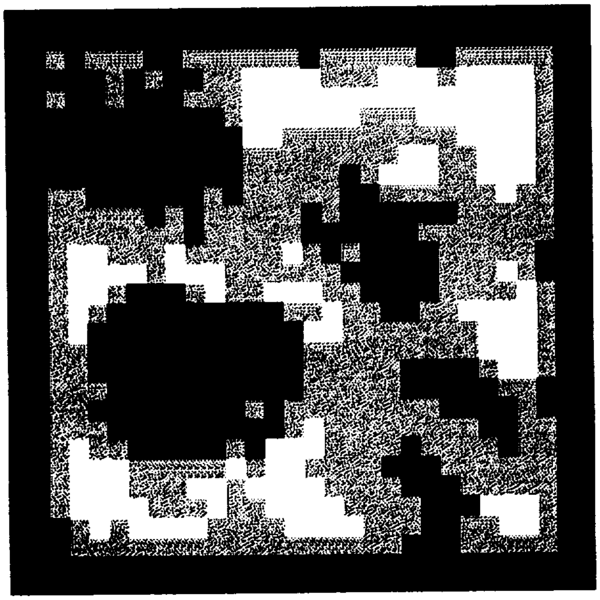


Figure 4. Traversable area

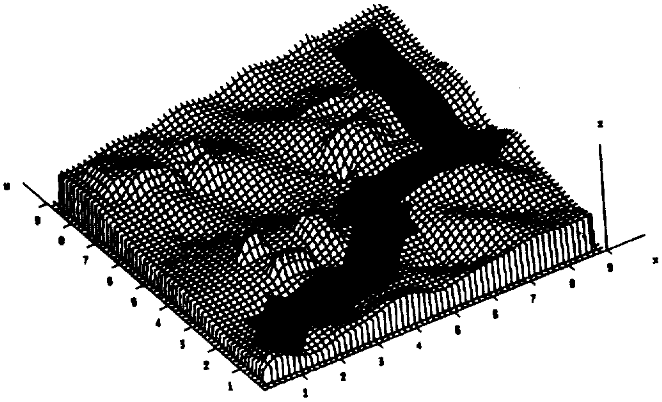


Figure 5. Simulation result

CONCLUSION

A new path planning for a planetary rover has been presented in this paper. The validity of the proposed method is confirmed by computer simulations. Experiments of mobile robot in an outdoor environment are under planning.

REFERENCES

[1] D.P.Miller, M.G.Slack, R.J.Firby, "Path planning and execution monitoring for a planetary rover," Proc. of IEEE Int. Conf. on R&A, pp.20-25, 1990.

[2] D.Lavery, R.J.Bedard, "NASA planetary rover program," Proc. of i-SAIRAS'90, pp.119-122, 1990.

[3] A.Baiget et al., "Rover: Autonomous concepts for Mars exploration," Missions, Technologies et Conception des Vehicles Mobiles Planeraires, pp.313-326, 1992.

[4] T.L.Perez, "Spatial Planning : A Configuration Space Approach", IEEE Trans. on Comp., pp.108-120, 1983.

[5] D.Gaw, A.Meystel, "Minimum-Time Navigation of An Unmanned Mobile Robot in A 2-1/2D World With Obstacles", IEEE Int. Conf. on R&A, pp.1670-1677, 1986.

[6] Z.Shiller, H.H.Lu, "Robust computation of path constrained time optimal motions," Proc. of IEEE Int. Conf. on R&A, pp.144-149, 1990.

[7] T.T.Lee, C.L.Shih, "Robust path planning in unknown environments via local distance function," Proc. of i-SAIRAS'90, pp.251-254, 1990.

[8] M.Amara, J.P.Friconneau, A.Micaelli, "Bond graph modeling, simulation and reflex control of the mars planetary automatic vehicle," Missions, Technologies et Conception des Vehicles Mobiles Planeraires, pp.509-529, 1992.

[9] R.Morlanss, A.Liegeois, "A D.T.M-based path planning method for planetary rovers," Missions, Technologies et Conception des Vehicles Mobiles Planeraires, pp.499-507, 1992.

[10] M.Hebert, T.Kanade, I.Kweon, "3-D Vision Techniques for Autonomous Vehicles", ICMU-RI-TR-88-12, CMU 1988.

Subsumption-based Architecture for Autonomous Movement Planning for Planetary Rovers

N95-23690

Shinichi Nakasuka and Seikoh Shirasaka
Department of Aeronautics and Astronautics, University of Tokyo
Hongo 7, Bunkyo-ku, Tokyo 113, JAPAN
TEL: +81-3-3812-2111 ext.6590 FAX: +81-3-3818-7493
e-mail: nakasuka@space.t.u-tokyo.ac.jp

KEY WORDS AND PHRASES

Autonomy, machine learning, planetary rover, subsumption architecture

ABSTRACT

The paper proposes a new architecture for autonomously generating and managing movement plans of planetary rovers. The system utilizes the uniform representation of the instantaneous subgoals in the form of virtual sensor states and the autonomous generation of the subsumption type plan network, which are expected to lead to the capability to persue the overall goal while efficiently managing various unpredicted anomalies in a partially unknown, ill-structured environment such as a planetary surface.

INTRODUCTION

Among the autonomous functions required for future unmanned planetary rovers, the one especially required for such rovers will be the capabiliy to generate and manage various movement plans under partially unknown, ill-structured environments. For example, the path planning will be made based on the maps of the planet which will have been obtained beforehand by the observation from the planetary orbit, but these maps will not be so accurate and there will be in many cases lots of obstacles (such as small rocks or gaps) not represented on the maps. The path planning system, therefore, must be flexible enough to compensate for the inaccuracy of the maps, quickly respond to the unpredicted events such as collisions with the unknown obstacles, gather geographical information, and re-plan the path to the goal. This kind of flexibility will be needed in many other planning activies of the planetary rovers as well.

The paper proposes a novel architecture for autonomously generating and managing such movement plans of planetary rovers. The architecture is, basically, similar to the well-known subsumption architecture (Fig.1)[1] in the sense that the finally obtained movement plans are represented in the form of a hierachical supression/promotion network of primitive reflex actions such as "moving towards a prescribed point", "wandering about", "moving towards the reverse direction when a certain touch sensor senses an obstacle", and so on. This representation of plans is, as has been discussed in many literatures, superior in 1) robustness in the actual world because no "symbolic world model" is utilized, 2) real-timeness because no complicated symbolic manipulation is required, and 3) easiness in system integration and ex-

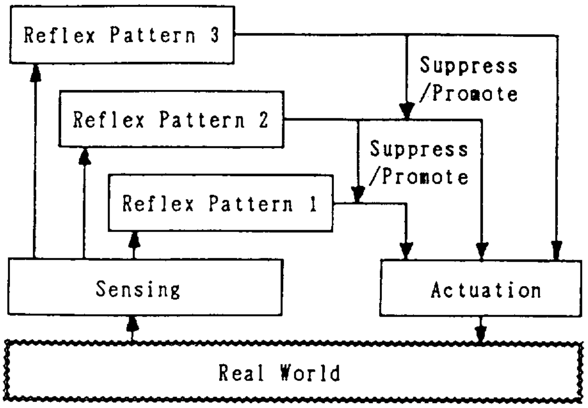


Figure 1.Subsumption Architecture

tension because a bottom-up-type system construction is quite easy. For this reason, this architecture is quite suit with the plan representation schema for rovers which move in an unstructured world. Its most significant departure from the conventional system concept is that the goal of the plan is not represented explicitly, but is achieved during the course of the interactions between the reflex actions' network (called "RAN" hereafter) and the environment. This feature is called "emergent functionality".

This architecture, however, has some difficult problems to be solved before the actual use, such as; 1) the RAN must be sophisticatedly designed by human designers so that the emergent functionality achieves the given goal, which is far more difficult task than to build a system which deals with the goal explicitly, and 2) once coded, the network is fixed during the actual operations, and the change of the environment or system itself cannot be dealt with. From these shortcomings, it can be said that the subsumption architecture cannot be employed in its original form for our objectives.

We modified and enhanced the subsumption architecture in the following three points: 1) uniform representation of the instantaneous subgoals is introduced in the form of virtual sensors so that the goal can be more explicitly persued, 2) the RAN is automatically generated by compiling the database of the actions' behavior networks obtained by machine learning, and 3) the RAN is modified during the actual operations to cope with the changes of the system and environment. The resultant system is expected to have the capability to persuit the overall goal while efficiently and more flexibly managing various unpredicted anomalies in a partially unknown, ill-structured environment such as a planetary surface.

In the following explanation, it is assumed an example task to fetch a certain object which is placed at a certain position (not at the rover position) and to carry it to a prescribed goal position. The rover is assumed to have four touch sensors (each is sensitive to two direction forth) and one camera, and be able to turn right/left and move forward/backward as illustrated in Fig.2. It is assumed the rover knows its current position and orientation.

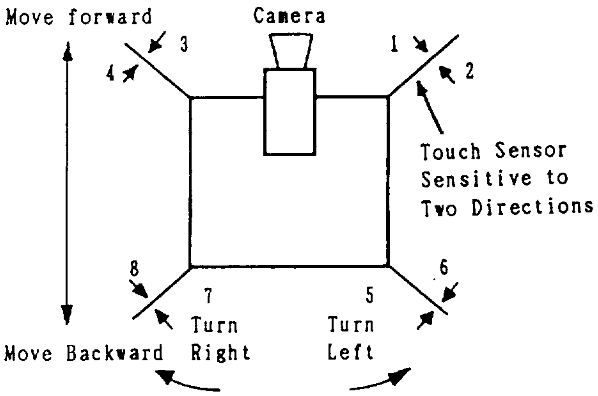


Figure 2.Schematic View of the Example Rover

NEW ARCHITECTURE

Virtual Sensor States

Various actions are uniformly represented in the form of change of sensor outputs. In order for the high-level tasks such as "Plan Path" or "Write Obstacle Position to Map" to be represented in the same way, the state such as "whether the map is updated or not" or "whether there are no obstacles between the current target and the rover position" has also been represented as one "virtual" sensor state. For the example task, the eight sensor states (including three virtual sensor states) such as

Example	< Sensor State >
35	X_1. Head Angle from the Goal Direction (0 - 360°)
10	X_2. Distance from the Goal
0	X_3. Head Angle from the Object Direction (0 - 360°)
0	X_4. Distance from the Object
3	X_5. Touch Sensor Output (0 - 8) (2 directions x 4 sensors: 0 for no touch)
1	X_6. Object Carried ? (0 for Yes and 1 for No)
1	X_7. No Obstacles between Target and Current Position ? (0 for No and 1 for Yes)
1	X_8. Map Updated ? (0 for Yes and 1 for No)

Figure 3.Content of Sensor States

in Fig.3 are employed (called $X_1 \sim X_8$.) The goal state for the example problem can be represented as $(*0*0*0**)^T$.

Learning of Behavior Network

The plan management system learns when a certain action can be applied and how the action changes the sensor state. During the learning phase, the rover chooses actions randomly, which is continued until at least one of the sensor state changes. The change of the sensor state is defined as follows; for the discrete-value type states (such as $X_5 \sim X_8$), any changes of the value, and for the continuous-value type states (the other states), transitions of the value between positive, negative and zero. Examples are described in the leftmost state transitions of Fig.4. These transitions are translated into the more abstract form of state transitions (the middle forms of Fig.4) and stored in the database. In this figure, the "* (wild card) " means an arbitrary value, ">" means a positive value and "**" means that the value has not been changed from the one before the action is taken.

After accumulating large amount of such data for each action, the conventional inductive learning algorithm is applied to yield generalized form of state transition of the action (such as the rightmost form of Fig.4.) The employed generalization rules include "turning a constant into a variable" rule, and "constraint deletion" rule. If the generalization between the current representation and the new instance would result in a trivial state transition (such as that all the states are represented as *), a disjunctive generalization is also introduced. Finally, several disjunctive representations are obtained for each action. These state transitions are called "Behavior Networks" in this paper.

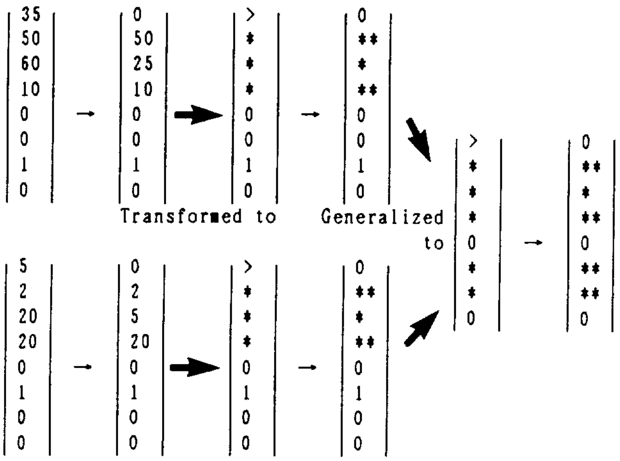


Figure 4. Acquisition and Generalization of Behavior Network

Higher level actions such as path planning also have the behavior networks. As these networks are hard to learn and can be easily defined beforehand, they are specified by the system designer. The anomalous events during the actual movements such as collisions with obstacles are also defined as state transitions.

Compilation of Behavior Networks

After behavior networks of all the actions become mature, they are compiled into a subsumption type plan network. The major tasks of this compilation are the identifications of sensor stimuli for each action to be fired and the extraction of priority relationships between the actions. The following rules are observed in constructing the plan network.

(1) Actions are defined in the form of "continue action A1 until X_k becomes a certain constant c." Therefore, for the "turn right" action, several variations of actions are generated such as "turn right until the head angle from the goal direction becomes zero" or "turn right until touch sensors sense no forth", and so on.

(2) The actions whose consequences match the goal state are considered as candidates of the lowest level of the plan network.

(3) If taking a certain action (say A1) requires that a certain state be a certain value (0 or other integers), then the action (say A2) whose consequences satisfy this precondition is categorized as a candidate of action

which must be performed before A1 (in other word, whose firing suppresses the activation of A1.) The preconditions of A1 which are not explicitly satisfied by A2 are registered as the stimuli for firing A1. Then all the consequence states of A2 are matched with the precondition states of A1, and the precondition states of A2 are replaced with the values obtained by this matching.

(4) Many hierarchical relationships will be acquired in the above processes. From these, the best plan network is obtained by searching the space of all the combinations, based on the following criteria;

- The network does not have any loops.
- The network can lead the system to the goal state from arbitrary states.

Figure 5 describes the obtained plan network for the example problem. In this figure, the wave line shows the "Supression Signal." For example, the action "MF(X_4=0)" (stands for "move forward until X_4=0") must be performed preferentially if X_4=0 is not satisfied when trying to start action "TR(X_1=0)" or "TL(X_1=0)". When trying to start "MF(X_4=0) ", if X_3=0 is not satisfied, then the action "TR(X_3=0)" or "TL(X_3=0)" is performed according to the sign of X_3. In this way, the plan network takes into account the priority relationships between actions and the anomaly handling (such as separating from a obstacle when a touch sensor finds it) as well. For example, if the rover, during a certain action (say A1), collides with an obstacle (X_7 and X_8 become 1), which first triggers the action "write obstacle position to the map (WO)" to change X_8 to 0, and then triggers "plan path (PP)" to change X_7 to 0. Then the system resumes A1, and if another action with higher priority is not triggered, action A1 is continued. Please note that as a side effect of the WO and PP actions, the states X_1 ~X_4 will be changed.

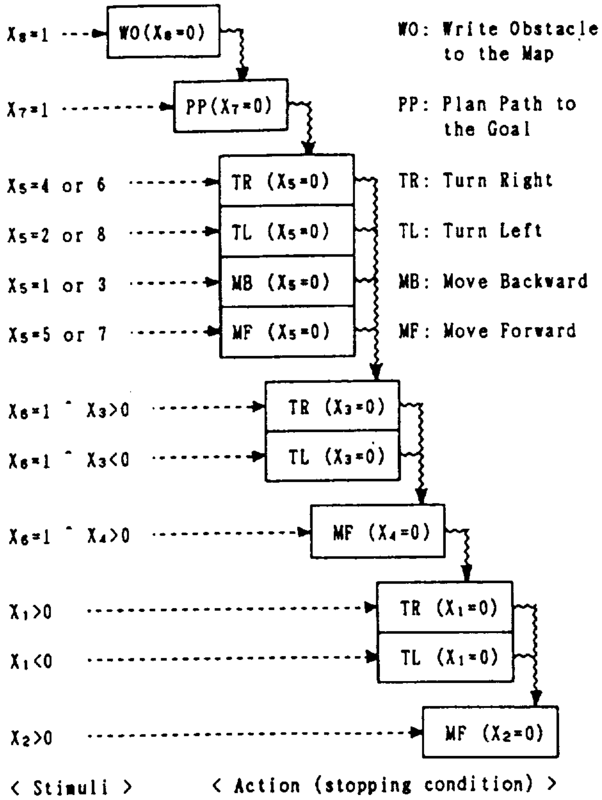


Figure 5. Obtained Plan Network for the Example Problem

If the consequence of a certain action is found inconsistent with the learned behavior network, then the learning of the correct behavior network is re-initiated for the specific action, which also triggers the recompilation of the behavior networks into the plan network. With this technique, the system has the flexibility to adapt itself to the change of the environment or the system itself.

CONCLUSIONS

An architecture to manage the rover movement plans under ill-structured, partially unknown environments has been proposed. Simulation studies have indicated the effectiveness of the architecture, and experiments using an actual rover-type vehicle is now being performed.

REFERENCES

[1] Brooks, R.A.,1986. A Robust Layered Control System for a Movile Robot, *IEEE J. Robotics and Automation*, Vol.RA-2, April: 14-23.

Very Fast Motion Planning for Highly Dexterous Articulated Robots

Daniel J. Challou, Maria Gini, and Vipin Kumar
Department of Computer Science
University of Minnesota, Minneapolis, MN 55455
Fax: (612) 625-0572
challou@cs.umn.edu (612) 626-7503
gini@cs.umn.edu (612) 625-5582
kumar@cs.umn.edu (612) 624-8023

KEY WORDS AND PHRASES

Collision avoidance, parallel algorithms, path planning, robot motion planning.

INTRODUCTION

Due to the inherent danger of space exploration, the need for greater use of teleoperated and autonomous robotic systems in space-based applications has long been apparent. However, the need for such systems has intensified lately because they will be necessary to carry out a variety of important missions. Free-flying robots carrying multiple highly dexterous robot arms have been proposed for aiding in the construction of the space station Freedom, and for assisting in satellite maintenance. Autonomous and semi-autonomous robotic devices have been proposed for carrying out routine functions associated with scientific experiments aboard the shuttle and space station. Finally, research into the use of such devices for planetary exploration continues [4].

To accomplish their assigned tasks, all such autonomous and semi-autonomous devices will require the ability to move themselves through space without hitting themselves or the objects which surround them. In space it is important to execute the necessary motions correctly when they are first attempted because repositioning is expensive in terms of both time and resources (e.g., fuel). Finally, such devices will have to function in a variety of different environments. Given these constraints, a means for fast motion planning to insure the correct movement of robotic devices would be ideal.

Unfortunately, motion planning algorithms are rarely used in practice because of their computational complexity [6]. Fast methods have been developed for detecting imminent collisions [10, 11], but the more general problem of motion planning remains computationally intractable. However, in this paper we show how the use of multicomputers and appropriate parallel algorithms can substantially reduce the time required to synthesize paths for dexterous articulated robots with a large number of joints.

We have developed a parallel formulation of the Randomized Path Planner proposed by Barraquand and Latombe [1]. We have shown that our parallel formulation is capable of formulating plans in a few seconds or less on various parallel architectures including: the nCUBE2 multicomputer with up to 1024 processors (nCUBE2 is a registered trademark of the nCUBE corporation); the CM-5 (CM-5 is a registered trademark of the Thinking Machines Corporation), and a network of workstations [3, 5]. (The results obtained on the CM-5 presented in this paper are based upon a beta version of the software and, consequently, are not necessarily representative of the performance of the full version of the software.)

One might argue that massively parallel machines are not a viable platform for space based applications due to their prohibitive cost. However, due to the continuing progress in VLSI design and economy of scale resulting from their widespread use, the cost of processors that massively parallel machines employ is expected to decrease. When this occurs, it will be feasible to build large scale

95,96,97,98

parallel computers with substantial raw computing performance at a relatively small cost.

Working projects that utilize embedded parallel processing, such as the autonomous land vehicle Navlab [8], indicate their viability. The fact that embedded parallel systems can also perform other tasks efficiently, such as image processing and image recognition, justifies their use in planning applications as well.

RANDOMIZED PARALLEL MOTION PLANNING

Most motion planning algorithms decompose the search space into discrete components called cells [9]. The motion planning problem then becomes one of computing a decomposition and searching through sequences of contiguous cells to find a path through free space (i.e. a sequence of configurations that involves no collisions with obstacles).

Unfortunately, as more degrees of freedom are added to the robot most methods become computationally impractical [9]. The only existing motion planning methods capable of synthesising plans in reasonable time frames (i.e., times on the order of minutes [6]), for robots with more than three degrees of freedom utilize an approximate decomposition of the configuration space (C-Space). The C-space is the space defined by parameters that uniquely specify the position of the robot. To obtain such performance, most methods precompute a significant portion of the C-space. Total precomputation is impossible because of both the time required to perform the computation and the amount of memory required to store the resulting C-Space. Unfortunately, precomputation relegates such methods to static workspaces, and hence they are not well suited to the space-based applications described earlier.

Our method is a parallel formulation of the Randomized Path Planner proposed by Barraquand and Latombe [1]. Space is represented with bitmap arrays. The configuration space is discretized and searched using best first search with random walks. Artificial potential fields are used as the heuristic to guide the search. The potential fields are precomputed, but their computation requires at most a few tens of seconds (and it is readily parallelizable). Furthermore, the method works with discrete representations of the environment, so it can readily be coupled with fast methods of producing such representations, such as the method proposed by [7].

The path is constructed incrementally as follows. A new configuration is randomly generated from the current configuration at the start of each step. If the heuristic value of the new configuration is smaller than the current value, and the move does not cause a collision, then the new configuration is added to the path and the search process is resumed. Otherwise another neighbor is investigated. When none of the neighbors has a smaller value than the current configuration, a random walk is executed and then the search process resumes. This process is repeated until a solution is found.

We first broadcast a bitmap representation of the workspace and the desired goal location to all processors, and then check for a message indicating that a processor has found a solution. Each processor runs the same basic program. The only interprocessor communication is the initial broadcast and the termination check. The search and random walks are the means by which the search-space is partitioned, as they insure that each processor searches different parts of the C-Space.

Although the method is only probabilistically complete, a large number of experimental results indicate that with a sufficient number of processors a solution is always found in very short time frames [3, 2].

DISCUSSION OF RESULTS

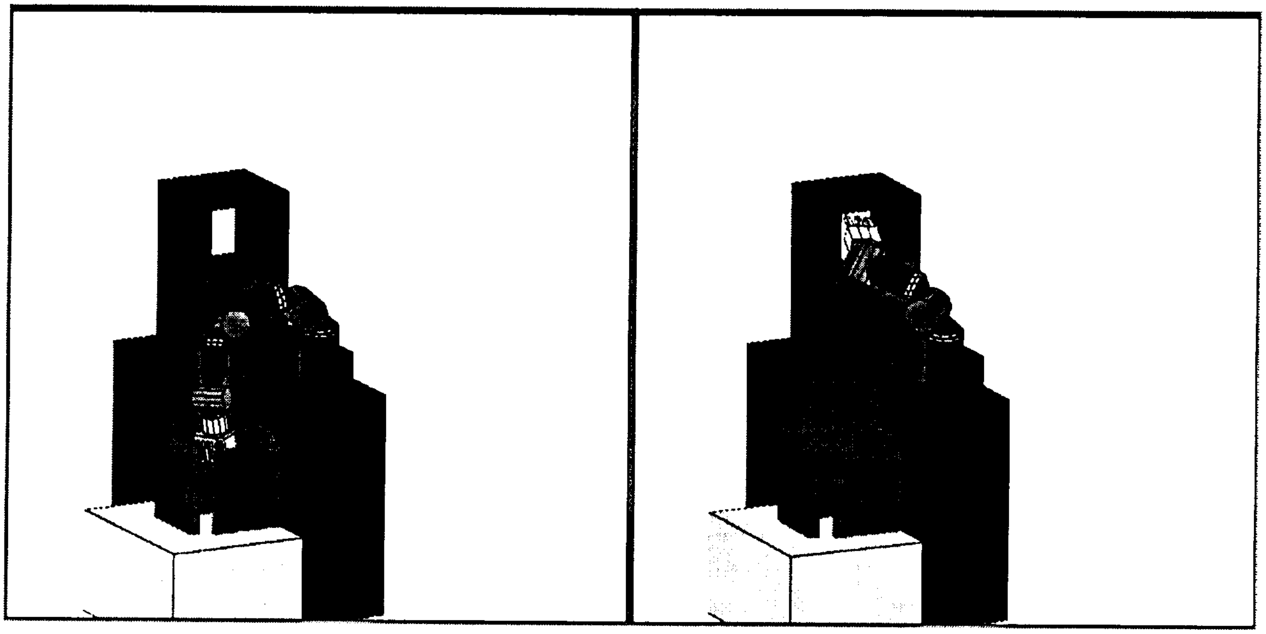
Figure 1 shows the start and goal configurations for one of our test cases for motions of a seven degree of freedom Robotics Research arm operating in a 128^3 cell workspace. Each cell in the robot's workspace represents a volume of 2.1 cubic centimeters. Each joint has up to 128 discrete positions (2.8125 degrees per position). The table shows the results on up to 256 processors on the CM-5 multicomputer. Each processor requires approximately 13.1 megabytes of random access memory.

The table indicates the benefits of parallelizing the planner. For the problem instance shown just

32 processors are required to cut the average solution an order of magnitude to under ten seconds, and 64 processors cut the average solution time to under five seconds.

In addition to delivering paths in shorter time frames, another important property of the parallel formulation is that, when it is executed with a larger number of processors, it tends to produce better solutions. We have observed this behavior in all the experiments we have performed to date. In the example, 32 processors yield a solution path length about one fourth as long as the average solution path length delivered by one processor, and 128 processors reduce the average solution path length by an order of magnitude. The variance in time to solution behaves similarly, that is, it falls off as the number of processors attempting to solve the problem increases.

The performance falls off and the average time taken to solve the problem moves toward a constant value as we increase the number of processors. This is because we hit a point where the number of processors required to insure that one processor will find a solution in the minimum possible time is optimal or near optimal for the problem instance. The probability that the random component of the algorithm will ensure that different processors are exploring different parts of the search space decreases as we add more processors. When we reach that point, then adding more processors will just result in more processors doing redundant work (in the average case).



No Processors	1	32	64	128	256
Avg Search Time	102.34	8.39	5.36	3.37	2.32
Std Dev	108.33	5.24	3.26	2.17	1.26
Avg Path Length	4264	1178	1351	967	531
Std Dev	5196	1550	1942	1277	476
Avg Speedup	1.00	12.02	19.09	30.37	44.11

Figure 1: The figure shows the start and goal configurations for a seven degree of freedom Robotics Research arm. The robot is reaching from the box in front of it, up and into the box on the left. The table shows data for at least 64 runs on a CM-5 multicomputer. All times are in seconds.

We have developed fast performance prediction methods that can be used to determine whether the number of processors available is adequate or excessive [2]. Because of the way the random planner escapes local minima and generates successors, as the minimum solution length and the degrees of freedom of the robot increase the number of different (not necessarily optimal) solution paths increases dramatically. The number of solution paths with similar lengths increases dramatically as well. This increased solution density enables the planner to perform well in instances where deterministic methods would encounter difficulty.

If a priori knowledge about obstacles allows a coarser discretization of C-space, (such as the 64 discrete positions used by [11]), then our experimental results [2] indicate that we can cut the planning time by at least a factor of three. Thus, coarser discretizations coupled with faster processors, such as Digital Equipment's alpha chip, would enable our system to deliver sub-second performance using a reasonable number of processors.

We are currently in the process of parallelizing the computation of the 3D artificial potential field maps. Preliminary results indicate that it is possible to complete the heuristic computation process in real-time. As a result, given a discrete 3D picture of an environment, our planner will be able to formulate motion plans in very fast time frames.

ACKNOWLEDGMENTS

This work was supported in part by Contract Number DAAL03-89-C-0038 between the Army Research Office and the University of Minnesota for the Army High Performance Computing Research Center. Additional support was furnished by NSF/CDA-9022509, DA/DAAH04-93-6-0800, NSG/IRI-9216941, and the Center for Advanced Manufacturing, Design and Control of the University of Minnesota.

We would like to sincerely acknowledge Mike Hennessey and Max Donath for helping us model the Robotics Research arm; Jean Claude Latombe at Stanford University for providing access to implementations of the Random Path Planner; David Strip and Robert Benner at Sandia National Laboratories for providing access to the nCUBE2.

REFERENCES

[1] J. Barraquand and J. C. Latombe. Robot motion planning: A distributed representation approach. *Int'l Journal of Robotics Research*, 10(6):628–649, 1991.

[2] D. Challou. Parallel search algorithms for robot motion planning. Ph.D. dissertation., The University of Minnesota, 1994.

[3] D. Challou, M. Gini, and V. Kumar. Parallel search algorithms for robot motion planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, volume 2, pages 46–51, 1993.

[4] S. Dubowski and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free floating space robotic systems. *IEEE Trans. Robotics and Automation*, 9(2):541–543, 1994.

[5] M. Gini, D. Challou, V. Kumar, and B. Camel. Parallel algorithms for path planning. In *Proc. Virtual Reality Systems '93, Teleoperation '93, and Beyond Speech Recognition '93*. Sig-Advanced Applications, Inc., New York, 1993.

[6] Y. Hwang and N. Ahuja. Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):219–291, 1992.

[7] J. P. Jones. Real-time construction of three-dimensional occupancy maps. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1993.

[8] T. Kanade, M. L. Reed, and L. E. Weiss. New technologies and applications in robotics. *Communications of the ACM*, 37(3):58–67, March 1994.

[9] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publ., Norwell, MA, 1991.

[10] C. A. Shaffer. A real-time robot arm collision avoidance system. *IEEE Trans. Robotics and Automation*, RA-8(2):149–160, 1992.

[11] T. S. Wikman, M. Branicky, and W. S. Newman. Reflexive collision avoidance: a generalized approach. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, volume 3, pages 31–36, 1993.

Control technique for Planetary rover

N95- 23692

Ichiro NAKATANI,Takashi KUBOTA
The Institute of Space and Astronautical Science
3-3-1 Yoshinodai Sagamihara-shi 229,Japan
Tel 0427-51-3911 Fax 0427-51-3988
Tadashi ADACHI,Hiroaki SAITOU,Sinya OKAMOTO
Aerospace Division,Nissan Motor Co.,Ltd.
21-1,Matoba-Shinmachi,Kawagoe-shi,Saitama-ken 350-11,Japan
Tel 0492-31-1113 Fax 0492-31-1116

Abstract

At the beginning of next century, several schemes sending a planetary rover to the moon or Mars are being planned. As part of development program, autonomous navigation technology is being studied for realizing the rover to be able to move autonomously in a long range on unknown planetary surface. In the previous study, we tried the autonomous navigation experiment on the outdoor test terrain by using rover test-bed which was controlled by a conventional sense-plan-act method. In the experiment, the problem that the rover moved into untraversable areas occurred in some cases. For improvement of this situation, new control technique have been developed that it has reaction behavior to react by the outputs of the proximity sensors. We have been trying to develop the rover test-bed system and autonomous navigation experiment were executed by newly developed control technique using the new rover test-bed. In this experiment, our new control technique was able to produce the control command effectively to avoid the obstacles and to guide it to the goal point safely in the outdoor test site.

1. Introduction

There are two main methods to navigate the rover to its destination. One is remote control by operators on the earth. The other is an autonomous navigation by a control system on-board the rover. In a practical navigation, these two methods will be used to complement each other. And so, both methods must be studied, then we have been studying the autonomous navigation technology for the rover. This paper introduces the rover navigation method applied hybrid behavior control technique and also, the results of the autonomous navigation experiment which has been executed in the outdoor terrain model are shown.

2. Basic concept of rover navigation

The basic concept of our rover navigation system is described in Figure 1. In this concept, a remote sensing satellite is sent to the orbit of the moon or Mars to collect the surface data before the rover exploration and a set of coarse map (global map) of the terrain might be compiled from the remote sensing data. Then, operators make a plan of global path on the global map to lead the rover to its destination. After that, the rover is guided along the global path and the rover observes the terrain in front of it with the Image laser range finder (ILRF) or the other 3D terrain sensors. If the rover finds out some areas where it can not go through because of limited

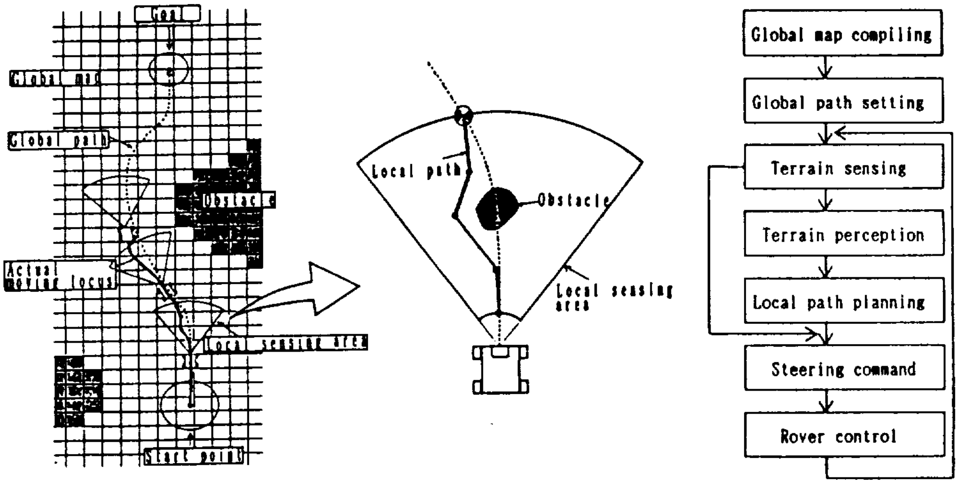


Figure 1 Basic concept of rover navigation

performance, then it executes local path planning to set up a new local route to avoid the untraversable area within the sensing area.

3. Control architecture

To realize the effective autonomous navigation algorithm for the rover, we tried to connect several functions effectively.

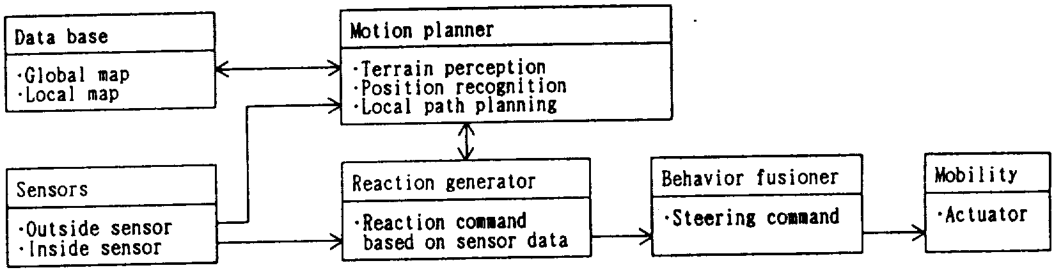


Figure 2 Block diagram of control architecture

The block diagram of newly developed control architecture for the autonomous navigation system is shown in Figure 2. Our control architecture consist of two layers and one behavior fusioner. The upper layer is called motion planner which has the role of deliberative task execution such as perception of the terrain condition in front of the rover, proper local goal searching in the sensing area for local path planning and executing local path planning and so on. The lower layer executes reaction control task, if the on-board sensors detect some dangerous situation, this layer immediately produces the reaction command to escape this situation. The reaction command will rescue the rover from collision with obstacles, tipping over, stack in loose ground and so on. The computational load of this layer must be kept as low as possible because the reaction command must be produced in a very short time. The behavior fusioner has the function as follows,

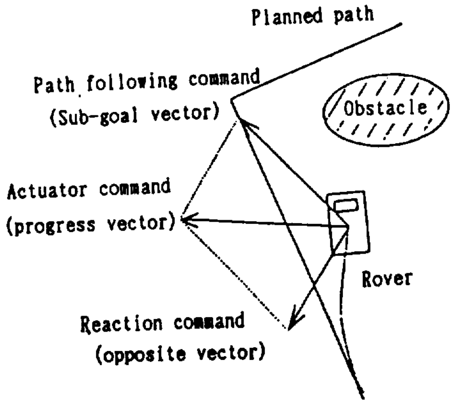
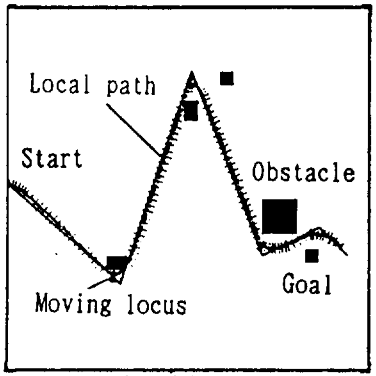
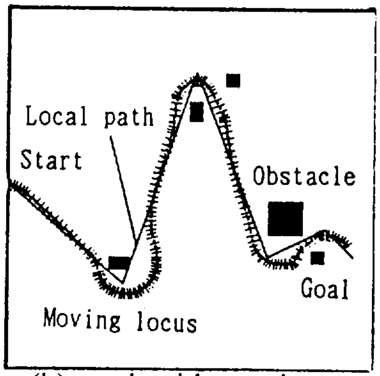


Figure 3 Command combination method

the behavior fusioner executes to combine the behavior command from upper layer and the reaction command from lower layer. In newly developed control method, a sort of potential method as shown in Figure 3 is used to combine the commands. In this figure 3, the behavior fusioner produces an actuator command to steer the rover with respect to progress vector which is a sum of sub-goal vector and opposite vector produced in the upper and lower layer. Before installation of this method into the rover test-bed, evaluating efforts for this method was done through computer simulation.



(a) Result without reaction



(b) result with reaction

Figure 4 One of the computer simulation results

The result of this simulation is shown in Figure 4. In case of no reaction control, when the planning path was very close to the obstacles, the rover collided with the obstacles as shown in Figure 4(a). While, by using reaction control, the rover could avoid the obstacles and arrived safely at a goal as shown in Figure 4(b).

4. Rover test-bed for autonomous navigation Experiments

We developed new rover test-bed for autonomous navigation experiments in the natural terrain. The characteristics of rover test-bed is described in Table 1 and the configuration of rover test-bed is shown in Figure 5.

Table 1 Characteristics of the test-bed

Mobility weight		75kg
Payload weight		45kg (battery included)
Size		length 1500mm
		height 1300mm
		width 1200mm
Driving mechanism		Servo motor Speed reduction gear
Velocity		15cm/sec
Climbable slope		30°
Maximum climbable height of obstacle		30cm
Sensor	Terrain sensor	Image laser range finder
	Posture sensor	Inclinometer (pitch, roll)
	Position sensor	Inertial sensor
	Proximity sensor	Laser proximity sensor ×5
On-board computer (signal treatment)		PC/AT (HOST) ×2 DSP ×4
Ground computer (environment perception action, planning)		Sun SS-10
Communication		Ethernet (optical fiber)

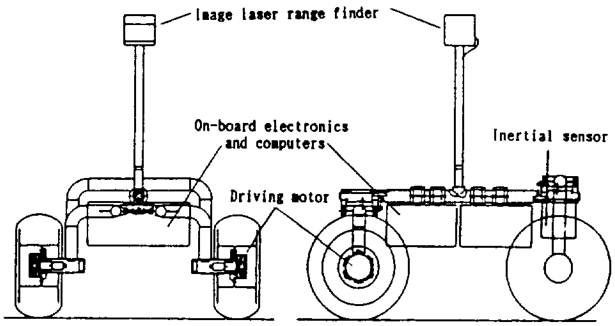


Figure 5 Configuration of the rover

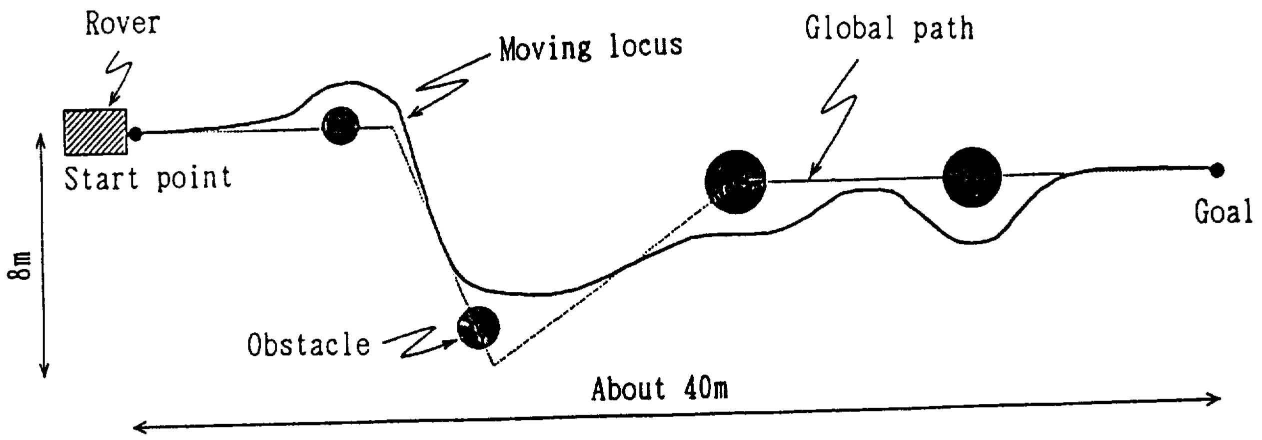


Figure 6 One of the autonomous navigation experiments results

5. Autonomous navigation experiments

The autonomous navigation experiments to evaluate the control method were executed using the rover test-bed in the outdoor terrain model. One of the experimental results is illustrated in Figure 6. In the experiments, the new control architecture was found to effectively work to avoid the untraversable area by generating reaction commands. As a result, the rover could arrive at the goal safely and it took about 20 minutes to move for about 40m.

6. Conclusion

In this study, the control technique for autonomous navigation to guide the rover to its destination area in outdoor environment has been developed. As we executed the actual autonomous navigation experiment, we could understand the characteristics and problems of our control technique and confirm the effectiveness of the hybrid method of two behavior commands newly adopted to improve the control performance. In next step, we will try to study to realize the higher level of autonomous navigation system with the performance which adapts to various situations that the rover would meet in the planetary environment.

Reference

[1] T.IWATA et al.,"Lunar Mobile Explorer, 450kg Concept and Demonstrations." IAF Paper No.93-Q.1.375 1993.

[2] T.Eto, M.Oda et al.,"The study of the mobile lunar explore." Mission, Technologies and Design of Planetary Mobile Vehicles, January 1993.

[3] D.Payton et al.,"Internalized Plan: A Representation for Action Resources" Robotics and Autonomous System 6 PP89-103, 1990.

[4] E.Gat et al.,"Navigation Templates: Enhancements, Extensions, and Excrements." Proc. 1993 IEEE Cont. on Robotics and Automatics, Atlanta USA, May 1993, PP541-547

[5] B.Bramitl et al.,"Dynamic trajectory planning for a cross-country navigator" SPIE Vol.1831 Mobile Robots 1992, PP564-578

[6] G.Giralt et al.,"Remote Intervention, Robot Autonomy, And Teleprograming: Generic Concepts And Real-World Application Cases" Proc. 1993 IEEE/RSJ on Intelligent Robots and Systems. Yokohama, Japan, July 1993 PP314-320

AUTONOMOUS NAVIGATION SYSTEM

FOR THE MARSOKHOD ROVER PROJECT

N95- 23693

C. Proy **M. Lamboley** **L. Rastel**

Centre National d'Etudes Spatiales (CNES)
18 Avenue E.Belin
31055 Toulouse Cedex - FRANCE

Tel: (33) 61 27 41 55 Fax : (33) 61 28 19 96
E-mail: proy@philae.cnes.fr

KEY WORDS AND PHRASES

Mars exploration - Mobile robotics - Marsokhod rover - Autonomous navigation - Stereovision

ABSTRACT

This paper presents a general overview over Marsokhod rover mission . The autonomous navigation for a Mars exploration rover is controlled by a vision system which has been developed on the basis of two CCD cameras, stereovision and path planning algorithms. Its performances have been tested on a Mars-like experimentation site.

INTRODUCTION

This study has been performed in the frame of the Russian-French project Marsokhod : Marsokhod is a small rover (less than 100 kg) designed for Mars exploration ; its launching is foreseen in 1998. This six-wheeled vehicle was tested 1992 in Death Valley and has shown a remarkable locomotion capacity ; it was foreseen to run it on the surface of Mars, being teleoperated from the Earth. However, due to important delays for data transmission, its operational range in such a mode is very limited. In order to improve this situation, an autonomous navigation facility is under study in CNES (French space agency), and in close cooperation with the Russian Marsokhod team. This should increase drastically the operational range of the rover and its scientific return, allowing movements of several tens of meters per days, only limited by available on board energy. Furthermore, this system enables a short range perception, thus a safer obstacles detection with an increase of the rover's security.

The goal of the work in progress is to prove the feasibility and to develop an autonomous path generation sub-system. This includes mainly a pair of stereo cameras and the necessary software to implement on-board 3D reconstruction and path-planning. The basic idea is to acquire and process a pair of stereo images after a stop of the rover every 5 to 10 meters.

A series of tests has been achieved since 1993 to assess precision, robustness and performances of algorithms as well as to define the specifications of the vision hardware, in particular the focal length and the stereo basis. This paper presents results of the studies and experiments performed.

The constraints of the project and the poor knowledge on the Martian environment make it an interesting challenge to improve the rover's autonomy. It is also a preparation for future projects with stronger requirements, presently being carried out by CNES [1].

CONSTRAINTS ON THE VISION HARDWARE

The autonomous navigation system is confronted with important constraints in the area of available resources, some of them are highlighted hereafter :

- use of a fixed stereovision device (2 CCD cameras) set at 1 meter high; CCD devices will be TH7861 matrix (300x400 pixels of 23x23 microns),
- no specific calibration facility,
- use of one Transputer based board to process pictures, with restricted memory size,
- hostile environment in which the device has to operate (low temperature with large variations and dust are major threats for mechanical stability and electronics).

The impossibility to move the cameras has some impacts on the stereovision device:

- a wide field of view is required to reduce the invisible area just in front of the rover, to get a fair optical range (5 to 10 meters) and to have a good probability of finding a path in the common field of view of both cameras. This leads to chose small focal lengths and to deal with optical distortions.

- no target can be seen by the cameras, so that no in-flight calibration can be performed. We need to fix rigidly the cameras and to know exactly their position and orientation to apply on-board stereovision.

AUTONOMOUS NAVIGATION

The general scheme of the process to be applied on Marsokhod images involves two main steps for each cycle :

- Stereovision and 3D reconstruction of the relief of the terrain surface,
- Detection of obstacles from the disparity map and path planning minimising risks for the rover.

The main algorithms have already been described in [2], so that we will focus here on 3D reconstruction accuracy tests, which enabled us to select the two basic parameters of the device : the length of the stereo basis and the focal length.

The 3D reconstruction of the points of the terrain surface is done without calibration, by using the camera's geometry and intrinsic characteristics.

3D reconstruction accuracy

As it is not necessary for the reconstruction error to be much less than the error on the knowledge of the rover's motion, the specification for 3D reconstruction accuracy has been defined as follows : up to 4 meter, the error shall be less than 4% of the distance; between 4 and 10 meters, the error shall be less than n%, n being the distance in meters.

An error model has been defined, with the following elements :

- uncertainty on the size of pixels and focal length,
- distortion residue,
- quantization on x and y,
- uncertainty on disparities.

Although this model is a worst case model (sum of absolute values of each error), the results are close enough of the error specification (see Figure 1).

To assess real errors, measurements have been done on a set of structured objects with specific marks. The distances between the marks have been measured (measurement accuracy : 1 mm) and compared with the values given by 3D reconstruction algorithms. This has been done on a set of about 20 distances in various conditions :

- distances from the cameras: 1.5, 3.4 and 6.7m
- stereo bases : 150 , 300 and 450 mm
- different resolutions : full resolution (i.e. 2 milliradians per pixel), resolution reduced to 1/2 (pixels merged by 4) and resolution reduced to 1/3 (pixels merged by 9).

Maximum and quadratic errors have been calculated for all these configurations. We found that the specified 3D reconstruction accuracy can be reached easily with a relatively short stereo basis (less than 300 mm) and low resolution (4 milliradians for pixel resolution); with a TH7861 matrix, this leads to a focal length of 5.7 mm. Results corresponding to this choice are shown in Figure 1.

Obstacle mapping and path planning

Several tens of image pairs have been acquired on a specific Mars like test area made up at CNES. The robustness of the process has been evaluated for various lightening conditions and terrain configurations . One example is presented here from two 512*512 CCD cameras, with 4.8 mm focal length optics and a 30 cm stereo basis. The top view of the obstacles mapping presents holes corresponding to areas occulted by rocks, but it has been possible to find a path large enough for the rover (Figures 2 and 3).

Performances

The performances given in Table 1 are not fully representative for the final software because the present release has not yet been fully optimised. A rough performance estimate for the in-flight hardware (384x288 CCD and T800 transputer) has also been performed .

CONCLUSION

The goal was to prove the feasibility of an autonomous and safe motion of Marsokhod on the surface of Mars, allowing the rover to get rid of teleoperation constraints, and thus to reach a range fitting with its locomotion capacity.

The results of the studies and experimentation done in CNES in the frame of Marsokhod project have shown that it is possible to reconstruct 3D points and to find trajectories on a non structured Mars like landscape with a sufficient accuracy for the requirements of a sub-system of autonomous navigation . According to these results, a prototype of stereovision device is presently under development and will be used in the coming months to test the complete sub-system , in terms of hardware as well as the implemented algorithms.

The new acquisition campaign scheduled for 1994 foresees real time processing of the images taken at each position, and moving the cameras in order to fulfil the planned path ; this will enable us to check the elementary paths given by the algorithms and also to see if it is possible to reach a given target by connecting consecutive trajectories. After this, we plan to make tests on board of the Russian rover at the end of 1994 to analyse the interface with the control of the rover and to define the exact use of vision in the Marsokhod mission.

REFERENCES

[1] L. Boissier, B. Hotz, C. Proy, O. Faugeras, P. Fua , 1992. Autonomous planetary rover (VAP): onboard perception system concept and stereovision by correlation. In *Proceedings of IEEE International Conference on Robotics and Automation.*

[2] C. Proy, M. Lamboley, I.Sitenko, T.N.Guyen , 1993. Improving Autonomy of Marsokhod 96 . In *Proceedings of 44th Congress of the International Astronautical Federation.*

Performances	256 x 256 images	512 x 512 images	384 x 288 images
Time	3.6 sec on Sparc10	40 sec on Sparc 10	90 sec estimated on T800
Memory (code + data)	65O Kbytes	2 Mbytes	1 Mbyte

Table 1 : Performances of the complete processing sequence

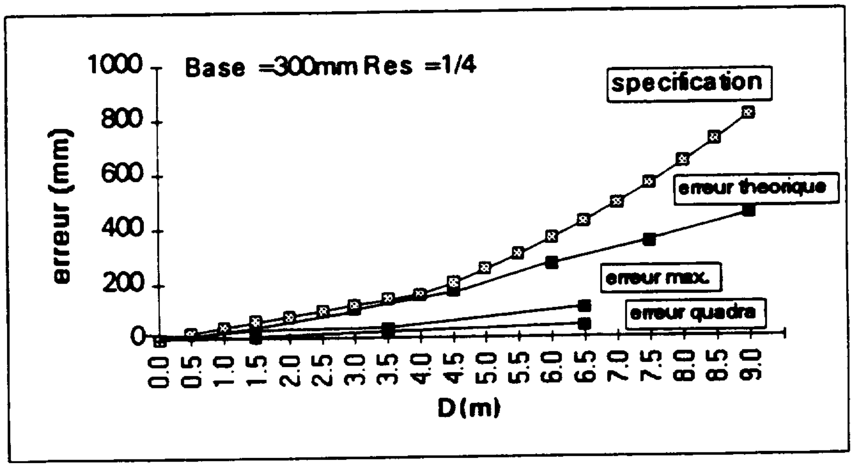


Figure 1 : Example of 3D reconstruction accuracy

Figure 2 : Image from Geroms Site
The path found for rover
has been traced in black

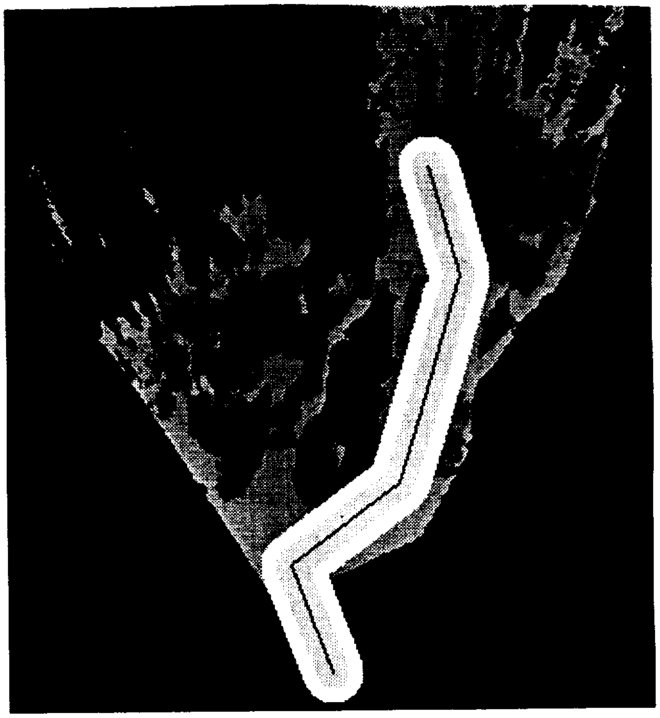


Figure 3 : Top view of the obstacle map
black: unknown areas, dark grey:
obstacles, white: path of the rover

A Unified Control Architecture for Planetary Rovers

A. M. Alvarez, W. De Peuter, and P. Putz

ESA
Noordwijk, The Netherlands

THIS PAPER WAS NOT SUBMITTED IN TIME FOR PUBLICATION.

Lunar Rover Technology Demonstrations with Dante and Ratler

N95- 23694

Eric Krotkov, John Bares, Lalitesh Katragadda, Reid Simmons, Red Whittaker
Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, Pennsylvania
Krotkov tel (412) 268-3058 fax (412) 621-1970 email: epk@cs.cmu.edu

KEY WORDS

Lunar rover, robotic exploration, autonomous systems

INTRODUCTION

Carnegie Mellon University has undertaken a research, development, and demonstration program to enable a robotic lunar mission. The two-year mission scenario is to traverse 1,000 kilometers, revisiting the historic sites of Apollo 11, Surveyor 5, Ranger 8, Apollo 17 and Lunokhod 2, and to return continuous live video amounting to more than 10 terabytes of data. Our vision blends autonomously safeguarded user driving with autonomous operation augmented with rich visual feedback, in order to enable facile interaction and exploration. The resulting experience is intended to attract mass participation and evoke strong public interest in lunar exploration.

The encompassing program that forwards this work is the Lunar Rover Initiative (LRI). Two concrete technology demonstration projects currently advancing the Lunar Rover Initiative are:

- The Dante/Mt. Spurr project, which at the time of writing is sending the walking robot Dante to explore the Mt. Spurr volcano, in rough terrain that is a realistic planetary analogue. This project will generate insights into robot system robustness in harsh environments, and into remote operation by novices.
- The Lunar Rover Demonstration project, which is developing and evaluating key technologies for navigation, teleoperation, and user interfaces in terrestrial demonstrations. The project timetable calls for a number of terrestrial traverses incorporating teleoperation and autonomy including natural terrain this year, 10 km in 1995, and 100 km in 1996.

This paper will discuss the goals of the Lunar Rover Initiative and then focus on the present state of the Dante/Mt. Spurr and Lunar Rover Demonstration projects.

LUNAR ROVER INITIATIVE

The programmatic goals of this initiative include conducting terrestrial demonstrations, and forming a consortium of partners and technical providers. The principal purpose of the demonstrations is to evaluate the readiness for lunar missions of key rover technologies such as teleoperation interfaces and on-board perception and planning.

Key participants to date include Carnegie Mellon, NASA, LunaCorp, and Sandia National Laboratories. LunaCorp is a commercial entity whose purpose is to foster commercial lunar exploration. The partners are negotiating with technical service providers and with potential customers.

An important participant in the initiative is the NASA Robotics Engineering Consortium, formed in 1994 to commercialize advanced robot technology. The consortium is providing large-scale indoor test tracks and an umbrella for the process of rover development and integration by industrial participants. These facilities will support extensive testing of lunar mission scenarios with different emphases on entertainment and science.

The Lunar Rover Initiative will substantially advance such planetary exploration technologies as high-bandwidth mobile communications, teleoperation, autonomous perception and planning, robotic safeguarding, and durability in harsh environments. By driving and demonstrating these technologies, the initiative provides a path to a lunar launch within the millennium.

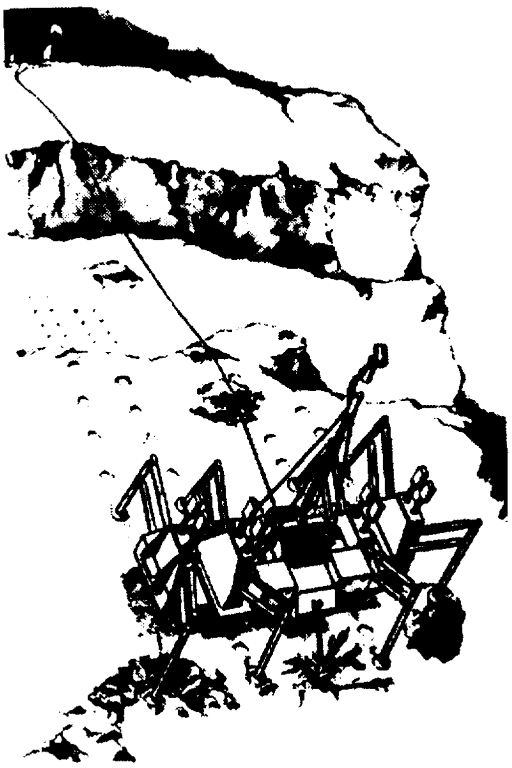


Figure 1 Dante

DANTE/MT. SPURR

Dante II (Figure 1) is an eight-legged, walking and rappelling robot for active volcano exploration. Following Dante I's attempt to explore the active Mount Erebus volcano in Antarctica in 1993, the robot has been reconfigured and further developed for a 1994 mission to Mount Spurr, Alaska. One of the primary objectives of the 1994 Mt. Spurr program is to demonstrate robotic exploration of harsh, barren, and steep terrain such as those found on the Moon and planets.

Presently, the robot is being used to explore and collect gas samples from the crater floors of active volcanos. High-temperature fumarole gas samples are prized by volcanologists. However, collecting the samples is very dangerous and poses many challenges for scientists. For example, in two separate events in 1993, eight volcanologists were killed while collecting samples and monitoring volcanoes. Without jeopardizing human safety, creation of robots such as Dante allow scientists to collect gas samples and examine crater floors from safe and remote locations.

Dante combines tether and leg motion to rappel up and down steep slopes and sheer cliffs. Dante's eight pantographic legs are organized in two groups of four, which alternately support and advance the robot. Similar to a mountain climber rappelling on a mountain cliff, the tether cable provides a reactive force to gravity and assists in maintaining equilibrium as the robot rappels up and down steep slopes or cliffs. Dante can also walk over obstacles as large as one meter high.

Dante receives power and telemetry through the tether cable, making it an ideal deployment platform for remote, multi-day explorations. Mounted on top of Dante is a laser rangefinder that perceives and maps the terrain around the robot within a six meter radius. An on-board computer then uses the terrain information to determine safe paths and adjusts its gait to overcome or avoid obstacles.

For the Mt. Spurr mission, Dante will operate in a self-reliant wireless mode, interacting with operators 130 kilometers from the volcano. During the expedition, Dante will demonstrate that it is capable of traversing escarpments and exploring craters in challenging environments. Dante will also demonstrate competent ascent and descent of steep and rough terrain as well as withstand environmental challenges from cold, high winds, high humidity, and exposure to acid gas. Other principal objectives for this mission are to demonstrate:

- key ingredients of teleoperation and control;
- autonomous control for certain segments;
- remote operation of a robotic walking system with interfaces appropriate for novices;
- ability to deploy scientific equipment and gather real-time data.

Dante has successfully completed a mission rehearsal totalling 400 m on a 35 degree slope, a critical part of its mission readiness review. At the time of writing, the robot is in Alaska, ready to begin its mission in the unforgiving environment of an active volcano.

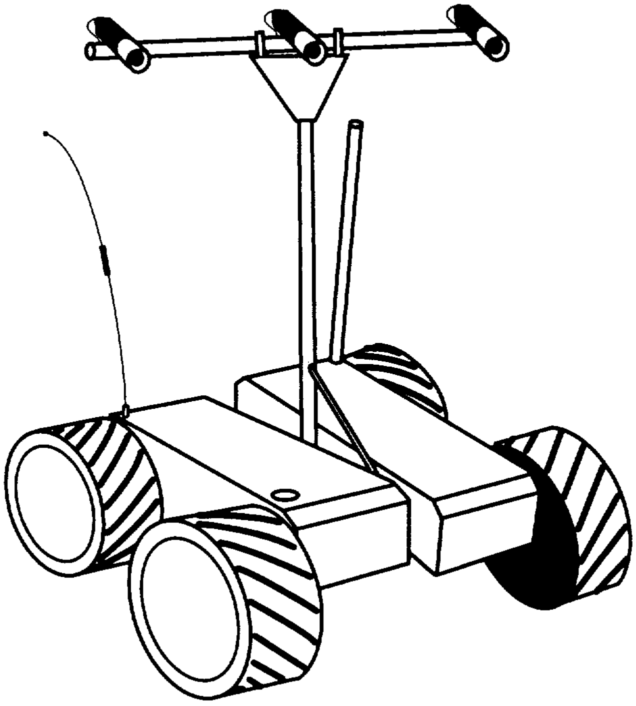


Figure 2 Ratler

LUNAR ROVER DEMONSTRATION

The Lunar Rover Demonstration (LRD) robot system is designed to competently and reliably traverse lunar-like terrain. This section describes the central system components: the rover mechanism and real-time controller, a perception system using trinocular stereo, local and global planning algorithms, and a task-level controller.

Mechanism and Controller

The Ratler (Figure 2) is a four-wheeled platform developed by Sandia National Laboratory. (The name is an acronym for Robotic All-Terrain Lunar Exploration Rover.) The skid-steered vehicle features an articulated chassis in which the body is divided into two halves, with two wheels on each side. The halves are joined together such that they may rotate along the lateral axis to enhance the mobility and stability of the platform.

Control of the Ratler may be directed from a local pendant, a remote command station, or on-board processors. An RF serial link and a microwave video link provide telemetry. State sensors include encoders on drive motors, a compass, three inclinometers, and a turn-rate gyro.

To estimate the position and attitude of the vehicle as it travels, we have formulated and implemented a dead reckoning algorithm that maintains an estimate of the robot's position and orientation in a fixed, external reference frame. To improve both reliability and accuracy, in addition to the conventional inputs from motor encoders, attitude inputs from the state sensors have been incorporated.

Perception

The perception system consists of a stereo mapping module that derives terrain information from stereo images. The hardware consists of three CCD cameras mounted on a mast 1.5 meters tall. To maximize image stability as the rover traverses surface irregularities, a motion-smoothing linkage averages the pitch of the two Ratler bodies.

The mapping software consists of a stereo matching module that computes disparities from trinocular images using normalized correlation, and a mapping module converting image-space disparities into camera referenced Cartesian coordinates.

Planning

Ranger is a local path planner that takes three-dimensional sensor data as input and produces viable driving commands as output. It is concerned neither with controlling actuators (that is the job of the vehicle controller) nor with generating strategic goals (that is the job of the global path planner).

The Ranger system is an intelligent, predictive, state-space controller: intelligent because it uses three-dimensional scene data; predictive because it reasons from its knowledge of its capability to react to hazards; state-space because it explicitly forms an expression of the vehicle dynamic state vector as the primary signal upon which decisions are based.

The D* algorithm is a global path planner that provides a means to evaluate terrain paths coupled with vehicle constraints to arrive at an optimum path given available information. D* is also efficient and can provide real-time replanning capabilities of the global path with incoming sensor data.

Task-Level Control

One effective way to interact with the Lunar Rover is in a semi-autonomous mode. The idea is for a human operator to use a virtual reality interface, such as the one developed at NASA Ames [1], to view the area surrounding the rover and to indicate preferred directions for the rover to follow. This type of interface has been implemented using topographic site maps, in order to facilitate planning and commanding large-scale routes for the rover to follow, and monitoring rover progress over terrain.

LUNAR ROVER CONFIGURATION

Although the Ratler has served as an effective testbed for terrestrial demonstrations, its configuration does not address a number of central concerns for operating on the Moon. We have confronted these issues in the preliminary configuration of a next-generation rover, to be operational in 1995.

The study focussed on the mechanism, power, thermal, and communication link [3]. The result is a six-wheeled 250 kg class rover (Figure 3) with active, two-axis pointing of the solar array to the Sun and the antenna to Earth, providing 400 W of power, and about 1.5 Mb/s downlink to Earth. The rover will hibernate during the night. The primary challenges in lunar rover design have proven to be

- Return continuous video with minimal interruption
- Accomplish an unprecedented 1000 km traverse spanning two years of operation in the extreme conditions on a surface of fine electrostatic dust.
- Survival in radiation, -180 deg C cold, vacuum, and operations in the heat of +130 deg C

A second stage of configuration is currently focussing on software requirements, computing, visualization, and mechanism analysis.

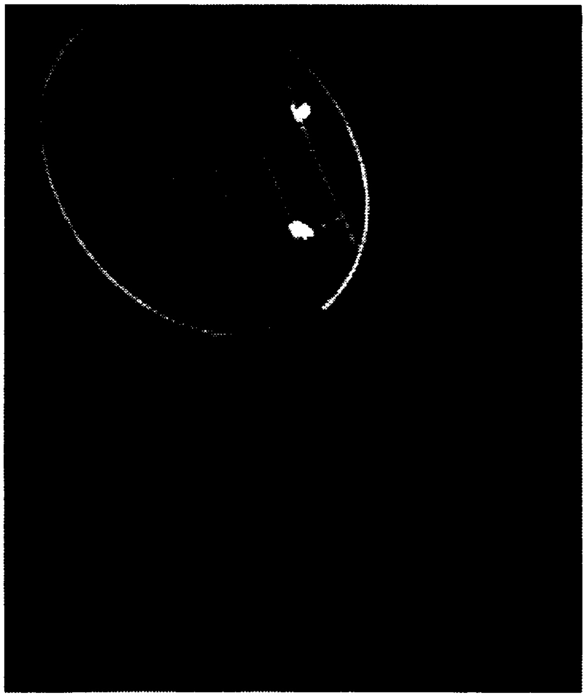


Figure 3 Scale model of preliminary configuration

SUMMARY

In this paper, we have described the Lunar Rover Initiative as a broad-based activity aiming to launch a lunar mission within the millennium. We have concentrated on two concrete technology demonstration projects advancing the initiative: the Dante/Mt. Spurr project emphasizing planetary analogue terrain and remote operation, and the Lunar Rover Demonstration project emphasizing large-scale navigation.

REFERENCES

[1] B. Hine. A Shared Virtual Environment for Robotic Vehicle Control. In Proc. IAF Symposium on Space Systems, 44th International Astronautical Congress, Paris, France, October 1993.

[2] E. Krotkov, M. Hebert, M. Buffa, F. Cozman, L. Robert, "Stereo Driving and Position Estimation for Autonomous Planetary Rovers", In Proc. IARP Workshop on Space Robotics, Montreal, Quebec, July, 1994.

[3] L. Katragadda, et al.,"Lunar Rover Initiative- Preliminary Configuration Document", Technical report CMU-RI-TR-94-09, Carnegie Mellon University, 1994.

SESSION RP

Robotics Programs and Projects

A PERSPECTIVE ON SPACE ROBOTICS IN JAPAN

N95- 23695

Yoshiaki Ohkami*, Ichiro Nakatani**, Yasufumi Wakabayashi+ and Tsutomu Iwata+

*Dept. of Mechano-Aerospace Engineering,Tokyo Institute of Technology
2-12-1 Ohokayama, Meguro,Tokyo 152 JAPAN
e-mail:ohkami@mes.titech.ac.jp
Tel:81-3-5734-2644 Fax:81-3-3729-0628

**The Institute of Space and Astronautical Science(ISAS)
3-1-1 Yoshinodai, Sagamihara, Kanagawa-ken 229 JAPAN
e-mail:nakatani@net202.newslan.isas.ac.jp
Tel:81-427-51-3911 Fax:81-427-51-3988

+Systems Engineering Department, National Space Development Agency
2-1-1, Sengen, Tsukuba, Ibaraki 305 JAPAN
e-mail:ywakaba@rd.tksc.nasda.go.jp
Tel:81-298-52-2241 Fax:81-298-52-2247

ABSTRACT

This report summarizes the research and development status and perspective on space robotics in Japan. The R & D status emphasizes the currently on-going projects at NASDA including the JEM Remote Manipulator System (JEMRMS) to be used on the Space Station and the robotics experiments on Engineering Satellite VII (ETS-VII). As future perspective, not only NASDA but also ISAS and other government institutes have been promoting their own research activities on space robotics in order to support widely spread space activities in future. Included are an autonomous satellite retrieval experiment, dexterous robot experiment, on-orbit servicing platform, IVA robot and moon/planetary rovers proposed by NASDA or ISAS and other organizations.

1. INTRODUCTION

NASDA started the development of JEMRMS in 1987 and ETS-VII project in 1993. The ETS-VII robotics experiments will be carried out in 1977 and the JEMRMS in 2000 and the developments of these two robots are in progress. Space robotics is considered one of the most important technologies in space research and development. This is endorsed by a report recently submitted by the Committee at Science and Technology Agency on the long-term vision for space development (English version not available yet). In this report, space robotics is referred as crucial for future space exploitation beyond the turn of the century especially for moon/Mars missions. In addition to the concerns inside the space community, there are many researchers interested in space robotics as a technical challenge in the area of robotics.

This report summarizes the space programs underway that are directly related to robotics, and secondly overviews the concept studies focusing on the space robotics inside the representative space organizations.

2. SPACE ROBOT RELATED PROJECTS

The space programs that are related to space robotics directly are the following three developed by NASDA: 1)JEMRMS, 2)JFD, and 3)ETS-VII.

2.1 JEMRMS

The manipulator attached to the Japanese Experiment Module (JEM) of the international space station is called JEM Remote Manipulator System (JEMRMS). The JEMRMS consists of a 10-meter main arm and a 1.6-meter small fine arm (SFA). Both arms have 6-DOF and are controlled by on-board crew using two 3-DOF hand controllers. The JEMRMS is currently scheduled to be launched in 2000 with the JEM pressurized module. The baseline configuration of JEM and JEMRMS is shown in Fig. 1.

2.2 JFD

The objective of the JEM Flight Demonstration (JFD) is to verify on-orbit maintainability of the JEM subsystems using JEMRMS and to provide an opportunity for operational experience for on-board crews and ground operators.

A sub-model of JEMRMS SFA will be launched on the shuttle cargo bay and controlled by on-board crews from the aft flight deck. Tasks

118

such as ORU exchange will be performed. The launch date of the shuttle with the JFD system is currently scheduled in 1997. Fig. 2 shows the concept of the JFD system.

2.3 ETS-VII

The main purpose of Engineering Test Satellite-VII (ETS-VII) shown in Fig.3 is to acquire the basic technology regarding rendezvous docking and space robotics. ETS-VII will be launched in 1997.

3. RESEARCH ACTIVITIES ON SPACE ROBOTICS

A number of researchers are now interested in space robotics in the near future, because there are various technical challenges in this field. NASDA is responsible for R & D in space applications with H-II rocket as launch vehicle, while ISAS is for scientific exploitation with M-V rocket. These two are leading organizations under the coordination of the Space Activities Committee. For future space robotics missions, there should be a tighter cooperation in some cases. In what follows, typical research topics are listed but this is not exhausted or authorized yet.

3.1 Dexterous Robot Experiment Using JEM

A dexterous robot concept is studied in NASDA in order to perform a potion of an astronauts' activities and to enhance on-orbit servicing capability in unmanned space systems. The JEM dexterous robot experiment will study and verify dexterous robot technologies using the JEM exposed facility. The implementation of the experiment is currently targeted in the first decade of the year 2000. Fig. 4 shows the basic concept of the experiment system.

3.2 On-orbit Servicing Platform

An unmanned platform, on which micro-gravity experiments, earth observing missions and other engineering experiments will be conducted, is currently being studied in NASDA. The platform has a robot system on board. Experiment samples and replacement units will be carried to the platform by supply satellites, and a robot arm on the platform will transfer and exchange them. Processed samples will be brought back to the earth by capsules. The concept of the system is shown in Fig. 5.

3.3 Autonomous Satellite Retrieval Experiment

Autonomous Satellite Retrieval Experiment (ASREX) is proposed by ISAS and refers to a space experiment for retrieving a floating object with tumbling motion using a manipulator aboard a satellite. The tumbling object can be a disabled spacecraft which need a repair operation. The proposed experiment will proceed as follows (see Fig. 6):

(1) A chaser is inserted into low earth orbit.
(2) A dummy target satellite is separated from the chaser using a manipulator aboard the chaser. The target is completely passive without any control capability which simulates a satellite whose functions have stopped.
(3) When the distance between the target and the chaser gets about 20[km], the chaser searches for the target using a laser radar and tracks it.
(4) The chaser makes a rendezvous with the target using the onboard guidance computer.
(5) After the chaser comes within 20[m] from the target, the relative and the relative attitude are estimated by processing the camera images.
(6) The onboard manipulator is autonomously operated and the hand grabs the target.
(7) After controlling the tumbling motion of the target, the manipulator retracts it using a force control algorithm.

For the realization of the autonomous system, many technical issues are to be investigated, and some of them are outlined in section 4.

3.4 IVA ROBOT

To maximize the on-orbit crew time, a concept of an Intra Vehicular Activity (IVA) robot which performs experimental activity or house keeping is an idea. Friendliness to crew members and an interface to experiment equipments are considered to be key technologies for IVA.

4. Research Areas for Future Space Robotics

Major research areas and technology issues related to the future advanced space robotics are listed in the in-house study of ISAS and NASDA. The following items are some of the research areas defined in these studies.

4.1 Laser Radar[1][2]

To hold a tumbling target satellite, the information on 3 dimensional position and attitude of the target with respect to the chaser is needed. A laser radar is the most promising candidate for this purpose. ISAS and NASDA have conducted a basic study on the long range scanning type laser radar which covers 20[km] range. A breadboard model was developed and evaluated.

4.2 Visual Data Processing[2][3][4]

By processing the image data from the CCD (Coupled Charged Device) camera, the 6 degrees of freedom rotational and translational motion can be identified. Several methods have been proposed, two of which are as follows:

(a) Four of the corner cube reflectors are arranged on each of the target surfaces and by processing the reflected images of the laser radar, the particular surface which faces the chaser is identified. Then, using extended Kalman filter technique, the relative attitude and the relative position are estimated.

(b) Reflective markers are arranged along each of the straight edges of the target and the edges are detected by processing CCD camera images[3]. We have introduced a new algorithm for edge detection, where the CCD camera image is partitioned into small areas and the two parameters which describe the straight line in Hugh transformation are statistically processed. In this method, a partially occluded or distorted line can easily be detected. Also, a new method for describing the rotational motion of the target without axis symmetry has been introduced where a rotation with complex nutation motion can be approximately described by superposition of conical motions. The extended Kalman filter is applied based on this simplified model.

4.3 Space Manipulator Control[5][6][7]

Several manipulator control schemes have been proposed.

(a) By introducing sliding mode control for grabbing a tumbling target satellite, the computation time is significantly reduced while the stability is guaranteed[5].

(b) After the completion of catching the tumbling target, the manipulator tries to retract it within allowable force. For this purpose, a new type of force control scheme using a sliding mode control is proposed[6]. Also, a new idea of redundancy in sliding mode control is introduced.

(c) When catching the target, much more time is consumed in visual data processing than in the calculation of control and so the degradation in control accuracy for tracking a moving object mainly comes from the delay in the former. Hence, we have introduced prediction for setting the target position and attitude[7]. This has significantly reduced the achieved control accuracy.

4.4 Physical Simulation Using Test Bed[8]

ISAS has developed a 9 DOF (degree of freedom) space robotics simulator for the purpose of conducting a physical simulation of the ASREX on the ground[8]. The drawing and the picture of the robotics simulator are shown in Fig. 7 and Fig. 8 respectively. This simulator has 3 DOF for rotation motion for each of the chaser and the target and another 3 DOF for relative translation motion. The dynamics of the target and the chaser with the manipulator is solved by one of the three workstations and the 9 motors of the motion simulator are driven by the result. The control of the manipulator is carried out by another workstation while the image data is processed by the remaining one. The system configuration of the simulator is shown in Fig.9. The main features of the simulator are also summarized in Table 1.

5. Planetary Rover

Around the turn of the century, Mars exploitation is considered to be initiated with unmanned Mars rovers. NASDA and ISAS have been conducting concept studies on small and simple rovers launched by H-II rocket(rover weight is 450 [kg]) or M-V rocket(rover weight is 100 [kg]). The following summarizes the results of these studies.

5.1 Mission Analysis

(1)Engineering Missions

Main objective of the planetary rover is to establish various engineering techniques for future deep space missions such as :

(a) Soft landing techniques using AI (Artificial Intelligence) to avoid obstacles which could potentially be found at the landing site,
(b) Navigation techniques for autonomous planetary rover,
(c) Tele-operation techniques for rover and instruments with time delay due to radio propagation,
(d) Image processing techniques,
(e) Weight reduction technique for the main structures

and the instruments.

(2) Science Missions

Candidates for the science missions are as follows:

(a) Geology by photo images to provide for topographical survey, size and shape of rocks, composition of rocks, craters etc.
(b) Element analysis of age using mass-spectrometer, element analysis using X-ray spectrometer, or g-ray spectrometer, study of mineral composition using visible or infrared reflection spectrometer etc.
(c) Wide Area Investigation for magnetic anomalies using magnetometer, gravity anomalies, electromagnetic structure of the crust using VLF, seismological observation using seismometer network etc.
(d) Investigation by Manipulator such as analysis of regolith, measurement of heat flux, element analysis etc.

5.2 System Overview

Various locomotion systems have been studied and 4-wheel system has been selected, because 4 wheels have advantages over caterpillars or articulated legs in terms of weight, simplicity and speed. As for drive motor, a brushless DC motor has advantages in terms of maintenance and life. A harmonic drive gear is used for deceleration. This locomotion system has ability to climb 30 degree slope. The speed of the rover is about 1 [km/hour] and the moving distance is about 1,000 [km/year].

5.3 Research Areas for Rover

Planetary rover covers a very wide variety of research areas. Followings are some of the research items.

(1) Path Planning[10][11]

A planetary rover is required to travel safely over a long distance for many days in unknown terrain. One of the important functions for planetary rover is to plan a path from a start point to a goal without hitting obstacles. A new path planning scheme has been proposed. The model of a rover is introduced to consider the size of the rover. This model can be easily modified into any other architecture. The planetary rover makes an elevation map by observing the environment. We have newly proposed EEM[Extended Elevation Map], which includes the effect of the size of the rover.

(2) Position Estimation[12][13][14]

A planetary rover needs to identify its position to reach a goal. Dead reckoning is one of the most widely used methods, which, however, has a drawback of inaccuracy due to the slipping of the rover tires. To supplement dead reckoning, we have proposed several methods as follows.

(a) The position and direction of the rover is obtained by observing the sun. Least squares method is used to estimate the position. This method has a position accuracy of about 1.0 [km], but during a long term trip, say for 6 months, this is very advantageous due to non-accumulation of errors.

(b) Three types of new map matching methods for 3-D terrain are proposed: differentiation map matching, altitude difference map matching and triangle map matching. The former two methods can be classified as template matching. where as the last method as structure matching. In these methods, terrain map information is used, which is derived from a laser range finder. The validity of the proposed methods is verified by computer simulations and experiments.

6. Conclusion

A brief summary is presented for the development status of space robot in Japan and for the research activities, mainly conducted by ISAS and NASDA, on orbiting spacecraft with robotics and planetary rovers.The authors wish that this article describs a very active research fields in Japan.

References

[1] H.Saito, I.Nakatani, K.Ninomiya, A.Furuta, "Scanning Laser Radar System for Rendezvous and Docking in Space," Proc. of 38th Congress of the Int. Astronautical Federation, 1987.
[2] I.Nakatani, T.Tanamachi, K.Ninomiya, "Satellite Motion Analysis via Laser Reflector Pattern Processing for Rendezvous and Docking," Proc. of 37th Congress of the Int. Astronautical Federation, 1986.
[3] S.Azuma, I.Nakatani, K.Ninomiya, "A New Method for Autonomous Retrieval of a Satellite Using a Visual Sensor and a Manipulator," Proc. of 40th Congress of the Int. Astronautical Federation, 1989.
[4] K.Ninomiya, I.Nakatani, T.Wakayama, "3D Motion

Estimation of an Object in Space by Image Processing" Proc. of 30th Annual Conf. on SICE, 1991.

[5] K.Harima, J.Kawaguchi, I.Nakatani, K.Ninomiya, "Control of Space Manipulator Using Sliding Mode Control", Journal of Robotics Society, Vol.9-5, pp.572-579, 1991.

[6] K.Harima, J.Kawaguchi, I.Nakatani, K.Ninomiya, "Control of Redundant Space Manipulator Using New Sliding Mode", Journal of Robotics Society, Vol.9-7, pp.821-829, 1991.

[7] H.Nagamatsu, T.Kubota, I.Nakatani, "Experimental Study for Autonomous Retrieval of Tumbling Satellite by Space Manipulator", Proc. of 3rd Workshop on Astro-Dynamics and Flight Mechanics, pp.6-11, 1993.

[8] K.Ninomiya, et.al. , "Development of 9-DOF Space Robotics Simulator", Proc. of the 34th Annual Conf. on Space Sciences and Technology, pp.440-441, 1991.

[9] T.Iwata, I.Nakatani, "Overviews on the Japanese Activities on Planetary Rovers," Proc. of 43rd Congress of the Int. Astronautical Federation, 1987.

[10] T.Kubota, I.Nakatani, T.Yoshimitsu, "Path Planning for Planetary Mobile Robot", Proc. of ROBOMEC'94, 1994.

[11] I.Nakatani, T.Kubota, T.Yoshimitsu, "Path Planning for Planetary Rover Using Extended Elevation Map", i-SAIRAS'94, 1994.(to be presented)

[12] I.Nakatani, T.Kubota, T.Yoshimitsu, "Position Estimation for Planetary Rover by Observation of the Sun ", Proc. of 37th Annual Conf. on Space Sciences and Technology, pp.369-370, 1993.

[13] K.Ninomiya, I.Nakatani, T.Endo, "Map Matching Methods for Planetary Rover" Proc. of 32nd Annual Conf. on SICE, 1993.

[14] I.Nakatani, T.Kubota, N.Tanida, "Position Identification for Planetary Rover by Map Matching Using Characteristic Points" Proc. of 33rd Annual Conf. on SICE, 1994.

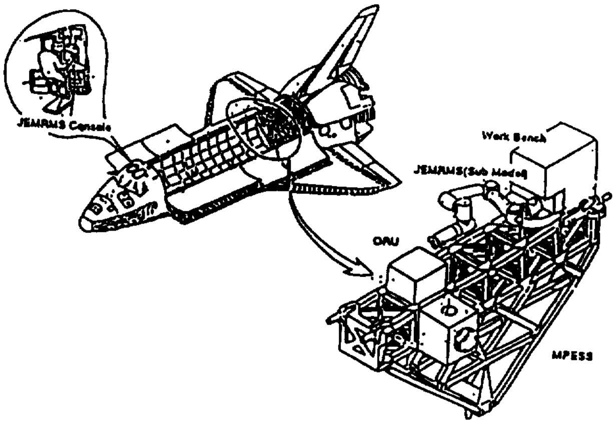


Fig. 2 JFD

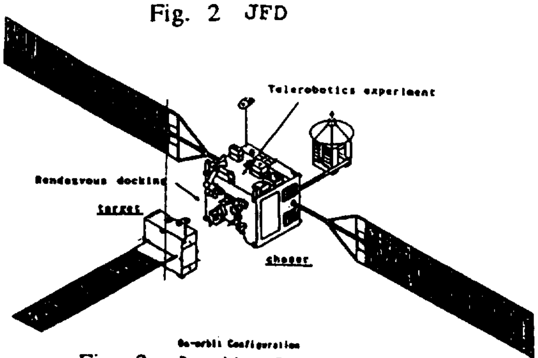


Fig. 3 Baseline Configuration of ETS-VII

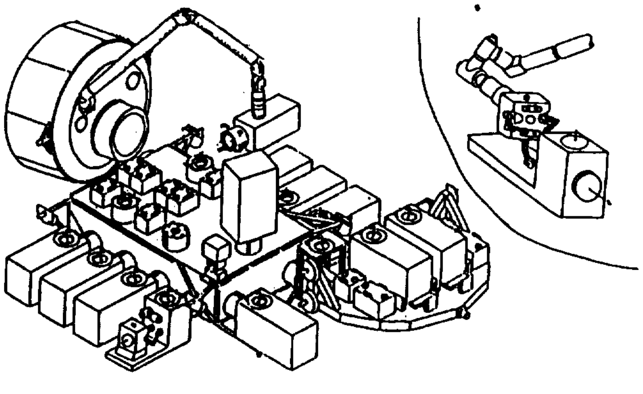
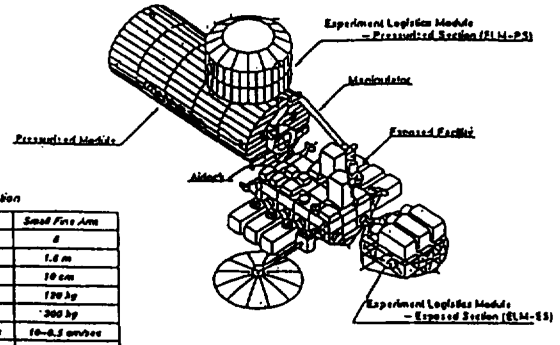


Fig. 4 Dexterous Robot Experiment Concept



JEMRMS specification

	Main Arm	Small Fine Arm
D.O.F.	6	6
Arm Length	9.7 m	1.6 m
Arm Beam	30 cm	10 cm
Weight	370 kg	120 kg
Payload	7000 kg	300 kg
Tip Speed	5–0.5 cm/sec	10–0.5 cm/sec
Control Mode	Program Control Joystick Control	Program Control Joystick Control

Fig. 1 JEM and JEMRMS

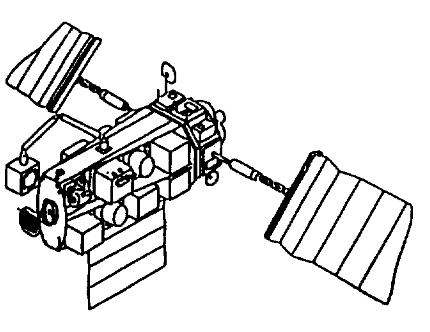


Fig. 5 On-orbit Servicing Platform

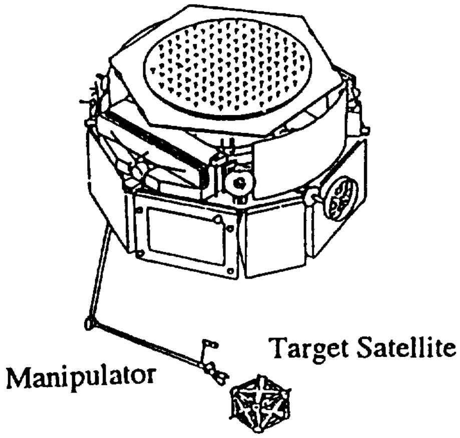


Fig. 6 Concept of ASREX

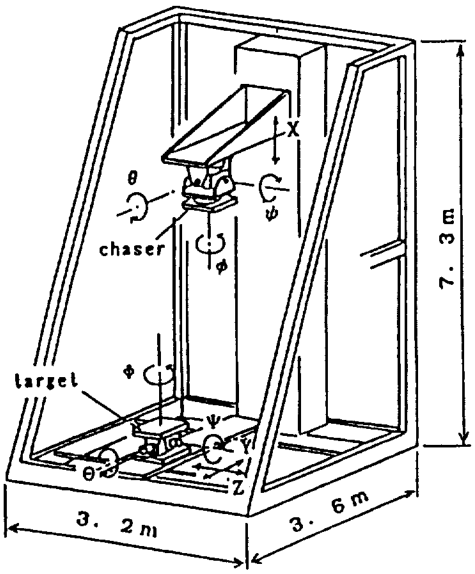


Fig. 7 Overview of robotics simulator

Table 1. Specification of simulator

Target Mount : payload weight 20[kg]				
direction	motion range	precision	maximum velocity	maximum torque
y	2.0 [m]	0.5 [mm]	0.2 [m/s]	150 [kgf]
z	2.0 [m]	0.5 [mm]	0.2 [m/s]	150 [kgf]
Φ	±180 [deg]	0.5 [deg]	30 [deg/s]	35 [Nm]
Θ	± 40 [deg]	0.2 [deg]	20 [deg/s]	50 [Nm]
Ψ	± 40 [deg]	0.2 [deg]	20 [deg/s]	50 [Nm]
Chaser Mount : payload weight 50[kg]				
x	4.0 [m]	1.0 [mm]	0.3 [m/s]	100 [kgf]
ϕ	±180 [deg]	0.5 [deg]	30 [deg/s]	50 [Nm]
θ	± 20 [deg]	0.2 [deg]	20 [deg/s]	75 [Nm]
ψ	± 20 [deg]	0.2 [deg]	20 [deg/s]	75 [Nm]

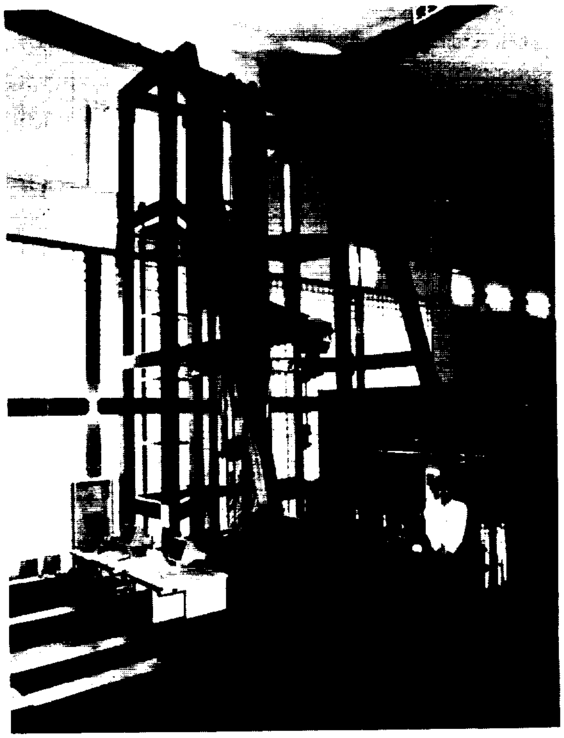


Fig. 8 Robotics Simulator

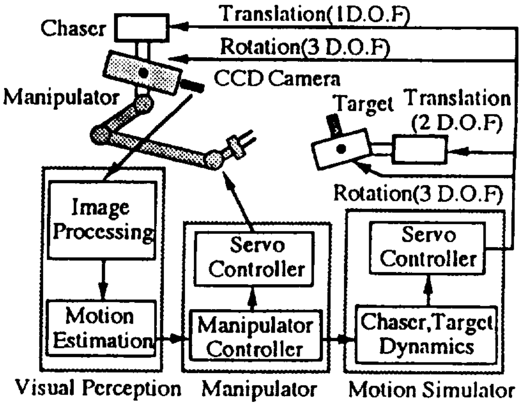


Fig. 9 System configuration of simulator

ASI's Space Automation & Robotics Programs: the Second Step

N95- 23696

Simonetta Di Pippo
Italian Space Agency
Via di Villa Patrizi, 13
00161 Rome, Italy
ph: +39 6 8567408 fax + 39 6 4404212
E. mail:Di Pippo@ASIMT0. MT. ASI. IT

KEY-WORDS AND PHRASES

ASI, Automation, Robotics, SPIDER

INTRODUCTION

The strategic decisions taken by ASI in the last few years in building up the overall A&R program, represent the technological drivers for other applications (i.e. internal automation of the Columbus Orbital Facility in the ESA Manned Space program, applications to mobile robots - both in space and non-space environments, etc...). In this context, the main area of application now emerging is the scientific missions domain.

The ASI strategy has been based on the following main guidelines [1]:

- **Long-term program**
  SPIDER: SPace Inspection Device for Extravehicular Repairs
- **Robot/Telerobot Control System Architecture**
  SAREM: SPIDER Architecture REference Model
- **Technological program**
  SARTDP: SPIDER Automation and Robotics Technological Demonstration Program
    - SPIDER manipulation System
    - TV-Trackmeter
    - Advanced man-machine interface

  BARTEX: Balloon for Automation and Robotics TEchnological Ex.
- **A&R Support and Testing facilities**
  CSR: Centre for Robotics Simulation
  ST-Lab: Sensor Testing Lab. for Space Robotics
- **Planetary Rovers**
  ARPE: Autonomous Rovers for Planetary Exploration
  IMEWG: International Mars Exploration Working Group
- **Italian Cooperation with ESA programs**
  ROSE-D: RObotics SErvicing Demonstrator
  AMTS: Automated Manipulation and Transportation System
  EUROMIR '95: Robotic Exp.
  ROSETTA Surface Science Package
  Moon Lander and Rovers

Due to the broad range of applications of the developed technologies, both in the in-orbit servicing and maintenance of space structures and scientific missions, ASI foresaw the need to have a common technological development path, mainly focusing on:

- control
- manipulation
- on-board computing
- sensors
- teleoperation

Before entering into new applications in the scientific missions field, a brief overview of the status of the SPIDER related projects is given, underlining also the possible new applications for the LEO/GEO space structures.

NEW ACTIVITIES IN THE FRAME OF SPIDER AND RELATED PROJECTS

The SPIDER New Phase

In the last years, ASI made great investments on A&R in space, due to growing importance of internal and external in-orbit servicing, maintenance and operations. In this context, ASI started a long-term program named SPIDER (SPace Inspection Device for Extravehicular Repairs) [2] and a Technological Program in order to support and guarantee system assembling with state-of-art technology. SPIDER is a free-flying space robot, designed to operate in external environment of manned and unmanned orbiting structures, both in LEO and GEO.

The phase B, now starting, beside the redefinition of the SPIDER operational missions in view of the changed world space scenario, will implement two of the major SPIDER system development stages identified in the phase A [3]

- the development and setup of a dedicated ground test-bed to perform hardware-in-the-loop tests for supporting the development and tuning of the items and technologies (rendezvous sensors, image processing, flyaround techniques, arm operation, grasping tools, etc...) enabling an autonomous rendezvous and capture of non-cooperative target, in a simulated space environment [see fig. la.,b].
- the design of a technological on-orbit demo mission (and of possible precursor tests in low-gravity environment - see BARTEX) and characterization of the SPIDER system performances in rendezvous and capture operations, in the real space environment and in a situation similar to an actual operational mission [see fig. 2].

SPIDER Manipulation System

The development of the SPIDER manipulation system is currently scheduled in three phases and will conduct to the engineering model of a bi-arm manipulation system, with the capability to operate both in robotics and teleoperated mode, provided with collision avoidance, vision and proximity sensors and with a co-operative bi-arm control capability. The first phase, which will end in mid '95, concerns with the development of the engineering model of a 7 d.o.f. robotic arm, belonging to the 1.5 meter length class and the breadboard of its controller [4].

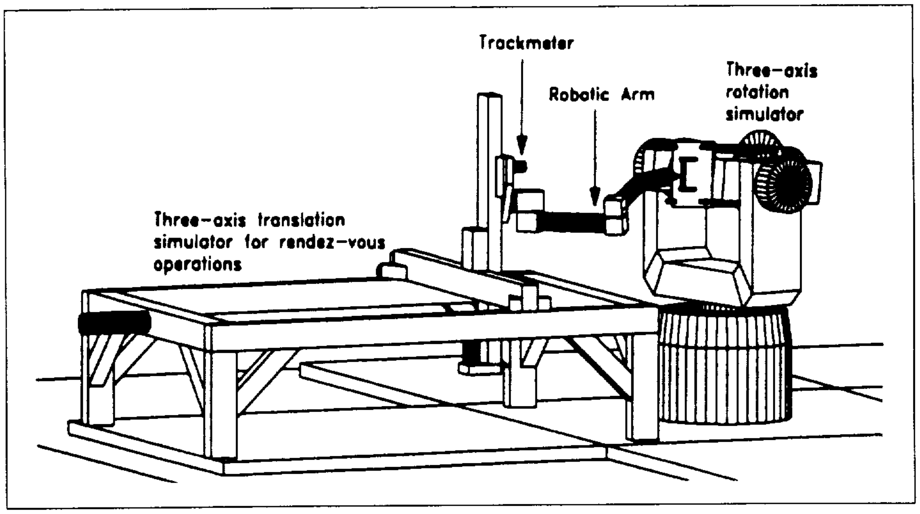


Figure 1a. SPIDER test-bed for rendezvous and arm alignment manoeuvre simulations

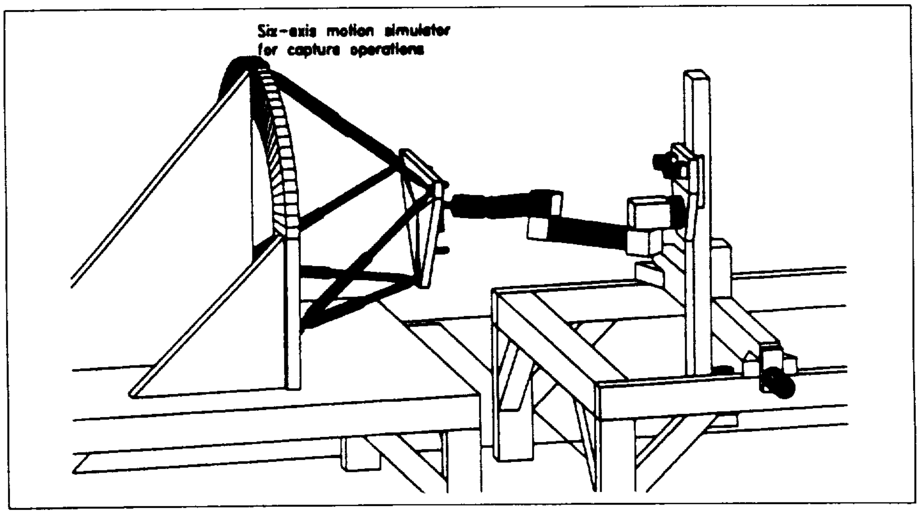


Figure 1b. SPIDER test-bed for the capture operation simulations

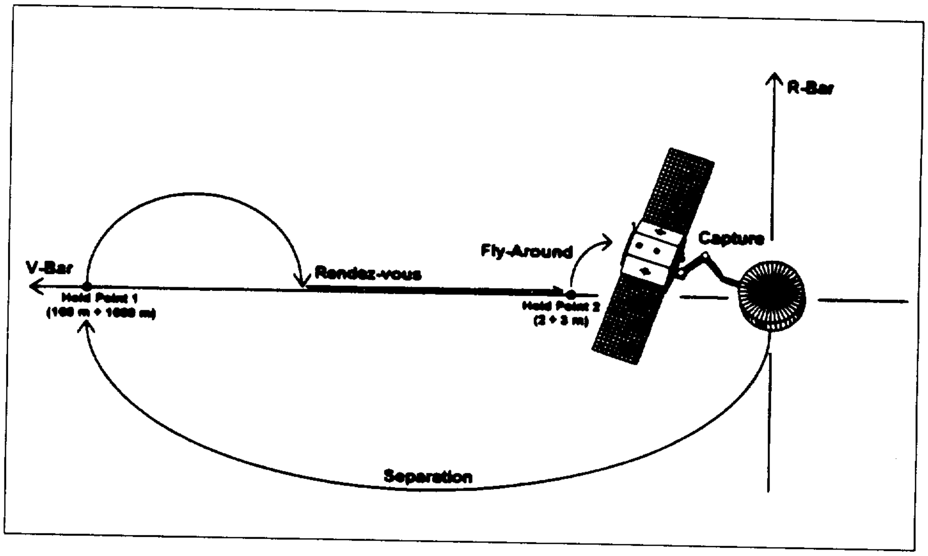


Figure 2. Example of operation profile of the SPIDER Technological demo mission

The manipulation arm is provided with a parallel gripper type end-effector and force/torque sensors. The kinematics configuration of the arm is shown in fig. 3. Two types of arm internal configuration were investigated in detail, specifically:

- internal configuration based on "Distributed joint approach" which implies cable passage in the out-skirt of the actuator bulk,
- internal configuration based on "Integrated joint approach" which implies cable passage in actuator central allowed shaft.

The arm basic design, after detailed analysis, foresees six out of seven axes based on the "distributed approach" and the seventh axis based on the "integrated approach".

The "distributed" joint will be tested next year; in fact, in the framework of the ESA Columbus precursor flights program-EUROMIR 95 mission, the approved Italian "In-Orbit Robotic Technology Experiment" is aimed at verifying, in actual 0g environment, the main performances of the breadboard robotic joint/technology already developed in the frame of the SPIDER contract, under ASI responsibility.

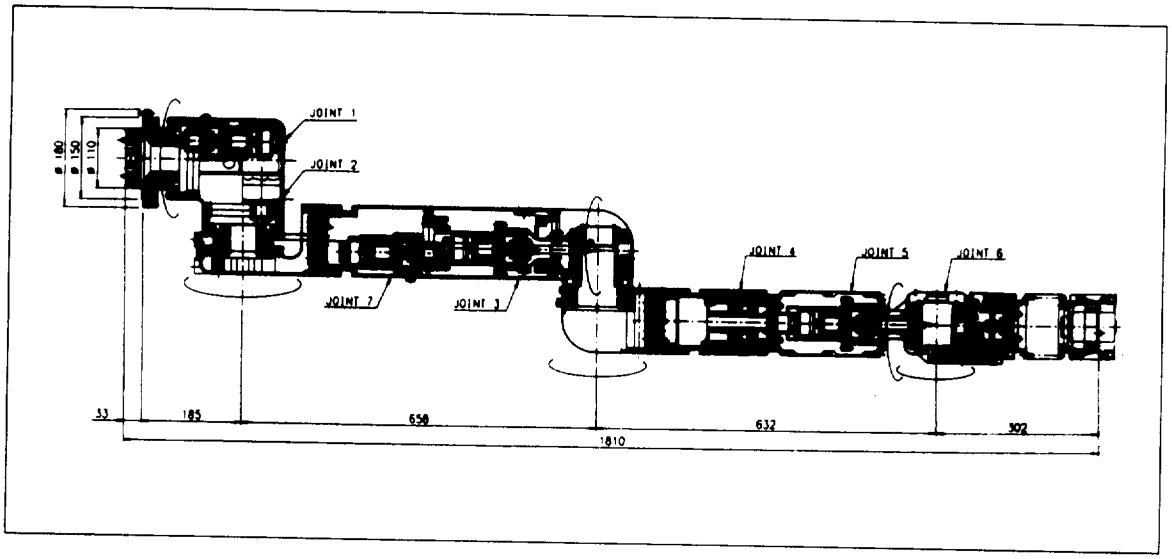


Figure 3. Kinematics configuration of the SPIDER arm

The in-orbit experimental phase will be structured to allow the testing of variants of the reference control algorithms with different gains. The proposed experimental verification is important also in view of the possible use of the SPIDER robotic arm in the frame of microgravity applications. The experiment concept is described in fig. 4.

Referring to the possible applications, the SPIDER manipulation system development contract has been redirected to take into account the ESA activity named AMTS (Automated Manipulation and Transportation System). The objective of the ongoing AMTS phase B is the detailed definition of the system, including robot arm, gantry, controller and support subsystems. The cross-analysis between AMTS and SPIDER arm requirements showed many commonalities, so that a certain degree of harmonisation between the two programs have been accomplished without additional effort. To be cooperative with ESA activities, ASI performed the evaluation of the micro-G disturbances of the SPIDER arm technology in order to analyse the possible impacts on the AMTS operational conditions.

For what concerns the "integrated" joint technology, ASI started few months ago an internal evaluation for the applicability of the SPIDER arm technology to the Lunar exploration, having in mind the need to reduce drastically the associated mass. The use of the integrated joint could reduce the total mass of the arm, but also can contribute to modify the length of the arm in any desirable shape [see Moon Exploration].

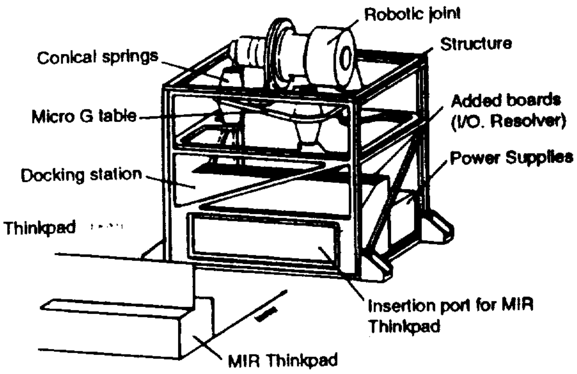


Figure 4. EUROMIR 95 robotic experiment layout

BARTEX

ASI has started an activity called BARTEX (Balloon for Automation & Robotics Technological EXperiment) [5] carrying out A&R technological experiments in a micro-gravity environment obtained within a capsule, lifted up to 40-45 km of altitude by a stratospheric balloon and then dropped down. During its free-fall motion, micro-gravity conditions are obtained inside the capsule.

As the reference experiment, ASI has chosen the Object Capture experiment, aiming at demonstrate the capability of capturing flying objects by means of an integrated telemanipulation-vision system with robotic functions.

The experiment will be performed using existing hardware, developed under ASI contracts (a Chinese copy of the SPIDER manipulator arm and the TV-trackmeter), taking advantage also of the development of the capsule, named GI-ZERO, under a parallel ASI contract.

The first flight opportunity has been selected in the '97 summer. The fig. 5 shows the layout of the BARTEX experiment.

A&R FOR SCIENTIFIC MISSIONS

The main characteristic of Automation and Robotics is to be applicable mainly to all the scientific missions, in particular to the deep space missions and the planetary exploration enterprises.

At the present time, there are several future missions under evaluation in the international framework based on automatic systems and autonomous mobile vehicles.

Looking at the main international enterprises, we will focus mainly on three of them as reference, underlining the primary Italian role in this missions. i.e.:

- Mars Exploration
- Moon exploration
- Deep space missions

Mars Exploration

First of all, the exploration of Mars. In the 1993, the International Mars Exploration Working Group was created by the main spacefaring agencies with the main goal to constitute a forum for discussing the various phases of the exploration and the possible contribution coming from each space agency, member of the group [6].

Taking into account the technological state-of-art, the first phase of the exploration of the Red Planet is based on remote sensing (i.e. without contact with the surface) to gain information for the subsequent phases, the network phase and the automated phase.

In the network mission, up to 12 fixed stations will be released on the Mars surface; ASI has now started the definition activity related to the possible use of microrovers for Mars Geo-Exploration, named MIGEMA (MIcrorovers for Geo-Exploration of MArs). The microrovers are seen as an "extension" of the capability of the fixed stations, allowing the exploration of a few meters around the landing site. Such a microrover could help in performing the following scientific measurements:

- thermal conductivity, using temperature probes to be placed under the martian surface at different depths
- seismic parameters, using seismometers (geophones); these sensors shall be placed some centimeters under the surface
- local radioactivity, using radioactivity probes in the subsurface
- soil consistency, using sensorized drilling tools

The technical feasibility of a 10 kg microrover has been already investigated and assessed by ASI in the activity named ARPE (Autonomous Rovers for Planetary Exploration) [7], conducted with the strong involvement of Russian firms and institutions.

Moon Exploration

The new Moon exploration program, now under evaluation in the frame of the ESA new activities, will follow a progressive phased approach, starting with the initial exploration using small satellites and surface probes, progressing to the use of robots for scientific and resource exploitation and culminating in manned lunar bases [8]. Italy has proposed two possible participations in this enterprise: the so-called "robotic lunar science kit" and the responsibility for the proximity and in situ operations for a mobile robot. Due to the know-how gained by Italy in the area of A&R, the robotic lunar science kit has the main goal, starting from theexisting technologies, to activate lunar surface collection and inspection, to store on-board collected material, performing also a scientific analysis, supporting in addition different scientific and servicing tasks. The idea

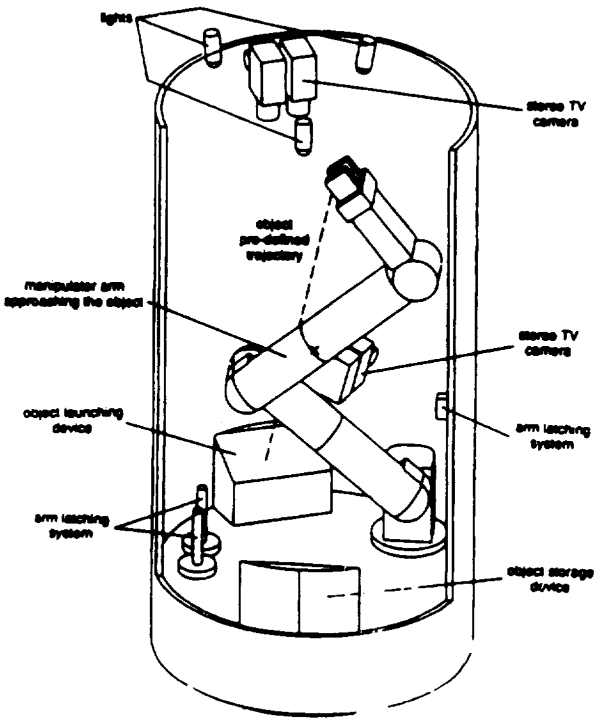


Figure 5a. BARTEX workcell layout and experiment accommodation

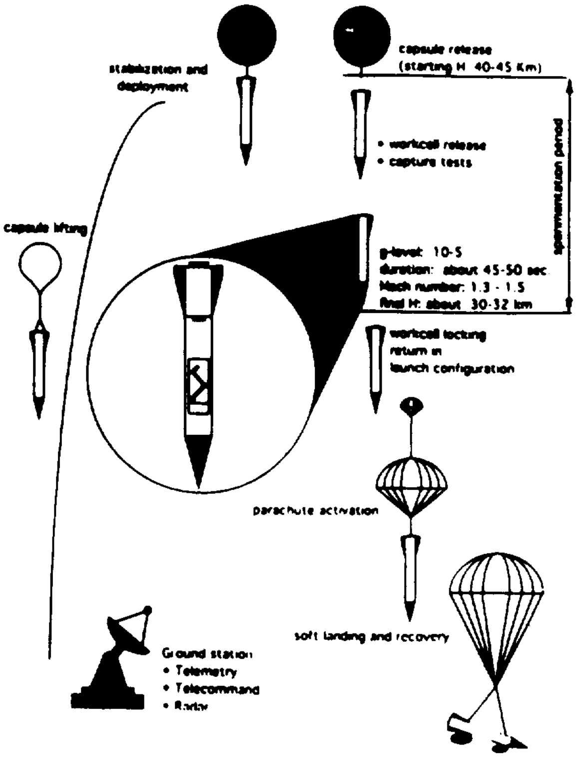


Figure 5b. BARTEX workcell layout and experiment accommodation

to include an Earth return capsule has been also presented. A possible Italian role for the mobile robot has been already discussed, in particular focusing on the in situ analysis and operations, due to the Italian expertise, gained also in the framework of the ESA ROSETTA mission at breadboard level.

The possible re-use of the SPIDER robotic arm is under evaluation (see SPIDER manipulation system).

Deep Space Missions

In the framework of the on-going ESA ROSETTA cometary mission, to be launched in 2003, one of the key elements to be developed is a small lander, to be released on the comet surface, in order to perform nucleus scientific measurements. Due to the Italian expertise, gained both on the national activities described above and on the development activities performed for the ESA Sample Acquisition System critical parts (Corer, Anchor and Surface tools) [9], Italy is claiming to get the primary responsibilities for the "Automated Interfaces" between the Surface Science Package (SSP) and the cometary soil, plus for the activities and subsystems aiming at improving the automation and, therefore, the scientific return of the overall mission [10].

CONCLUSIONS

The paper mainly deals with some of the new ongoing activities in the Italian Space Agency in the field of Automation & Robotics.

Due to the strategy adopted in the past few years, in this second step ASI is stressing its intervention in scientific missions in which the robotic technologies under developed and/or under development in the A&R area can be useful and easily transferred.

Taking into account the complexity of such exploration missions, ASI approach is looking forward in parallel to achieve cooperation agreement with international partners, focusing also on possible joint developments of challenging technologies.

REFERENCES

[1] Di Pippo S.,1994. ASI's Space Automation and Robotics Programs. In *Proceedings of the IAP second workshop of Robotics in Space.* Montreal, 6-8 July

[2] SPIDER phase A contract:final report. ASI contract n.108/AF/90,1991

[3] Di Pippo S., Barraco I. 1994. New Perspectives for the SPIDER project. In *Proceedings of the IAP second workshop of Robotics in Space.* Montreal, 6-8 July

[4] Mugnuolo R., Magnani P.G., Terribile A. ,Gallo E. ,Dario P. 1994. SPIDER: Design and development of the high performance dexterous robotic arm. In *Proceedings of the IARP second workshop of Robotics in Space.* Montreal, 6-8 July

[5] BARTEX phase A final report. ASI contract 1993.

[6] IMEWG activities. Presentations to COSPAR symposium. Hamburg, July 1994

[71 Di Pippo S. et al., 1993. Autonomous Rovers for Planetary Exploration. *Executive summary*, October 1993

[8] A Moon Program: The European View. *ESA BR-101*, May 1994

[9] Costa P.A., Fenzi M., Heiden M. 1993. Collecting cometary soil samples: development of the ROSETTA Sample Acquisition system.

[10] ASI scientific and technological potential involvement in the Surface Science Package. *Presentation to the Workshop on Rosetta SSP.* Estec, 9 February 1994

JPL SPACE ROBOTICS: PRESENT ACCOMPLISHMENTS AND FUTURE THRUSTS

N95- 23697

C. R. Weisbin, S. A. Hayati, and G. Rodriguez
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Tel. (818) 354-2013
Fax (818) 354-7354
e-mail: Charles.R.Weisbin@jpl.nasa.gov

KEY WORDS AND PHRASES

Space Telerobotics, Surface Inspection Rover Technology, Microsurgery, Emergency Response Robotics, Remote Control.

TELEROBOTIC SERVICING

Remote Surface Inspection

Complex missions require routine and unscheduled inspection for safe operation [1]. The purpose of research in this task is to facilitate structural inspection of the planned Space Station while mitigating the need for extravehicular activity (EVA), and giving the operator supervisory control over detailed and somewhat mundane, but important tasks. The telerobotic system enables inspection relative to a given reference (e.g., the status of the facility at the time of the last inspection) and alerts the operator to potential anomalies for verification and action [2]. One example might be the inspection of truss struts for micrometeoroid damage and visible cracks on the thermal radiator surface. Simulation of realistic dynamic lighting is included. In addition, configuration control of manipulators with redundant degrees of freedom has been developed and implemented to assure dexterous manipulation near complex structures [3]. To assure safe operation, collision detection and avoidance algorithms monitor the arm motion.

A multi-sensor end-effector [4] includes a gas sensor for detection of gas leaks and a pyrometer to measure surface temperatures, in addition to CCD cameras. This end-effector also houses two proximity sensors to provide collision avoidance and a force/torque sensor for safe contact with the environment. Algorithms for flaw detection based on real-time image differencing with appropriate registration to account for variable lighting and manipulator/camera position have been developed and validated. A serpentine robot with 12 degrees-of-freedom (external diameter 3.31 cm, 91.44 cm extended length, and less than 2.73 kg) has been developed for use as a tool for inspecting regions with small openings [5]. This tool is to be picked up by the larger robotic arm and placed near small openings for inspection. The serpentine robot carries a fiber optic light/camera system and is self-contained. Several of the developed technologies within this task have successfully been transferred to the Johnson Space Center (JSC) for realistic tests in a high-fidelity robotics laboratory with evaluation by astronauts.

Ground Operator Environment

There are two primary objectives of this project: To develop technologies that enable well-integrated NASA ground-to-orbit telerobotics operations, and to develop a prototype common architecture workstation which implements these capabilities for other NASA technology projects and planned NASA flight applications.

This task develops and supports three telerobot control modes which are applicable to time delay operation: Preview teleoperation [6], teleprogramming [7], and supervised autonomy [8]. Preview teleoperation provides a graphical robot simulation which moves in real time according to the operator's motion input to a hand controller. This same teleoperation motion is sent to the real robotic system for execution. In teleprogramming, the operator's manual interaction with a 3-D virtual environment (physically identical to preview teleoperation) is symbolically interpreted by computer software (e.g. for a grasping operation) to a low-bandwidth, low-level sequence of autonomous commands that are synchronously transmitted to the remote site, which has a simple sensor-referenced behavioral control capability. The best features of dexterous teleoperation are preserved, while providing greatly increased operational robustness against extended (2-10 s) and possibly intermittent time delay. The operator's graphical workstation environment can be automatically updated based upon modeled sensor data feedback from the remote site, and robot sensor data is used instantaneously at the remote site to behaviorally compensate for operator motion errors and positioning uncertainties. Finally, supervised autonomy provides capability to generate high-level autonomous command sequences via either a graphically programmed operator interaction with the modeled environment, or using conventional menus.

Distributed Space Telerobotics

This effort is a cooperative research and development activity between NASA-JPL (Jet Propulsion Laboratory) and the Ministry of International Trade and Industry (MITI)-Electrotechnical Laboratory (ETL) of Tsukuba, Japan. The main technical thrust of the project is safe ground control of orbital robots under operational uncertainties caused by impaired remote viewing, communication time delay, and tasking contingencies. Each of these technological areas manifests itself in respective application interests; the main Japanese application interest is in space assembly, while the U.S. focus is in space servicing.

There are two key research areas currently under development. Intelligent Viewing Control (IVC) involves computerized planning and sequencing of multi-camera views which are fused with calibrated 3-D virtual workspace presentations. This capability includes software facilities for interactive modeling, i.e., the capture of new workspace features, their rendering/presentation, and calibration, intended to improve workspace perception and facilitate camera management. Intelligent Motion Control (IMC) or teleprogramming has already been mentioned in the previous section. The teleprogrammed mode is intended to extend time-delay teleoperation to useful low-Earth-orbit (LEO) applications, and provides a mission resource for contingency tasking in partially structured environments (having geometric uncertainties).

Initial interface specifications have already been developed resulting in successful remote operation of robots in the collaborating country.

Exoskeleton and Telepresence

The focus of this task involves the augmentation to telemanipulation capabilities through the development of human-equivalent dexterity of remotely operated hands, with emphasis on minimal training and use of human rated tools. The technical objective is to prototype a force-reflecting master-slave arm-hand system in exoskeleton form with a 7-DOF (degree-of-freedom) arm and 16-DOF four-fingered hand [9]. This includes integration with a visual telepresence system. The programmatic objective is to determine how far an exoskeleton alternative can perform EVA-glove rated manipulative activities without changing EVA tools or adding new ones to the existing repertoire.

PLANETARY EXPLORATION

Rover Technology Program

Rover technology is enabling for extensive robotic exploration of selected areas of Mars. The rover technology base emerging from this activity has enabled the MESUR/Pathfinder project microrover currently planned for launch in 1996. An active research and development program aimed at significant capabilities beyond Pathfinder microrover is in place at JPL [10-12]. This technology base will greatly expand the current MESUR/ Pathfinder microrover performance in the areas of goal identification, increased vehicle mobility, intelligent terrain navigation with in situ resource management, and manipulation of science instruments. The goal is to combine both research and system demonstrations to advance the state of rover technologies while maintaining flight program relevance. Specific goals over the next four years are: (1) autonomously traverse 100 m of rough terrain within sight of a lander; (2) autonomously traverse 100 m of rough terrain over the horizon with return to lander; (3) autonomously traverse 1 km of rough terrain with execution of select manipulation tasks; (4) complete science/sample acquisition and return to lander with over the horizon navigation. A series of rover vehicles is being used to conduct these tests.

The rover technology program at JPL is being implemented with extensive university and industrial involvement in such areas as: Sensor suites for long-distance navigation on planetary surfaces; legged vs. wheeled mobility; virtual environment operator interfaces; robotic grasping devices; and behavior based obstacle avoidance and fault tolerance.

NASA TERRESTRIAL APPLICATIONS AND COMMERCIALIZATION

Robot-Assisted Microsurgery

Through a cooperative NASA-Industry effort, the Robot-Assisted Microsurgery (RAMS) task develops a dexterity-enhanced master-slave telemanipulator enabling both breakthrough procedures in micro/minimally invasive surgery [13]. The applicable medical practice includes eye, ear, nose, throat, face, hand, and cranial surgeries. As part of planned task activities, the resulting NASA robot technologies will be benchmarked in actual operating room procedures for vitreous retinal surgery.

The primary objective of this task is to provide an integrated robotic platform for master-slave dual-arm manipulation operational in a one-cubic-inch work volume at features in the 100-micron range (our goal is to extend these capabilities to features in the 20-micron range). The research is a natural evolution of our extensive experience in force-reflecting teleoperation with dissimilar master/slave. Capabilities will include force-reflection and textural tactile feedback, and in situ multiple-imaging modalities for improved surgical visualization and tissue discrimination. Potential NASA applications may include EVA/IVA (intravehicular activity) telescience, bioprocessing, materials process and micromechanical assembly, small-instrument servicing, and terrestrial environmental testing in vacuum.

Emergency Response Robotics

Following four years of effort, this project has prototyped a teleoperated mobile robot enabling the JPL HAZMAT (hazardous material) response team to remotely explore sites where hazardous materials have been accidentally spilled or released rather than risk entry team personnel [14]. JPL robotic researchers, engineers, Fire Department and Safety personnel have worked in close cooperation to develop the system. The primary mission of the robot, called HAZBOT, is first entry and reconnaissance of an incident site; the most dangerous part of a response since the type of materials involved and the magnitude of the spill may not be fully known. During such missions HAZBOT must first gain entry into the incident site. This may involve climbing stairs, unlocking and opening doors, and maneuvering in tight spaces. Once the spill is located, an onboard chemical gas sensor is used for material identification. The robot can also be used to aid in remediation or containment of the incident by, for instance, closing a leaking valve, deploying absorbent material, or placing a broken container in secondary containment. HAZBOT has been specially designed to enclose all electrical components and provide internal pressurization, enabling operation in atmospheres that contain combustible vapors. Other system features include a track drive base with front and rear articulating sections for obstacle/stair climbing, a six-DOF manipulator with five-foot reach and 40-pound payload capacity, custom tools for unlocking and opening doors, and 2-color CCD cameras. To date, the robot has been used by the JPL HAZMAT team in three simulated response missions to test and demonstrate system capability. HAZBOT is currently being prepared for actual field use, responding to HAZMAT incidents at JPL. Future work includes the integration of onboard sensors, as well as improvement to the operator control station.

Satellite Test Assistant Robot (STAR)

STAR is a remote inspection robot which has been developed to assist engineers in the ground testing of spacecraft in simulated space environments. STAR is designed to operate inside JPL's 10-ft and 25-ft thermal/vacuum test chambers where temperatures range from -190° C to $+100^\circ$ C and extremely high vacuums can be achieved. STAR consists of a 25-ft vertical axis and an azimuthal axis which provides mobility around the inside diameter of the chamber. A 2-axis scanning platform is instrumented with two high-resolution video cameras, controlled lighting and an Infrared Imaging Camera.

At an Operator Control Station engineers remotely control the position and orientation of STAR's lighting and camera instrumentation allowing close-up real-time visual inspection and infrared thermal mapping of a spacecraft under test in the simulated space environment inside the chamber. STAR will help engineers by improving test reliability and reducing overall test costs.

ACKNOWLEDGMENT

The research described in this paper was performed by the Jet Propulsion Laboratory, of the California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

A comprehensive list of references, not included here due to length considerations, can be found in SPACE TELEROBOTICS and ROVER RESEARCH AT JPL, C. Weisbin, S. Hayati, G. Rodriguez, ANS 6th Topical Meeting on Robotics and Remote Systems, February 5–10, 1995, Monterey, CA.

A U.S.-Japan Collaborative Robotics Research Program

N95- 23698

Paul S. Schenker
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive/ MS 198-219
Pasadena, CA 91109 U.S.A.
Email: schenker@telerobotics.jpl.nasa.gov
Phone: (818) 354-2681
FAX: (818) 393-5007

Shigeoki Hirai
Electrotechnical Laboratory, MITI
1-1-4 Umezono, Tsukuba-shi, Ibaraki, 305
JAPAN
Email: hirai@etl.go.jp
Phone: +81-298-58-5976
FAX: +81-298-58-5989

KEY WORDS AND PHRASES

Robotics, teleoperation, intelligent control, graphics user interfaces, space servicing.

INTRODUCTION

The Jet Propulsion Laboratory (JPL) and the Electrotechnical Laboratory (ETL) have recently initiated a cooperative R&D effort in telerobotics. This new effort, sponsored by the U.S. National Aeronautics and Space Administration (NASA) and Japan's Ministry of International Trade and Industry (MITI), has two major themes. First, our work broadens the outreach of space telerobotics R&D to international technical collaboration and facilities usage in the United States and Japan. This is natural, given plans for a common U.S.-Japan robotic presence on the International Space Station (the Japanese Experimental Module and U.S. Mobile Servicing Center), as well as ongoing U.S-Japan discussions of possible shared ground control assets. Second, our work fosters development and demonstration of new operator interface technologies to improve the flexibility and reliability of ground-to-orbit telerobotic operations. This new technology is important, given the continuing imperatives to off-load platform maintenance from the extravehicular activity/intravehicular activity (EVA/IVA) crew to on-board robot assists under direct ground mission control [1]. Permanent human capability and productive science on platforms such as the Space Station will otherwise be delayed.

COMMON TRADITIONS, COMPLEMENTARY STRENGTHS AT ETL AND JPL

JPL and ETL share a long-standing interest in human-computer cooperative control of robots, and its applications in casually structured tasks such as space assembly and servicing, hazardous materials handling, and telescience. Use of such supervised autonomy [2], versus total robotic automation, is necessitated because computer control of robots is not yet adequate to make complete task plans, learn tasks at the cognitive and motor skill of humans, or execute tasks with the dexterity of human servo-motor performance. At the other extreme, purely manual control of robots by teleoperation is often time-consuming and fatiguing, poorly suited to repeated actions of high precision, and impractical in scenarios where the operator's sensory feedback is significantly time-delayed. As regards the technologies they bring to the NASA-MITI collaboration, JPL and ETL have chosen complementary approaches to developing supervisory automation. JPL's approach, consistent with its space operations charter, derives from computer-augmented teleoperation [3, 4], the goal to date having been to maximize manual tasking dexterity and telepresence, and extend both to multiple-second time-delay remote-servicing scenarios. For example, JPL, utilizing its development of a dual eight-degree-of-freedom force reflecting teleoperator with multi-mode controls (position, force, rate, compliance, shared computer control of some axes) has recently re-enacted various aspects of the Solar Maximum Satellite repair mission conducted on shuttle flight STS-13; in this earth laboratory simulation, JPL telerobotically performed key sequences of this benchmark 1984 dual-EVA shuttle-bay servicing mission. Two JPL enabling technology developments have been *calibrated preview/predictive graphics displays* [4] and *shared compliance control* [3]. Using such a preview/predictive graphics operator interface and a related robot compliance control, JPL and NASA Goddard Space Flight Center (GSFC) recently performed simulated ground-to-orbit space telerobotic servicing under multiple-second variable communication time delay, wherein JPL successfully changed out an Orbital Replaceable Unit (ORU) of a Hubble Space Telescope-like spacecraft mock-up located some 4000 kilometers distant at GSFC [4].

By comparison to the above JPL work, ETL has recently emphasized higher level intelligent and cooperative control interactions between humans and robots [5, 6]. Consistent with a strong interest in flexible assembly operations, ETL seeks to relieve operator fatigue through automation, yet allow the operator to manually interact with robot automation with ease if needed. For example, ETL has demonstrated *intelligent and cooperative control* in robotic chemical assay by flame test. The robot, under supervised autonomy, sets up, pulverizes, samples and flame-tests chemical substances, with the operator intervening to graphically re-designate locations of desired actions or teleoperate to deal with task anomalies. ETL has developed the MEISTER (Model Enhanced Intelligent and Skillful Teleoperational Robot) system architecture to enable such supervisory control [5]. A key design feature of this architecture is the embedding of environmental and control knowledge within a collection of task-oriented object models, wherein the model representation itself is "object-oriented," e.g., each object model contains self-knowledge such as position and orientation with respect to world coordinates ("object localization") and its affixment relationships to other objects. The object models embed both generic and specific handling knowledge, such that the commanding of a control operation, e.g., **pick_and_ place**, invokes a linked hierarchy of processes, including the automatic sequencing of basic camera-viewing primitives [6].

NEAR-TERM PLANS AND PROGRESS

JPL and ETL separately fund their U.S.-Japan telerobotics R&D cooperation through projects respectively entitled "Distributed Space Telerobotics," and "Interoperation Technology for Long-Distance Robotics." These efforts, which independently develop their component technologies, converge in jointly implemented overseas system demonstrations. The first planned experiment (US-FY95) is *truss-based telerobotic deployment of a solar-powered Orbital Replaceable Unit (ORU) and electrical connectors.* This operation will be performed from JPL by a joint ETL-JPL team controlling an ETL robot. There will be a reciprocal operations experiment (US-FY96) from ETL to JPL where a joint JPL-ETL team will perform *telerobotic servicing of a limited-access ORU in a simulated Space Station environment.* In general, these experimental demonstrations and underlying technology developments highlight *robust telerobotic operation under uncertainty.* Major sources of operational uncertainty include effects of time delay, limited camera viewing, and lack of prior task knowledge. We are addressing two corresponding key technology needs [1]. The first technology need is to develop *an intelligent interface for operator visualization of complex workspaces*, as motivated by the requirements to safely perform robotic servicing tasks in physically obstructed, limited viewing access structures, and also to maximize viewing automation under well-structured operating conditions. Desired capabilities include a computer planned-and-task-synchronized presentation of the global workspace that fuses remote multi-camera video with 3-D graphics, and correlates this display with operator information requirements for specific task processes and interventions. Measurable outcomes will include: a) reduction of the operations time used for manual camera control during a task, which often outweighs manipulation time, and usually requires an additional operator, b) the capability, through a coherently integrated presentation of real and synthesized task views, to safely operate in scenarios where camera viewing alone is inadequate. JPL refers to this work as ***Intelligent Viewing Control (IVC)***, which is well-motivated by the limited camera resources and on-orbit time available for their use in future Space Shuttle/Space Station external robotic operations [1]. Other important IVC applications are areal surveillance, medical viewing, and flexible automation workcells. ETL and JPL both conduct related R&D, with ETL emphasizing object-based models for camera view planning. JPL, emphasizing the real-time integration of 3-D graphics and an AI-based view controller, carried out this year proof-of-concept robotics experiments with a first-cut IVC subsystem implementation.

The second, complementary technology need is to develop a *ground control interface for dexterous robotic tasking under extended (2-10 s) and intermittent time delays*, as motivated by the requirement to safely telemanipulate in casually structured, and a priori less-well-modeled scenarios. The problems of teleoperation at time delays exceeding one second are well known [2], and the most recent predictive graphics-based approaches [4], per above, have as yet advanced reliable operations to one-to-four seconds' delay for a priori well-modeled tasks. The desired new capabilities are to elevate the predictive graphics-and-compliance control paradigm [3] to a more flexible "teleprogrammed" form of supervisory control. In this new approach, the operator still manually inputs motions to a modeled task environment.

However, rather than these continuous operator motions being sent directly to the robot, they are first parsed by computer to discrete low-level autonomous commands, which are then communicated to the remote site asynchronously. Once received, the commands are interpreted by the robot controller as simple guarded motion control primitives referenced to real-time robot sensor data. This approach enables introduction of intelligent, corrective robot behaviors to compensate for problems that the time-delayed operator cannot immediately address -- we note some preliminary progress below. Measurable outcomes of this work will include: a) successful demonstrations of the teleprogrammed mode at time delays up to 10 seconds for representative **align, cut, grasp, insert and detach** operations, b) application to situations where prior object model knowledge is of low quality (re: shape, position, orientation), requiring either significant qualitative control adaptation by the robot and/or on-line task model refinement by operator-interactive 3-D graphics acquisition-and-calibration. JPL refers to this work as ***Intelligent Motion Control (IMC)***, which is well-motivated by needs for more flexibly structured ground control of spacecraft EVA/robotic maintenance and telescience handling on the Space Station [1].

ETL and JPL initiated experimental interactions and reciprocal engineering visits between our robotics laboratories in fall 1993. To date, ETL-JPL have performed several simple experiments to verify basic inter-lab operability. Also, JPL, working with the University of Pennsylvania, implemented and demonstrated important elements of an IMC subsystem.

ETL->JPL Remote Operations. JPL and ETL engineers, working together at ETL's Intelligent Interface Systems Lab, remotely commanded the guarded motion trajectory of an 8-degree-of-freedom JPL arm about the perimeter of a satellite ORU access panel door, simulating a proximity operations inspection (eye-in-hand camera).

JPL->ETL Remote Operations. JPL engineers commanded a robot at ETL in simple pick-and-place operations, via a high-level control interface. JPL sent the ETL robot control commands via a socket connection over Internet, from a LISP control program at JPL, to another LISP robot control program at ETL. Workspace models that enabled successful execution were resident within the robot control program at ETL.

Intelligent Motion Control. JPL, working with University of Pennsylvania researchers [7] installed at JPL a real-time robot controller and command interfaces that compose the robot site of a "teleprogramming" facility, wherein the operator will command a time-and-space distant robot over communication links that may have variable delay. The fundamental University of Pennsylvania contribution in this work includes development at JPL of a novel layered behavioral control architecture. When active, this behavioral control replaces a more conventional hybrid position/force control, as conventionally used to correct quantitative variations in robot force and position along various axes of robot tool or gripper contact with an object of interest [3], and can autonomously compensate and strategically correct for undesirable qualitative changes in the task state, as determined by the robot sensors. For example, the controller can assist a time-delayed operator in dealing with sudden, unpredictable disturbances and variations in contact with a workpiece being serviced, or object encountered. JPL-UPenn successfully demonstrated in January 1994 use of the behavioral controller to puncture and slice a Kapton tape seam securing satellite thermal blankets about a replica ORU main electronic box (MEB) access panel door. The controller successfully managed multiple, unpredictable metal-to-metal sidewall contacts as a cutting tool traveled laterally in a 2-mm-wide groove of a continuous 40-cm path sweep. Such tasks have challenged the skills of even experienced human operators in teleoperations tests.

ACKNOWLEDGMENTS

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. JPL participants include Won Soo Kim, Eric Paljug, and Stephen Peters; the ETL team includes Kazuo Machida and Yujin Wakita, and a collaboration with Kawasaki Heavy Industries, Inc. The authors appreciate the assistance of Charles Weisbin (JPL) and Kunikatsu Takase (ETL) in developing the NASA-MITI cooperation.

REFERENCES

1. *Space Station/OACT Robotics Technology Study*, Vol. 1., NASA TM B93DSS-039, Oceaneering Space Systems with McDonnell Douglas Aerospace, 1993.
2. T. Sheridan, *Telerobotics, Automation, and Human Supervisory Control.* Cambridge, MA: The MIT Press, 1992.
3. P. S. Schenker, A. K. Bejczy, W. S. Kim, and S. Lee, "Advanced man-machine interfaces and control architecture for dexterous teleoperations," *Proc. Oceans `91*, pp. 1500-1525, Honolulu, HI, October, 1991.
4. P. S. Schenker and W. S. Kim, "Remote robotic operations and graphics-based operator interfaces," in *Proc. 5th Intl. Symp. on Robotics and Manufacturing* (ISRAM `94), Maui, HI, August 14-18, 1994, and references therein.
5. S. Hirai, T. Sato and T. Matsui, "Intelligent and cooperative control of telerobot tasks," *IEEE Control Systems Magazine*, pp. 51-56, February, 1992, and references therein.
6. Y. Wakita, S. Hirai and T. Kino, "Automatic camera-work control for intelligent monitoring of telerobotic tasks," *Proc. 1992 IEEE/RSJ Intl. Conf. IROS*, Raleigh-Durham, NC.
7. M. R. Stein and R. P. Paul, "Operator interface for time delayed teleoperation, with a behavior-based controller," *Proc. 1994 IEEE Intl. Conf. Robotics and Automation*, San Diego, CA, May 1994.

ROBOTICS RESEARCH AT CANADIAN SPACE AGENCY

Raymond Hui
Directorate of Space Mechanics
Canadian Space Agency
6767 route de l'Aéroport
Saint-Hubert Québec
Canada J3Y 8Y9
Tel: (514) 926-4679 Fax: (514) 926-4695
email: huir@sp-agency.ca

KEY WORDS AND PHRASES

Canadian Space Agency, robotics research, technology spin-off.

ABSTRACT

In addition to major crown projects such as the Mobile Servicing System for Space Station, the Canadian Space Agency is also engaged in internal, industrial and academic research and development activities in robotics and other space-related areas of science and technology. These activities support current and future space projects, and lead to technology development which can be spun off to terrestrial applications, thus satisfying the Agency's objective of providing economic benefits to the public at large through its space-related work.

INTRODUCTION

The Canadian Space Agency (CSA) was formally established on in 1989 to bring to a central organization the responsbilities of coordinating and managing the Canadian Space Program. Our formal mandate is to (i) promote the peaceful use and development of space, (ii) advance the knowledge of space through science, and (iii) ensure that space science and technology provide social and economic benefits for Canadians. In the fall of 1993, most of the constituent groups of CSA (previously distributed in a number of locations in Ottawa, Ontario) and carrying out this mandate were moved to a central location in St-Hubert, Québec.

In the recently finalized Canadian Long Term Space Plan [9], the objectives and action plan for the Canadian Space Program in the next ten years are described. It is clear from this plan that in addition to major crown projects such as Radarsat and Mobile Servicing System (MSS) for Space Station, the Canadian Space Agency will continue to engage in research and development activities under the auspices of its Space Science and Space Technology Branches, as well as the Canadian Astronaut Program. Herein, the discussion is focused on R & D in robotics at CSA. This subject has been previously discussed in [3, 4]. More updated information is provided here.

ROBOTICS RESEARCH

The Strategic Technologies for Automation and Robotics (STEAR) program was established to complement the work on MSS and to promote Canadian robotics activities. Specifically, companies, universities and research organizations across Canada are given contracts to develop technologies for the MSS evolution and terrestrial spinoff applications.

To date, contracts for eight different STEAR projects have been awarded. The areas of technology being investigated are (i) automation of operations and expert systems, (ii) health and automated power management, (iii) autonomous robotics, (iv) enhanced space vision systems, (v) trajectory planning and obstacle avoidance, (vi) protection of materials in the space environment, (vii) tactile and proximity sensors, and (viii) MSS ground control.

The Canadian Astronaut Program (CAP) pursues R & D activities which will be tested and/or implemented by Canadian astronauts in future space flight missions. One such example is the Space Vision System (SVS) [11]. Current projects which are being considered for future flights include a motion isolation mount based on magnetic levitation [13] and human machine interface based on speech recognition [12].

The Space Technology Branch at CSA has a dual mandate to develop necessary space technologies to support current and future missions as well as to develop and transfer terrestrial technology spin-offs to the industry. In the area of robotics, there is ongoing R & D in the areas of teleoperation, sensor fusion, development of advanced control techniques, control of flexible manipulators, free flyers, human-machine interface and robot calibration in space. These activities include theoretical and experimental work and are described in detail below.

Successful testing of the SVS on mission STS-52 [11] showed that it was possible to calibrate robot performance in space using photogrammetry techniques. The REACH project [8] is aimed at evaluating and characterizing the Space Station Remote Manipulator System (SSRMS) in orbit. Parameters such as accuracy, repeatability, stopping distance, etc. will be measured. A ground test-bed is being constructed to verify the validity of the characterization procedure. The concept of REACH may be tested in a upcoming Shuttle flight using the Shuttle Remote Manipulator System (SRMS) with the SSRMS to be characterized after evaluating the initial flight data. Other experiments involving calibration of the SRMS and the Shuttle are being contemplated as an operational version of SVS will become available on future Shuttle missions.

Since it will be difficult and expensive to alter the hardware of MSS once it is set up in space, a natural area to incorporate new technology is remote control. To this end, CSA researchers are examining issues such as bilateral teleoperation and effects of communications delay. Novel hand controllers and haptic interface devices have been invented as a result of this work [7].

To facilitate experiments involving multiple devices such as hand controllers and robots, a general host environment dubbed *Ghost* is being developed to enable non-expert users to link up devices and processes across a network to form customized experimental systems [6]. Currently, the available devices include a 7-dof robot, a Polhemus sensor connected to the CSA network via a PC running QNX and a UNIX workstation respectively. The processes which can be executed include two types of simulated robots, a driver which allows a computer mouse to be used like a hand controller, and various display processes. Other robots including the above-mentioned force-reflecting mechanisms and a planar free flyer as well as processes such as simulation programs will be added. In the near future, it will be possible to communicate with and control devices at sites outside CSA.

Dynamics and control of flexible manipulators are a natural problem for CSA to study because Canada has contributed and is contributing large, flexible robotic manipulators to the Shuttle and Space Station programs respectively. The incorporation of intelligent control methods such as fuzzy control, neural networks and genetic algorithms [10, 16], as well as the use of smart structures is being examined. Another area of interest is force control of manipulators with flexible links and/or joints [15]. Experiments with direct-drive motors and a flexible link, as well as with flexible joints (harmonic drives) are being conducted. The dynamic coupling between the Special Purpose Dextrous Manipulator (SPDM) and the SSRMS is also a subject of interest being studied in collaboration with Laval University [14].

Space servicing is potentially a viable commercialization opportunity in space in the near future. Although it is not part of the Long Term Space Plan, it is important to understand the dynamics and control of servicers and servicing manipulators [5]. To this end, a planar test bed based on air bearings has been designed at CSA [2]. Ongoing research on the dynamics of space manipulators is leading to the development of a new philosophy on their design.

TECHNOLOGY SPIN-OFF

One mandate of CSA and the Space Technology Branch is the transfer of technology to the industrial sector. The STEAR program mentioned above is the main channel through which MSS technology can be spun off for terrestrial applications. The approach taken in developing the spin-off technology is somewhat unique. STEAR contracts have been and will be given out to companies other than the prime contractors responsible for the design, construction and delivery of MSS. The objectives of these contracts are to develop technology which may be used in evolutionary MSS in parallel with but independently of the prime contractors as well as terrestrial applications.

In addition to STEAR which is a program to direct, fund and coordinate industrial R & D, internal research at CSA, in particular the robotics has led to technology which can be spun off into commercial products for terrestrial applications. The hand controller mechanisms and *Ghost* mentioned above are examples of research results which have great commercialization potential.

CONCLUSIONS

CSA is engaged in various research and development activities in addition to major crown projects. In the area of robotics, the focus is on characterization of robots in space, remote manipulation, human-machine interface, flexible manipulators and space servicing. Research in these areas support the major projects as well as lead to technology spin-offs which provide socio-economic benefits for Canada.

REFERENCES

[1] Hayward, V. and Reynier, F., “Report on the Kinematic, Dynamic and Structural Design for the Development of a Miniature Hand-Controller with Force and Tactile Feedback Capability”, Canadian Space Agency contract report 9F009-1-1441/01-SR, 1992.

[2] Hui, R., “Preliminary Design of the DSM Free Flyer”, CSA Internal Report, 1993.

[3] Hui, R., "Robotics Research at the Canadian Space Agency", *IEEE Society of Robotics and Automation Magazine*, April, 1994.

[4] Hui, R., "In-House R & D in Robotics at the Canadian Space Agency", IARP, Montreal, Canada, 1994.

[5] Hui, R., "Analysis of Space Manipulators with Non-Specific Reference Links", to be presented at IROS '94, München, Germany, 1994.

[6] Hui, R., Kry, P. and Williams, S., "Development of Ghost - A General Host Environment for Robot Control", in preparation, 1994.

[7] Hui, R., Ouellet, A., Wang, A., Vukovich, G. and Peruzzini, W., "CSA Bilateral Mechanical Impedance Devices", in preparation, 1994.

[8] Kalaycioglu, S., Zheng, W. and Giray, M., "Robotics Evaluation & Characterization (REACH) of the SSRMS", IARP, Montreal, Canada, 1994.

[9] Knapp, B., "Canada Broadens Focus in New Space Plan", *Space News*, vol. 5, no. 23, June 13 - 19, 1994, p. 4.

[10] Lee, J. X., Vukovich, G. S, and Sasiadek, J. Z., "Fuzzy Control of a Flexible Link Manipulator", *Proceedings of the American Control Conference*, Baltimore, MD, 1994.

[11] Maclean, S. G. and Pinkney, H. F. L., "Machine Vision in Space", *Canadian Astronautics and Space Journal*, vol. 39, no. 2, 1993.

[12] Payette, J., De Mori, R. and Hayward, V., "An Experiment in Robot Operator Control with Automatic Speech Recognition", IARP, Montreal, Canada, 1994.

[13] Salcudean, S., Davis, H., Chen, C., Goertz, E. and Tryggvason, B., "Coarse-Fine Residual Gravity Cancellation System with Magnetic Levitation", *Proc. of IEEE Int. Conf. on Robotics and Automation*, Nice, France, 1992.

[14] Vaillancourt, C. and Gosselin, C., "Compensation of the Structural Flexibility of the SSRMS by the SPDM", IARP, Montreal, Canada, 1994.

[15] Vukovich, G. and Mills, J. K., "Modeling, Stability and Control Issues of Elastic Joint Manipulators During Contact with a Stiff Environment", *Proceedings of IEEE Conference on Decision and Control*, Honolulu, HI, 1990.

[16] Xu, H. Y. and Vukovich, G., "Robotic Modeling and Control Using a Fuzzy Neural Network", *Proceedings of the 2nd IEEE Int. Conf. on Fuzzy Systems/1993 Int. Conf. on Neural Networks*, San Francisco, CA., 1993.

Design and Development Status of ETS-7, an RVD and Space Robot Experiment Satellite

M. Oda, T. Inagaki, M. Nishida, K. Kibe, F. Yamagata

National Space Development Agency of Japan
E-Mail: ODA@rd.tksc.nasda.go.jp
tel: +81-3-5401-8630, fax: +81-3-5401-8672

ABSTRACT

ETS-7 (Engineering Test Satellite #7) is an experimental satellite for the in-orbit experiment of the Rendezvous Docking (RVD) and the space robot (RBT) technologies. ETS-7 is a set of two satellites, a chaser satellite and a target satellite. Both satellites will be launched together by NASDA's H-2 rocket into a low earth orbit. Development of ETS-7 started in 1990. Basic design and EM (Engineering Model) development are in progress now in 1994. The satellite will be launched in mid-1997 and the above in-orbit experiments will be conducted for 1.5 years. Design of ETS-7 RBT experiment system and development status are described in this paper.

MISSION OF ETS-7

Mission Objective of ETS-7

Space development activities of our mankind are evolving since its start. Development of geostationary communication, broadcasting and weather observing satellite is nearing its maturity. It is desired by many people to expand our manned presence in the earth orbit and its beyond such as the Lunar and the Mars. It is also desired to deploy robotic spacecraft as a precursor or as an alternative of the manned presence.

The RVD and RBT technologies are the "must technologies" for the future space activities as above whichever it is manned or unmanned. NASDA has a history of RVD and RBT technology research for more than ten years. These technologies are difficult to fully verify their capability on ground. Therefore, NASDA decided to develop and launch an engineering test satellite called ETS-7 (Engineering Test Satellite #7) to perform the in-orbit experiments of the RVD and RBT technologies. The RVD technology will be applied for the future spacecraft development such as the HOPE space plane. The RBT technology will be applied for the future robotic servicing spacecraft, lunar/planetary exploration, space station utilization and others.

RBT Experiment Plan

Detail of the ETS-7 robot experiment plan is described in reference [1] and [2]. It is summarized as follows:

- Performance evaluation of the onboard robot subsystem and its equipment such as a robot arm, tool, vision system, orbital replacement unit and others in the actual space environment.
- Experiment of the cooperative satellite attitude and robot arm control.
- Teleoperation experiment of the robot arm from a ground control station. It must be noticed that this experiment is a so-called telerobot experiment and the telemanipulation is just part of the experiments.
- Experiment of the in-orbit satellite servicing such as ORU (Orbital Replacement Unit) exchange, fuel supply (dummy fuel is used) and target satellite handling.
- Some optional advanced experiments such as target satellite capture by the robot arm.
- National Lab's robot technology experiments by MITI/ETL (Electrotechnical Lab), NAL (National Aerospace Lab) and CRL (Communication Research Lab).

RVD Experiment Plan

Detail of the RVD experiments is described in reference [3]. It is summarized as follows:

- In-robot performance evaluation of the newly developed RVD subsystem's components such as GPS receiver (GPSR), Rendezvous Radar (RVR), Proximity Sensor (PXS), Docking Mechanism (DM) and Guidance Control Computer (GCC).
- Experiment of autonomous rendezvous

navigation and control of the chaser satellite toward the target satellite.
- Experiment of autonomous docking operation between the chaser satellite and the target satellite.
- Some optional advanced experiments such as remote piloting of the chaser from a ground control station.

SYSTEM DESIGN OF ETS-7 EXPERIMENT SYSTEM

Mission Requirements

In order to perform the above RVD and RBT experiments using NASDA's limited ground and in-orbit infrastructures, the following major mission requirements are identified:

- ETS-7 should consist of two satellites, a chaser satellite and a target satellite. Both satellites should be launched together by NASDA's H-2 rocket.
- NASDA's experimental data relay satellite called COMETS (Communication Engineering Test Satellite, to be launched in 1997) will be used to communicate with ETS-7 from a ground control station which will be located at NASDA's Tsukuba Space Center.
- Since only one data relay satellite can be used, communication coverage area in the orbit is limited. Therefore ETS-7's RVD system should be an autonomous system which is managed by the onboard guidance and control computer and related RVD sensors.
- ETS-7's RBT system should be a telerobot system whose control functions are shared by an onboard controller and a ground control facility and operator.

Satellite Platform

(1) Satellite Configuration. From the above mission requirements, ETS-7 is a set of two satellites, a chaser satellite and a target satellite. Both satellites will be launched together by NASDA's H-2 rocket in 1997. Launch date (year) was decided to meet the dual launch opportunity with the TRMM (Tropical Rainfall Measurement Mission) satellite which will be developed jointly by NASDA and NASA. Capability of the H-2 rocket and mass of the TRMM satellite decide the mass of the ETS-7 chaser and the target satellite. Those are approximately 2.2 ton and 0.4 ton, respectively.

(2) Communication System. Communication between the ETS-7 satellite and the ground operation facility will be established using NASDA's experimental data relay satellite called COMETS and NASDA's tracking control network. The chaser satellite has a dish-type antenna for the intersatellite communication. COMETS will be located at 121 degrees east longitude on the geostationary orbit. Communication with the target satellite will be established through the chaser satellite. Since excess communication capability requirements will make it difficult to plan and design future space missions which use RVD and RBT technologies, requirements for the communication system were set to be minimum. Allocations of communication capacity for RVD or RBT experiments (except satellite platform operation) are as follows:

- Command: 1.3 kbps
- Telemetry: 16 kbps
- Video data: 1.2 Mbps
- Round trip time: approximately 4 to 5 s

(3) Attitude Control. Since much electrical power is necessary for the RVD and RBT experiments, the chaser and the target satellite are three-axis-stabilized satellite with deployed solar panels. During the RVD experiments, attitude and position of the chaser satellite are controlled by the RVD system. During the RBT experiments, satellite attitude is maintained by the attitude control system. Attitude control performance requirements during the RBT experiments come mainly from the intersatellite communication. Since robot arm motion will affect the satellite attitude stability, the cooperative control between the satellite attitude controller and the robot arm controller is necessary and will be tested as one of the RBT experiments.

RVD System

The onboard RBD system consists of GPS receiver (GPSR), Rendezvous Radar (RVR), Proximity Sensor (PXS), Docking Mechanism (DM) and Guidance Control Computer (GCC).

GCC will select a proper sensor from the above three sensors to measure relative location of the chaser and the target satellites. Docking of both satellites is so-called "soft docking". Relative velocity of both spacecraft at the docking is about 10 mm/s. Detail of the RVD system is described in the reference [3].

RBT System

Onboard and Ground Functions. ETS-7 is a kind of a telerobot system which includes the ground control system and the onboard control system. Allocation of robot control related functions to the ground control station and the onboard system is as follows:

Functions	Ground Function	Onboard Function
Teleoperation	– Generate Teleoperation Commands from Hand Controller Input – Teleoperation support (Real-time CG simulation) – Generate Robot Language Command	– Interpolate Teleoperation Commands – Automatic Path Generation – Generate Arm Path from the Robot Language Commands – Joint Servo Control
Cooperative Control with S/C Attitude	– Arm path planning which does not bother satellite attitude motion – Satellite attitude control status check	– Feedforward Compensation of the RBT motion
Vision Data Processing	– On-ground Vision Data processing (On-line)	– Onboard Vision Data Processing (On-Line/Off-Line)
Target Satellite Handling	– Experiment Planning – Arm path generation	– Rendezvous Control – Docking Mechanism Control – Robot Arm Control

Onboard Robot Subsystem. The onboard robot subsystem is composed of the following equipment: All robot and vision subsystem equipment except RMOC, ADE and VDP are mounted on the +Z panel (Earth-pointing surface) of the ETS-7 satellite. Size of the +Z panel is 2.28 m * 1.85 m. The Earth sensor and other satellite platform equipment such as omni antenna are also mounted on this panel. ETS-7 satellite and the onboard RBT system are shown in Figure 1 and Figure 2.

Onboard Robot Components

(1) Robot Arm

ETS-7 robot arm (ERA) is a 6-degrees-of-freedom manipulator whose length is about 2 m. Each joint is driven by a combination of the DC brushless motor and the harmonic drive gear. ERA has following control modes.

- Joint Position/Rate Control mode
- Cartesian Position/Rate Control mode
- Cartesian Compliance Control mode

(2) Robot arm end effector and tools

ERA has an end effector to handle the ORU. Tools which can be attached to the ERA's end effector are used for some specific tasks. A taskboard handling tool (TBTL) is used to operate experimental elements on the taskboard. A target satellite handling tool (TSTL) is used to grasp the target satellite by its handle. The hand of the Advance Robot Hand (ARH) experiment system is removable from its miniarm and can be attached to the ERA's end effector.

(3) RMOC/ADE

RMOC (Robot Mission Onboard Controller) is a 32-bit onboard computer which manages onboard robot subsystem. RMOC can perform parallel and distributed processing by 3 set of 32-bit processors which run at 20 MHz. Commands from the ground control station come every 250 ms. Interpolation of these commands into the actual robot arm control commands is done by RMOC. Robot arm joint servo control is managed by ADE (arm drive electronics).

(4) ORU

An experimental ORU (Orbital Replacement Unit) is mounted on the robot experiment platform to be handled by the robot arm. The ORU can be grasped, removed and restored by the robot arm. The ORU houses a fuel supply experiment subsystem which is to demonstrate mate/demate of a liquid QD (Quick Disconnect) connector and fuel (dummy fuel) transfer.

(5) Taskboard
The taskboard is used to evaluate the robot arm performance such as robot arm control performance. The taskboard has many experiment elements such as a Force Torque Sensor Calibration Mechanism, a Peg-in-hole experiment mechanism, small Floating Object, a slide handle and others. A tool called taskboard handling tool (TBTL) is used to handle these experiments. TBTL is stored on the taskboard.

(6) Vision System
A pair of AHCs (Arm Hand Camera) which are mounted on the ERA's end effector are used to measure relative attitude/distance between the robot arm and the payloads. A pair of AMC (Arm Monitor Camera) which are mounted on the ERA's first joint are used to monitor the robot arm motion. Both pair of cameras can be used as a stereo camera or a single camera with a backup.

For the robot teleoperation, 2-channel video data can be sent simultaneously to the ground control station. A digital video data compression in the JPEG standard is used to reduce data size. For the onboard robot arm motion planning, video data from the robot arm wrist camera (AHC) can be provided to the onboard robot controller (RMOC) for the onboard video data processing. Video data processing of the target maker will take about 500 ms by RMOC.

(7) National Lab's equipment.
Beside the above equipment, following equipment from national laboratories (ETL, NAL, CRL) are mounted on ETS-7 to perform their robot experiments:

- ARH (Advanced Robot Hand experiment system: developed by MITI/ETL)
- AAM (Antenna Assembly experiment Mechanism: developed by CRL)
- TSE (Truss Structure handling Experiment system: developed by NAL)
  (Note)
  - MITI/ETL: Ministry of International Trade and Industry, Electro Technical laboratory
  - NAL: National Aerospace Laboratory
  - CRL: Communication Research Laboratory

Ground Segment. ETS-7 related ground operation facilities are all located at the NASDA Tsukuba Space Center. Data relay satellite's ground station will also be located at Tsukuba Space Center.

ETS-7 ground segment of the following elements:

- Data relay satellite (COMETS) ground station
- Satellite tracking and control center
- RVD experiment operation facility
- RBT experiment operation facility
- National Lab's ground operation facility

Figure 3 shows overall ETS-7 experiment system.

DEVELOPMENT STATUS AND FUTURE DEVELOPMENT SCHEDULE

Development Schedule

Development of ETS-7 started in 1990. The conceptual study was done in 1990 and 1991. The preliminary design and BBM development were done in 1992 and 1993. The basic design and EM development are in progress in 1994. Series of PDR (preliminary design review) meetings of ETS-7 Components, subsystem and satellite system are held between June 1994 and October 1994.

Budget for the flight hardware (PFM) production is approved this spring by the Diet and the launch of the satellite is planned in mid-(August/September) 1997. Mission life of the satellite is 1.5 years.

CONCLUSION

This paper summarized design and the current development status of ETS-7. Detail of the ETS-7 mission is described in reference [1] and [3]. Feasibility study result of optional experiments will be presented in other paper of this conference. Development of the ETS-7 satellite is now in the phase C and the satellite will be launched in 1997 by NASDA's H-2 rocket.

REFERENCE

[1] M. Oda et al. "ETS-7, the world first telerobotic satellite (Mission and its design concept)", Proc. of i-SAIRAS'92, Sept. 30-Oct. 2, 1992, Toulouse, France.

[2] M. Oda, "Preliminary design result of ETS-7, a RVD and space robot experiment satellite", Proc. of ICAR (Intl' conf. of Advanced Robotics), Nov. 1-2, 1993, Tokyo, Japan.

[3] S. Ichikawa, et al. "ETS-7 RVD experiment system, Proc. of EIOOTS (ESA-NPP-059), June 22-24, 1993, ESTEC, the Netherlands.

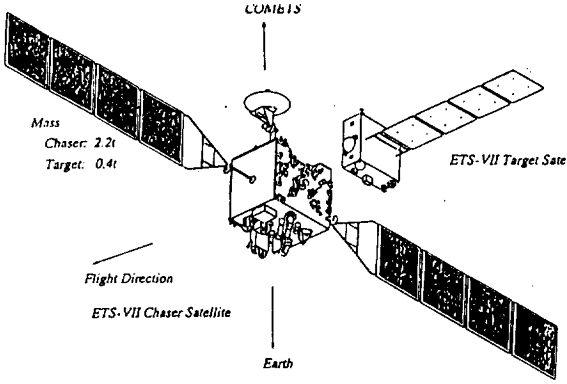


Figure-1 In-Orbit Configuration of ETS-7 Chaser and Target Satellite

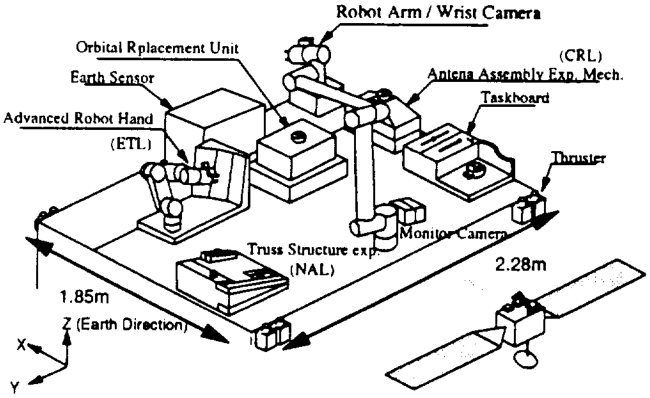


Figure-2 ETS-7 Onboard Robot System

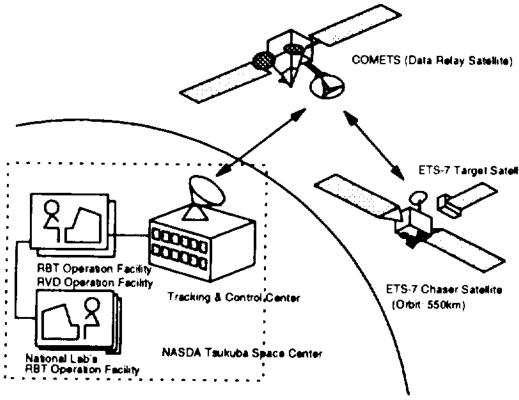


Figure-3 ETS-7 RVD & RBT Experiment System

N95- 23701

Space Robotic Experiment in JEM Flight Demonstration

Masanori Nagatomo
National Space Development Agency of Japan
2-1-1 Sengen, Tsukuba 305, Japan
Masaki Tanaka, Kazuyuki Nakamura, Shinichi Tsuda
TOSHIBA Corporation
1 Komukai, Saiwai Kawasaki 210, Japan

KEY WORDS AND PHRASES

Remote Manipulator System, Space Robot, Space Shuttle, Space Station

1. INTRODUCTION

Japan is collaborating on the multinational space station program. The JEM, Japanese Experiment Module, has both a pressurized module and an Exposed Facility(EF) as shown in Fig.1. JEM Remote Manipulator System(JEMRMS) will play a dominant role in handling/servicing payloads and the maintenance of the EF, and consists of two robotics arms, a main arm and a small fine arm.

JEM Flight Demonstration(JFD) is a space robotics experiment using the prototype small fine arm to demonstrate its capability, prior to the Space Station operation. The small fine arm will be installed in the Space Shuttle cargo bay and operated by a crew from a dedicated workstation in the Aft Flight Deck of the orbiter.

2. PROGRAM OVERVIEW

The major program milestones and activities are shown in Fig.2, in which the launch is scheduled in 1997. The preliminary design review was completed in Dec. 1992, and the detailed design has been conducted. In parallel with those design efforts, the phase 0 and the phase 1 safety reviews were also conducted as a payload of the Space Shuttle. Especially, the safety is a major design driver in this manned mission flight, and safety features have been incorporated according to the Shuttle safety requirements. As mentioned later, EVA compatibility will be tested using a weightless environment facility during the detailed design phase.

3. SYSTEM DESIGN FEATURES

As stated above, the JFD system basically consists of two elements, the cargo bay element and the AFD element. Fig.3 illustrates the cargo bay element, in which the small fine arm is installed with the support structure on the MPESS(multi-purpose experiment support structure) and also the robotics task components, an ORU(Orbital Replacement Unit) and a Task Panel, are also mounted. Two vision equipments, two sets of TVC and light, are provided to give the visual information on robotics tasks.

The small fine arm is deployed on-orbit using Arm Hold & Release Mechanism. The small fine arm has six joints to achieve six degrees of freedom movement and also has a tool, three fingers type of end effector, to capture and release the payload. A torque driver is incorporated into the tool to fasten and unfasten bolts installed in the ORU and the Task Panel. Once tool fingers are positioned and then engaged in the tool fixture, bolts which structurally attach ORU and Task Panel to the structure will be loosened by the torque driver. Then, the payload grasped by tool fingers will be manipulated by the small fine arm. The above robotics operation will be done by a crew from the AFD workstation shown in Fig.4. Two CCTV monitors, equipped for the Space Shuttle operation, will be used to show the video information at the work site. A dedicated workstation will be assembled on-orbit in the optimal location, relatively to the CCTV monitors from human-machine interface point of view. Translational and rotational handcontrollers are installed at both sides of the workstation for manual operation of the small fine arm with velocity command. PGSC(NASA-provided payload general support computer) will be equipped to display telemetry data including force and torque sensor data applied to the small fine arm. Preprogrammed control will be also available to deploy and to restow the small fine arm. From the safety point of view, appropriate number of inhibits and failure tolerance are provided to prevent an inadvertent release of payload and mechanism, according to the criticality of potential hazard.

Another feature of the JFD is EVA compatibility. As usual, for a deployable type of payload the capability to be jettisoned is required for orbiter safing, however, the contingency and unscheduled EVA design is also accommodated in the JFD system to minimize the generation of orbital debris. Those mechanisms for small fine arm joint, arm hold & release mechanism and ORU are EVA compatible to secure the safe return configuration.

4. MISSION OPERATION

The JFD will conduct end-to-end verification involving flight crew and has the following objectives:

a.Evaluate the small fine arm control performance with the actual behavior in space environment,

b.Evaluate the crew operational interface in micro gravity

The JFD mission operations are grouped into performance evaluation tasks and demonstration tasks. The performance evaluation tasks evaluate the JEM small fine arm control performance and operability, and they will provide the basis for the JEM operability evaluation. The demonstration tasks demonstrate the on-orbit maintenance functions and the payload operational support functions. The demonstration of replacing ORUs and the dexterous tasks using the small fine arm tool, its end effector, will evaluate the end-to-end JEM operability.

The performance evaluation tasks will evaluate the small fine arm control performance and human-machine interface through the actual operations. The followings will be evaluated:

a. Arm control performance
b. Single-joint drive control performance
c. Active compliance control performance
d. Human-machine interface.

The demonstration tasks are defined as follows:

a. Orbital Replacement Unit(ORU) replacement task
b. Hinged door opening/closing task.

A sequence of the performance evaluation and the demonstration tasks will be performed during a 16-hours mission timeline, i.e., two mission days. After the orbit injection, the JFD thermal control activation by a crew member follows the PLB door open. The workstation for the JFD operation is assembled at the Orbiter payload station by Intravehicular Activity(IVA) and then, the software to monitor and control the arm will be initialized. After the system checkout, the arm hold & release mechanism is activated to release the arm and then, the arm will be deployed. The basic system familialization task will be performed first. The crew member will operate the arm in the manual control mode and evaluate the human-machine interface. Then, the arm will be operated in all the control modes with and without active compliance control for unloaded conditions.

The crew will perform the ORU replacement task varying the control parameters to evaluate the operability and control performance. Also the task to open and close the hinged door in the task panel will be performed as a constrained motion of the arm.

After all the demonstration tasks are completed, the arm will be folded and the arm hold & release mechanism is activated to hold it. The equipments in the AFD and PLB are deactivated and the system will be shut down. The workstation is disassembled from the payload station and PGSC will be stowed in a MDK locker.

The crew deactivates the JFD thermal control after the PLB door closure.

The video and test data recorded and crew subjective comments transcribed during the mission are provided for the engineering evaluation of the small fine arm control performance and the crew interface.

5. CONCLUDING REMARKS

In this paper, the JFD ,an on-orbit experiment for space robotics, was described. The basic performance will be evaluated and some of the tasks in the future space operation will be demonstrated. Through the experiment, end-to-end space robot operational verification will be available and those results and experience will be reflected to the JEM development and operations and future applications.

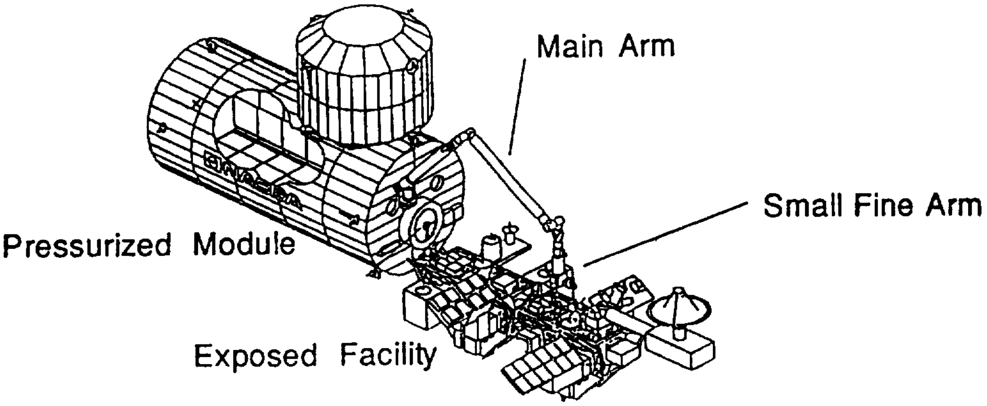


Fig. 1 JEM Configuration

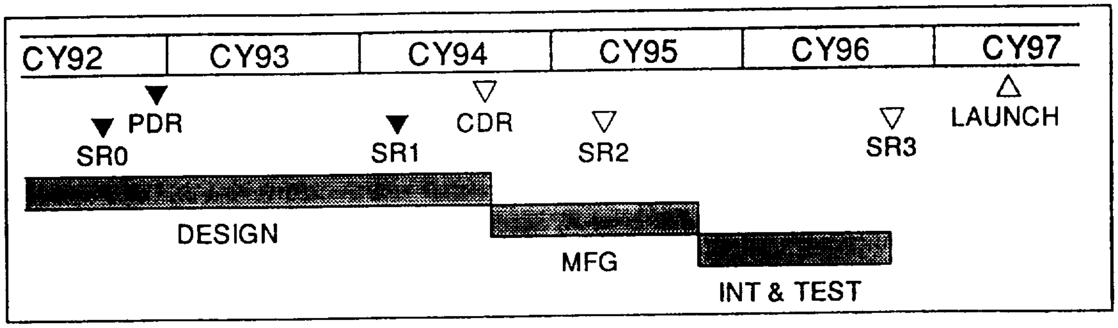


Fig. 2 Program Schedule

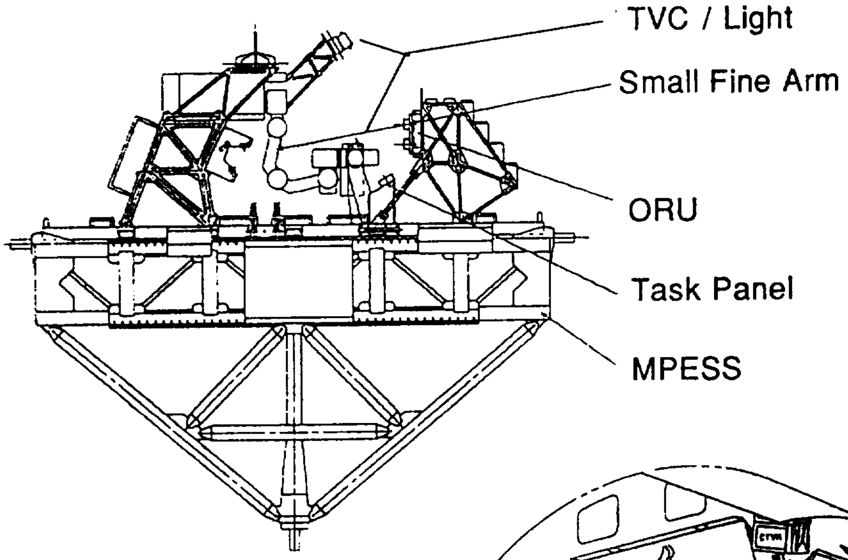


Fig. 3 Cargo Bay Element

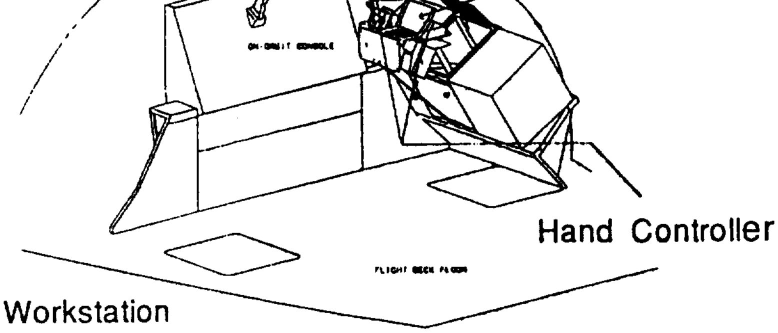


Fig. 4 AFD Workstation

Concept Verification of Three Dimensional Free Motion Simulator for Space Robot

N95-23702

Osamu Okamoto†, **Teruomi Nakaya**†, **Brett Pokines**††
†National Aerospace Laboratory, Japan
7-44-1 Jindaijihigashi-machi,
Chofu-city, Tokyo
182 Japan
Tel:+81-422-47-5911, Fax:+81-422-49-8813
E-Mail:okamoto@nal.go.jp
††Engineering of Science and Mechanics,
Virginia Polytechnic Institute and State University
Blacksburg, VA
24061-0219 U.S.A.
Tel:+1-703-231-9366, Fax:+1-703-231-4574
E-Mail;pokines@rio.esm.vt.edu

KEY WORDS AND PHRASES

Motion simulator, space robot.

INTRODUCTION

In the development of automatic assembling technologies for space structures, it is an indispensable matter to investigate and simulate the movements of robot satellites concerned with mission operation. The movement investigation and simulation on the ground will be effectively realized by a free motion simulator. Various types of ground systems for simulating free motion have been proposed and utilized. Some of these methods are a neutral buoyancy system, an air or magnetic suspension system, a passive suspension balance system, and a free flying aircraft or drop tower system. In addition systems can be simulated by computers using an analytical model. Each free motion simulation method has limitations and well known problems, specifically, disturbance by water viscosity, limited number of degrees-of-freedom, complex dynamics induced by the attachment of the simulation system, short experiment time, and the lack of high speed super-computer simulation systems, respectively.

The basic idea presented here is to realize 3-dimensional free motion. This is achieved by combining a spherical air bearing, a cylindrical air bearing, and a flat air bearing. A conventional air bearing system has difficulty realizing free vertical motion suspension. The idea of free vertical suspension is that a cylindrical air bearing and counter balance weight realize vertical free motion. This paper presents a design concept, configuration, and basic performance characteristics of a innovative free motion simulator. A prototype simulator verifies the feasibility of 3-dimensional free motion simulation.

DESIGN CONCEPT

The suspension system of the simulator developed consists of three air bearings. A spherical air bearing is located at the top of the suspension system, and a flat air bearing at the bottom of it. A cylindrical air bearing is placed between the spherical air bearing and the flat air bearing.

The use of a high pressure air feed mechanism to each air bearing is a key aspect in achieving free motion simulation. The flat air bearing at the bottom and flat bearing in the middle are fed high pressure air through a rigid air pipe from a tank. A rigid air pipe can be used here because there is no relative displacement between the flat and cylindrical bearings. The spherical bearing at the top can not be fed high pressure air by a rigid or flexible air pipe from the same tank without restricting motion.

One solution to this problem might be to mount another tank at the top of suspension system. The spherical air bearing would be fed high pressure air through a rigid air pipe from another tank. Therefore, the pipe would not disturb the free vertical motion, because there is no relative displacement. Unfortunately, this mechanism make the system too complex causing troublesome operation.

An alternative solution to feeding air to the spherical air bearing without restricting motion is to use the cylindrical air bearing as a expandable air duct from the air tank to the spherical air bearing. The expandable air duct is a part of the cylindrical bearing and the duct is supported and sealed by small air bearings inside of the cylindrical bearing. This is the method implemented. Figure 1 shows a sectional view of the air duct mechanism.

CONFIGURATION

The simulator developed consists of a flat air bearing base plate which supports the cylindrical and spherical air bearings and an air-tank. The flat air bearing rides on top of a smooth granite table. The configuration of the free motion simulator is shown in Figure 2.

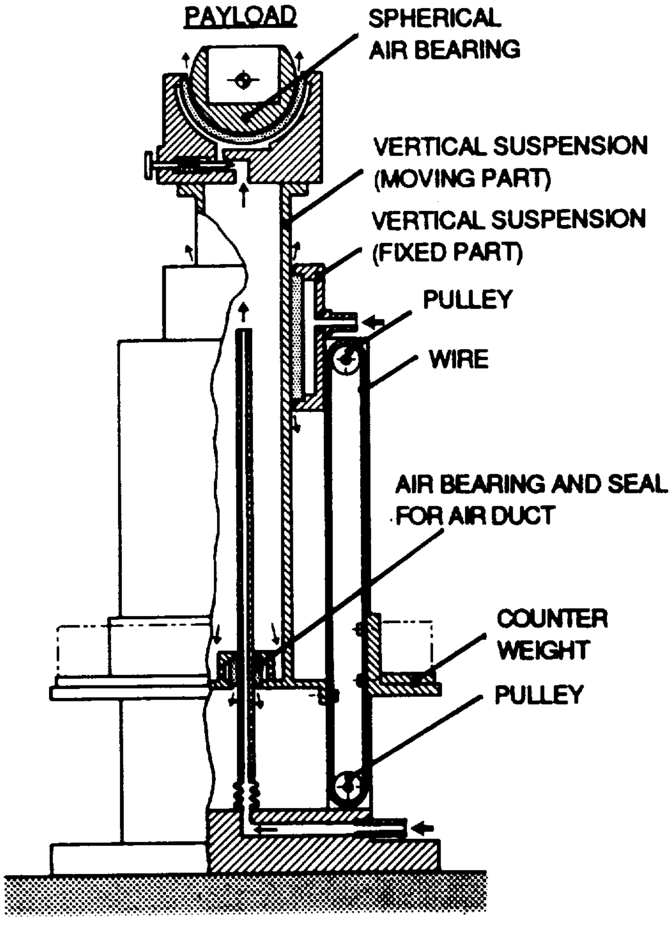


Figure 1. Illustration of sectional view of air duct mechanism.

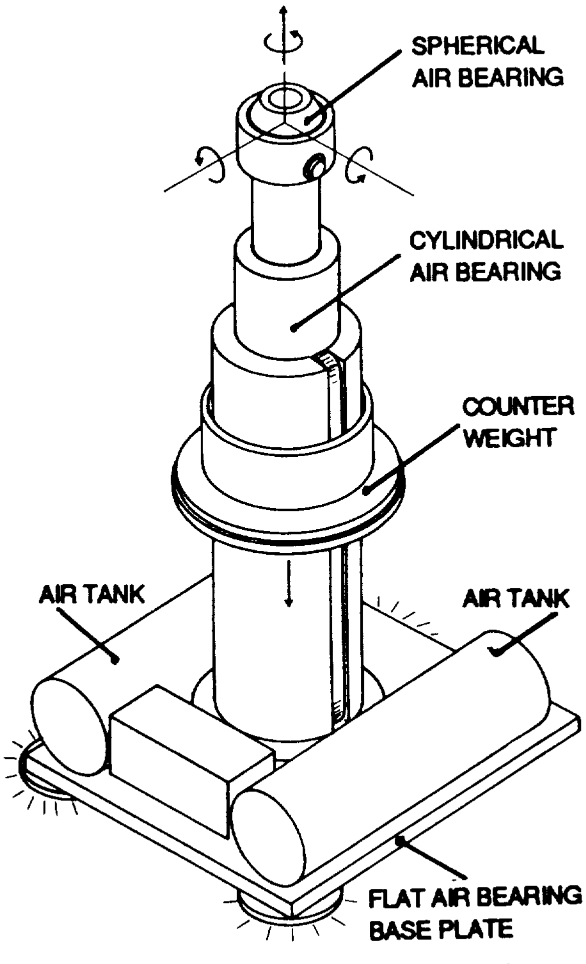


Figure 2. Configuration of free motion simulator.

The flat air bearing is made of porous sintered metal and gives smooth and stable 2-dimensional motion. The cylindrical and spherical air bearings are made of a porous graphite material and allow free vertical and rotational motion. The porous graphite material used in the air bearings prevents the seizing of the bearing.

Vertical mass is balanced by a counter weight which is suspended by a thin wire and pulley.

The payload is mounted at the top of the spherical air bearing. The center of mass of the payload is coincident with the center of the spherical bearing.

SPECIFICATION

Total mass of the free motion simulator is 80 kilograms and payload capacity is 20 kilograms. The maximum stroke of the vertical axis is 0.2 meters and +/- 45 degrees of rotational motion. Air pressure is 4 kilogram forces per square centimeter. The size of the base plate is 0.5 meter by 0.5 meters. The height of the simulator is 1.2 meters without a payload. The mass of the counter weight is 23 kilograms plus a payload weight. The capacity of the air tank is 8 litters and operational free motion time is 1 minutes. The friction of vertical suspension is less than 0.1 newtons.

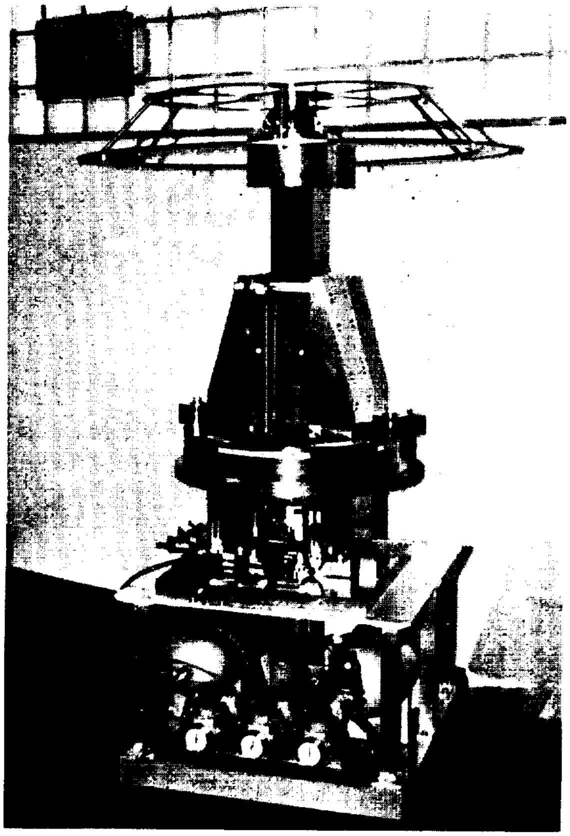


Figure 3. Photo of prototype model of simulator.

Figure 3 shows an overview of the free motion simulator and Figure 4 shows a photo of cylindrical air bearings.

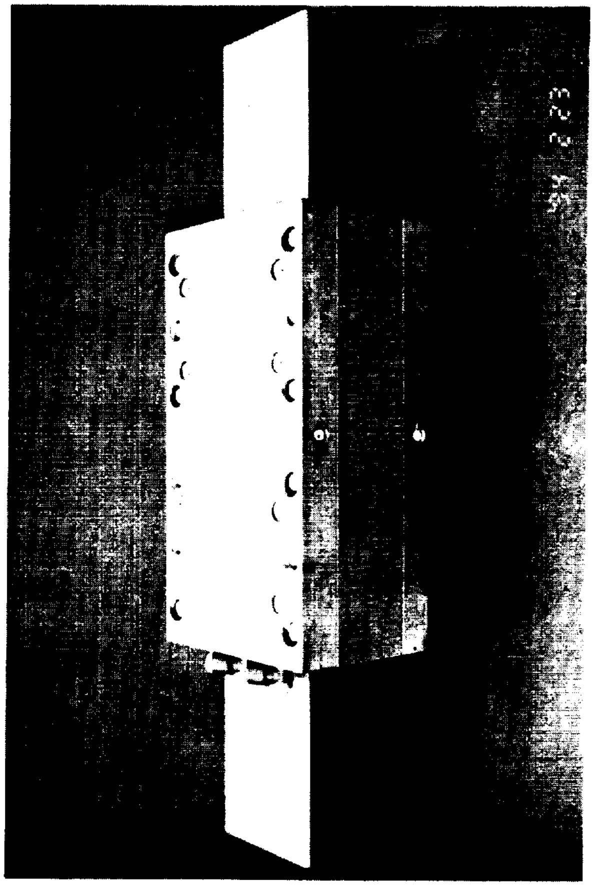


Figure 4. Photo of cylindrical air bearing.

CONCLUSION

A concept of ground free motion simulation and a prototype model for verification of concept feasibility was presented. Some future applications are illustrated in Figures 5, 6 and 7. Figure 5 illustrates the concept of a 3-dimensional manipulator test system. Figure 6 details a method of simulating docking or berthing systems. Figure 7 illustrates the application of the simulator for astronaut extra vehicular activity training.

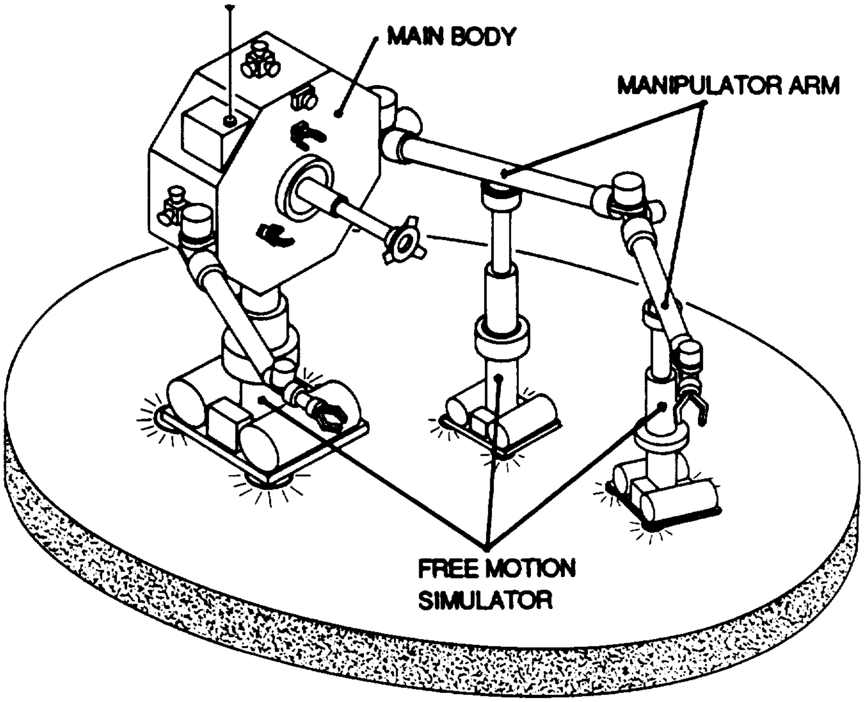


Figure 5. Concept of 3-dimensional manipulator test system.

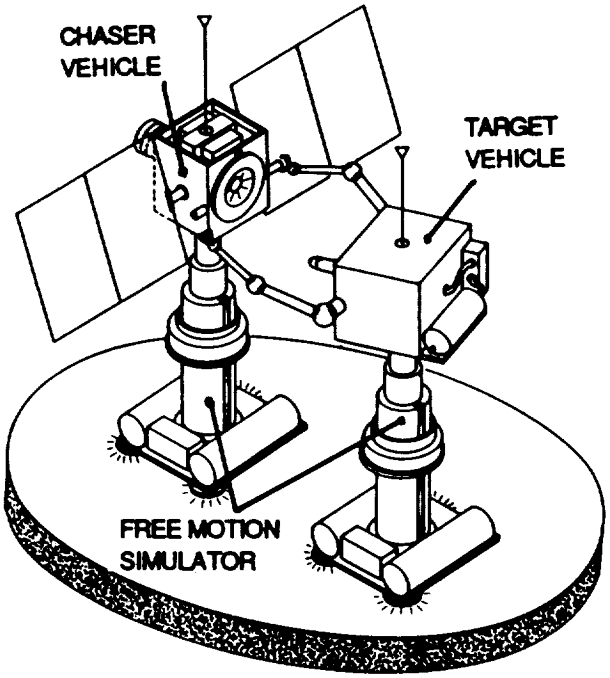


Figure 6. Method of simulating docking or berthing systems.

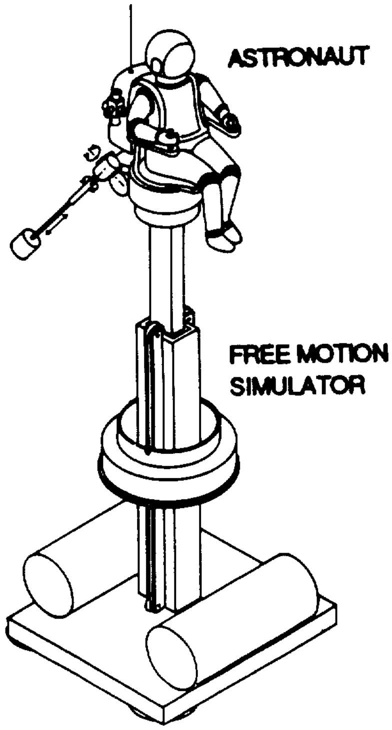
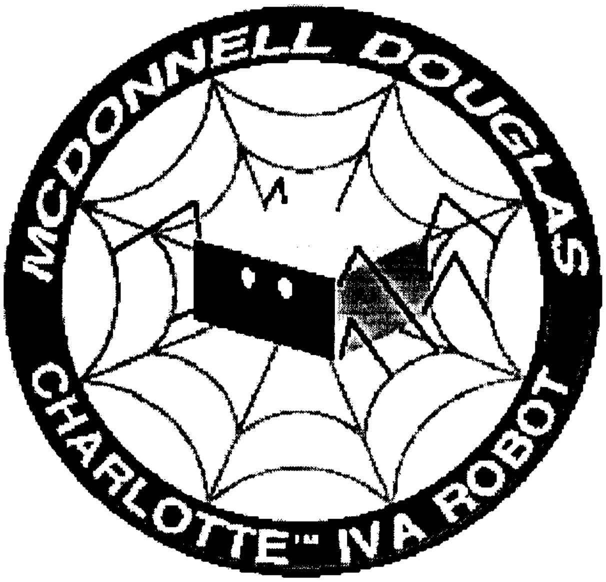


Figure 7. Application of simulator for astronaut extra vehicular activity training.

N95- 23703

The *Charlotte* ™ Intra-Vehicular Robot

Patrick L. Swaim, Clark J. Thompson, Perry D. Campbell
McDonnell Douglas Aerospace
13100 Space Center Blvd.
Houston, TX 77059
Tel (713) 283-1101   Fax: (713) 280-1925
E-mail: swaim@pat.mdc.com, thompson#d#clark@ssdgwy.mdc.com



KEY WORDS AND PHRASES

Space robotics, intra-vehicular robots, parallel manipulators

INTRODUCTION

NASA has identified telerobotics and telescience as essential technologies to reduce the crew extra-vehicular activity (EVA) and intra-vehicular activity (IVA) workloads. Under this project, we are developing and flight testing a novel IVA robot to relieve the crew of tedious and routine tasks. Through ground telerobotic control of this robot, we will enable ground researchers to routinely interact with experiments in space.

PROJECT NEED

Past crew workload projections for the Space Station Freedom have exceeded available crew time by as much as 200%. Although significant effort has been expended in the transition to the International Space Station design, few readily identifiable modifications directly improve the availability of crew time for intra-vehicular activities. Flight experience from Shuttle, Spacelab and SpaceHab missions provide corroborating evidence of the need to off load crew time. Nominal crew timelines are often exceeded, particularly when contingency operations are required. And failures of experiment apparatus between scheduled crew status checks may compromise science results. Thus, we need to not only reduce the existing crew workload, but should provide for increased monitoring of experiments. Finally, much of the activity associated with such monitoring is routine and tedious, and represents a ready target for automation.

Providing this capability by individual experiment automation will add to cost, complexity, and weight without providing a robust capability for interaction. This is particularly true in applications requiring mobility and dextrous operations. Further, many experiments and systems have already been designed for crew operation. Modifications to existing crew interfaces to make them "robot friendly" would be cost-prohibitive.

Crew familiarization and training to operate onboard experiments adds to the cost of conducting space experimentation. Principal investigators must summarize potentially years of research and specialized knowledge to deal with routine and contingency operations. And during real-time support, crew observations must be relayed to the ground, interpreted, and response measures defined, transmitted, verified, and initiated. Clearly, the efficiency of operations could be improved and more ambitious experiments conducted if ground researchers had the opportunity to directly interact with their experiments.

Conduct of experiments in the crew volume requires adherence to stringent safety procedures. The cost and effort to comply with safety standards and develop supporting documentation will often exceed the cost to develop the experiment itself. The ability to remotely conduct science in a separate enclosed volume from the crew (possibly an inert or vacuum environment) could substantially reduce these costs. Further, the ability to place a module such as the SpaceHab at locations in the Shuttle payload bay other than in the front, as required by the existing connection to the crew cabin, would enhance its manifesting options, which are constrained by the combined vehicle center of gravity location for entry and landing.

OBJECTIVE

We have presented a clear need for a system that can utilize existing crew interfaces, allow preprogrammed or teleoperation and monitoring, enable telescience, and have the potential to operate in a volume detached from the crew. Our overall objective in this project is to develop a flight-rated and tested IVA robot to meet these needs at the earliest possible date. Our system will be easily adapted to the Space Shuttle, SpaceHab, SpaceLab, MIR, and the International Space Station environments. Our specific objective for 1994 has been to complete development and certification of a flight unit for demonstration on the SpaceHab 3 mission in February of 1995.

APPROACH

Our approach is to develop an IVA robot system incrementally by employing a series of flight tests with increasing complexity. This approach has the advantages of providing an early IVA capability that can assist the crew, demonstrate capabilities that ground researchers can be confident of in planning for future experiments, and allow incremental refinement of system capabilities and insertion of new technology. In parallel with this approach to flight testing, we seek to establish ground test beds, in which the requirements of payload experimenters can be further investigated.

To these ends, we have developed an affiliation with SpaceHab Incorporated, which will allow us to gain IVA robotic flight experience. A series of flight tests, beginning with the SpaceHab 3 mission, will lead to an operational subsystem, whose services can be employed by SpaceHab experiments. We are also developing a partnership with NASA to use this platform as a test-bed to develop and integrate new IVA robotics technologies into the system. Current plans seek to provide an early demonstration of ground remote operations, followed by the integration of more dextrous end-effectors, ground telepresence control modes, and active proximity and force sensing capabilities.

In 1993 we reviewed manifested SpaceHab experiments and defined IVA robot requirements to assist in their operation. We also examined previous IVA robot designs and assessed them against flight requirements. We rejected previous design concepts on the basis of threat to crew safety, operability, and maintainability. Based on this insight, we developed an entirely new concept for IVA robotics, the *CHARLOTTE*™ robot system. Ground based testing of a prototype version of the system has already proven its ability to perform most common tasks demanded of the crew, including operation of switches, buttons, knobs, dials, and performing video surveys of experiments and switch panels.

SYSTEM DESCRIPTION

The Charlotte robot system is shown in Figure 1. Its design was driven primarily by the requirements for a compact and lightweight system which could safely operate in conjunction with crew members in a large workspace volume.

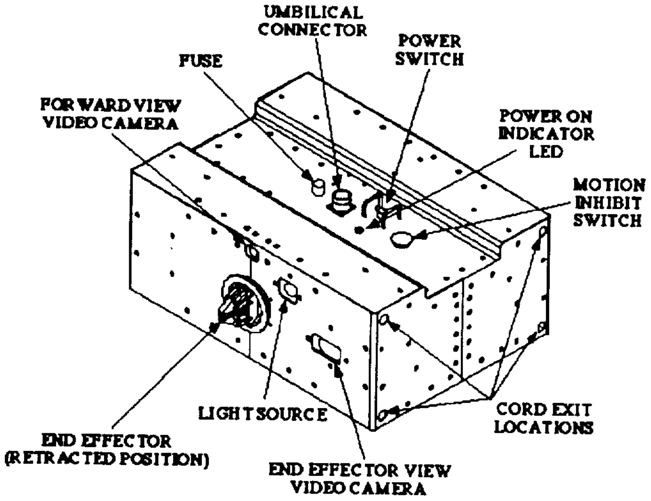


Figure 1. - The Charlotte robot.

Functionally, the system consists of a six-degree-of-freedom (DOF) motion platform with an attached 3-DOF end-effector. Eight servo controlled cables emanate from the corners of the frame to support the robot. Coordinated control of the cables allows the robot to translate and rotate within a workspace defined by the cable anchor points. The end effector is attached to the front of the frame, and consists of an extendible gripper with an infinite roll capability. The take-up spool mechanisms, drive components, control and operational computing capability are all contained within the robot's frame. A video subsystem with two CCTV cameras is also integrated inside the frame to provide views of the workspace and end-effector. The flight unit weighs less than 40 lb and measures approximately 8 x 19 X 14 inches. Power and data lines are the only external connections.

Designed as a parallel redundant cable driven manipulator, the Charlotte robot offers a number of unique features. Foremost among these for space applications is safety. Because motion in any direction requires coordinated control of all servo motors, the system has a high immunity to joint runaway. Because the manipulator is redundant, it is also highly reliable. In the unlikely event of a cable break or jam, the system will still retain full 6-DOF control, although the effective workspace volume and stiffness may be affected.

While not readily apparent, another striking characteristic of the system is its high rigidity and repeatability. The use of high modulus cables ensures a high, albeit varying stiffness throughout the workspace volume. At the center of its workspace, the Charlotte robot exhibits a stiffness greater than 1000 lb /inch. In general, as the robot moves toward the edges of the workspace, the stiffness in that direction and the normal direction increases, while the stiffness in the opposite direction decreases. These characteristics can often be exploited to great opportunity in a variety of situations.

Coupling the high stiffness with a high bandwidth position-based control system using velocity and acceleration command shaping results in very precise control and high repeatability. Cable lengths are theoretically controlled to better than 1/64000 of an inch. Positioning repeatability within the workspace has been demonstrated to be better than 0.005 inch. Angular positioning repeatability is on the order of 0.04 degrees. The current system can be controlled at rates as low as 0.001 inch/second. To minimize crew hazard, the unit has been sized to keep the applied force less than 40 lb., thereby defining the acceleration limit. Command shaping also allows the system to be controlled to minimize micro-gravity disturbances. Nominal power consumption is less than 54 watts with a 180 watts peak.

Production and maintainability of the Charlotte robot are facilitated by the use of commercial off-the-shelf components that are integrated into modular, easily replaceable units. This approach enabled us to complete a working prototype system within four months of concept development. A standard industrial computer chassis with a Intel-486 based CPU card, an electronic disk drive emulator, and multiple commercial servo-amplifiers are employed to effect motion control. A video subsystem and two CCTV cameras are also integrated into the robot. The spool and drive mechanisms have been integrated into eight identical and interchangeable cable control modules to simplify production, sparing, and logistics.

FLIGHT OPERATIONS

From an operational perspective, the system is compact, lightweight, easy to transport, and quickly installed. The crew can remove the robot from its flight locker and install it in an operational configuration in less than five minutes. The unit is transported with all cables reeled in, holding the anchor pins to the cable feed grommets at the corners of the robot enclosure. Installation is accomplished by powering up the unit, pulling each cable in turn to reel them out under active control, and attaching the anchor pins to anchor points at the boundary of the workspace. Figure 2 shows the Charlotte robot in the deployed configuration in the SpaceHab module.

Once deployed, command and control of the robot is initiated through a portable personal computer which is used as a communications terminal and operator interface to the control software that resides within the on-board master computer. Crew members will initially test the robot in a teleoperated mode, using keypad mapped controls to test the robot in each translation and rotation axis, and execute relative-move and move-to commands. Visual observation of the robot, digital position information displayed at the portable computer, and CCTV images from the robot's video cameras will be used to monitor these actions. Image recognition is used for visual calibration. Next, scripted command sequences will test the

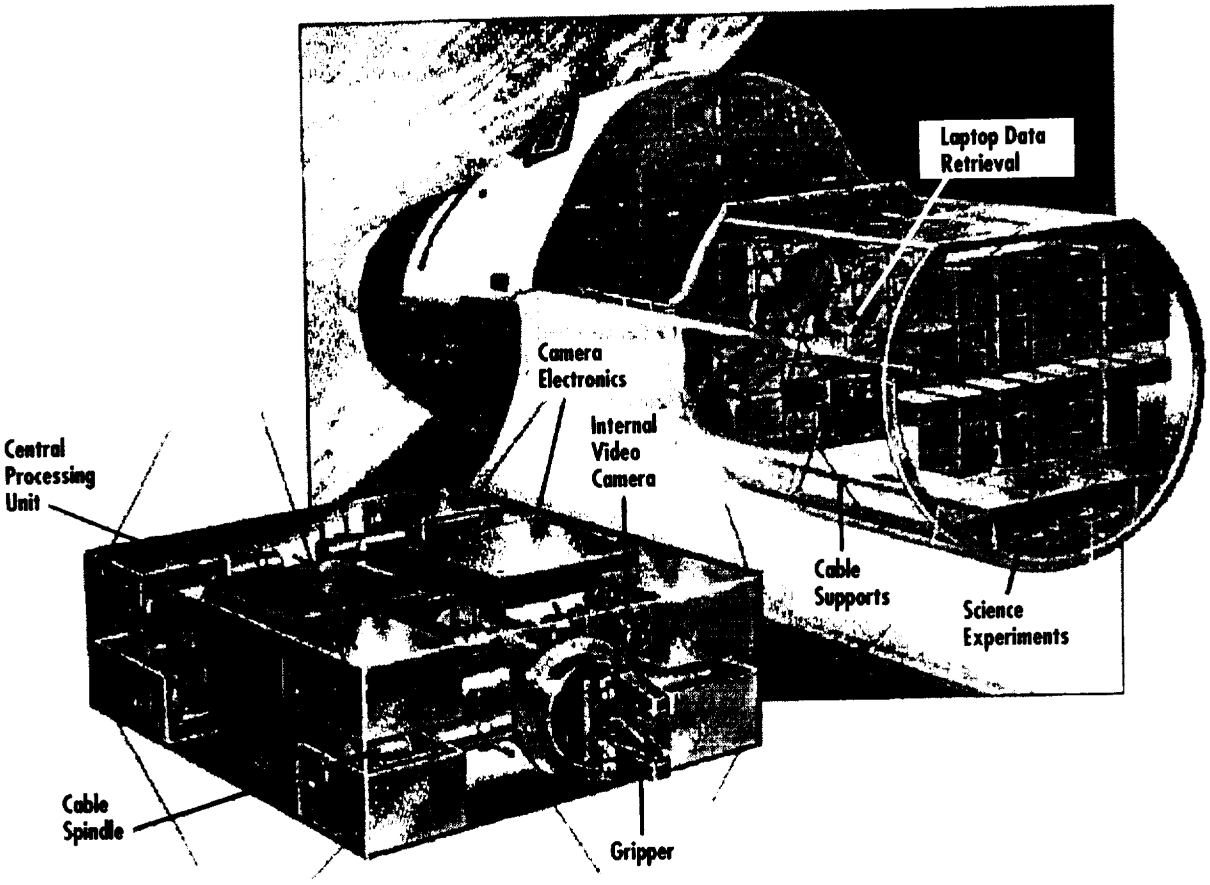


Figure 2. - The Charlotte IVA robot and its deployed configuration in the SpaceHab module.

system's ability to operate a representative set of SpaceHab experiment switches, buttons, dials, and knobs. A second set of scripts will demonstrate the robot's ability to perform video surveys of experiments in the SpaceHab module.

Following successful completion of the first phase of tests, ground command and control of the robot will be evaluated. Using the services of the SpaceHab and Space Shuttle data and communications subsystems, a second phase of testing will be initiated by ground controllers. The primary purpose of this testing will be to demonstrate the ability to operate the robot independently of the crew. This will enhance experiment monitoring and crew scheduling flexibility by enabling ground controlled operations during crew meal and sleep periods.

SCHEDULE

Final assembly of the Charlotte robot flight unit and early safety reviews for the SpaceHab 3 flight were completed by May 1994. A series of unit test and flight safety reviews remain to be conducted, culminating in a Flight Readiness Review in January 1995, with a expected launch in February of 1995 on the SpaceHab 3 mission on STS-63.

FUTURE SPACE INITIATIVES

Plans are underway for follow-on flights. We expect that successful completion of the first flight's objectives will lead to designation of the Charlotte robot as an operational subsystem of each SpaceHab module on subsequent missions. We are also planning a series of robot technology flight experiments to extend and enhance the system's capabilities. A project plan is under development that seeks to integrate robot technology developed at NASA centers and several small businesses with the basic Charlotte platform. Capabilities added may include a serpentine manipulator arm to enhance dexterity, ground telepresence control utilizing a virtual reality environment, and active on-board proximity and collision detection.

To facilitate this use of the Charlotte robot as an experimental test-bed, we intend to develop a number of industry-standard modular interfaces for structural, power, and data interconnects with the robot. Ultimately, it is hoped that this approach will lead to the development of a complement of end-effectors and tools that can be employed by researchers in conducting space telescience. To facilitate and expedite this development, we seek to develop a set of international ground test-beds, in which 1-g capable versions of the Charlotte system can be employed as research tools.

TERRESTRIAL APPLICATIONS

Many terrestrial uses of the system are also envisioned. Most of the desirable features of the Charlotte system transition well from space to terrestrial applications. The system is inherently scaleable, allowing us to consider both larger and smaller units. Large scale applications are envisioned requiring cable lengths of several hundred feet and payload capacities of several thousand pounds. The Charlotte robot might find applications in industries with requirements for systems with large workspace volumes, controlled transport, or precise positioning, such as aircraft production and maintenance, construction, and warehousing (Figures 3 and 4).

Smaller scale applications envisioned for the device include certain machining, materials handling, and laboratory applications. In general, Charlotte derived systems are best suited to applications with large, uncluttered

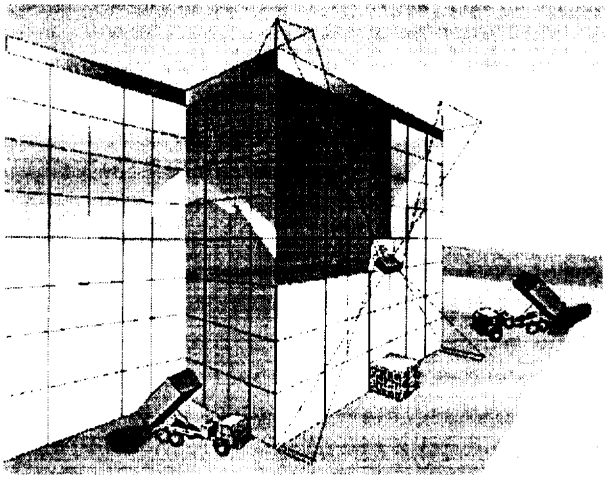


Figure 3. - A Charlotte robot could transport and install siding material for construction.

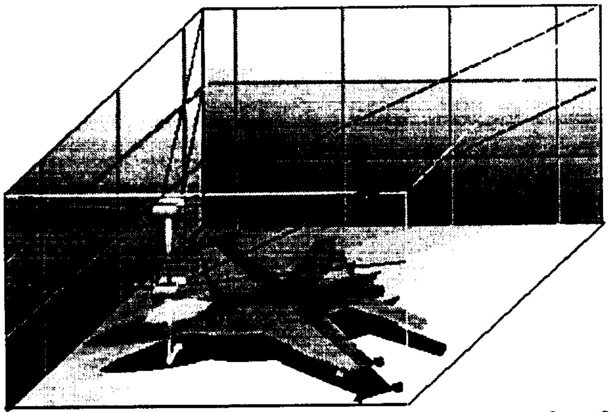


Figure 4. A Charlotte derived robot may be employed in single or multiple work cells to affect aircraft refurbishment and maintenance.

workspaces relative to the size of the objects to be manipulated, but with precise positioning needs; those which require only temporary use of robot; or in environments with evolving task or workspace requirements which need an easily reconfigurable robotic system.

CHALLENGING NEW FRONTIERS

One of the most exciting uses of the Charlotte robot currently in development employs it as a force feedback device in a man-in-the-loop simulator. The application is being developed by the Automation and Robotics Division of NASA's Johnson Space Center, and seeks to evaluate the use of virtual reality in astronaut training. This type of training must provide the appropriate visual environment and some of the sensory stimulus of weightless operations. Traditional training methods include flying parabola's in aircraft to achieve brief periods of weightlessness and, primarily, the use of neutral buoyancy facilities (water tanks) with immersed test subjects and hardware mockups. Underwater test facilities have several shortcomings, including the need to manufacture hardware mockups, the limited size of the tank, the cost to maintain and operate the facility, and the viscous damping effects which prevent objects from responding to applied forces as they would on-orbit.

The alternate approach under investigation uses virtual reality to simulate interaction with the visual environment, and uses the Charlotte robot to provide tactile sensory stimulus. Sensors are used to measure forces applied to the robot, a computer model computes the motion that would result, and the robot is commanded to move accordingly. Complicated dynamic interactions involving spacecraft systems can be modeled in the host computer. Reflecting this motion as movement of "virtual" objects in a helmet mounted display and physical motion of handholds or other crew interfaces mounted on the Charlotte robot allow the astronaut to "see and feel" simulated zero gravity effects. (Figure 5). Such simulators have the advantage of being easily reconfigurable to a variety of simulation scenarios with minor changes in data loads and visual models. Similar techniques can be applied to other training or entertainment applications.

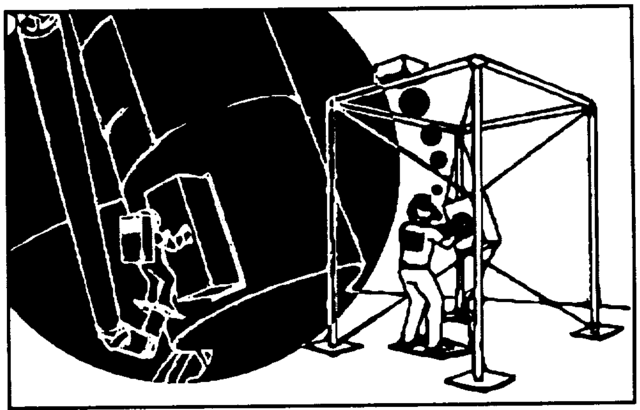


Figure 5. - A test subject wearing a helmet display in a laboratory experiences the exertion and visual sensations associated with an EVA task.

SUMMARY

A novel approach has been described to fulfill space intra-vehicular robotic needs. The solution is elegant in its simplicity, but surpasses other approaches in the intrinsic safety it provides and its ratios of workspace volume to weight and power requirements. The exceptional stiffness of the robot enables it to be highly precise, especially with regard to its workspace volume. Easily transportable, the device can be installed quickly, and its cables attach points can be configured to optimize performance for a variety of tasks. United States and international patents are pending.

SESSION RT

Robotics Technology

Development of Advanced Robotic Hand System for Space Application

Kazuo Machida*, Kenzo Akita**, Tatsuo Mikami***, Satoru Komada****

N95- 23704

*: Electromechanical Laboratory, MITI, 1-1-4 Umezono, Tsukuba, Ibaraki, 305 Japan
Phone: +81-298-58-5700, Fax: +81-298-58-5709, E-mail: machida@qm.etl.go.jp
**: Institute for Unmanned Space Experiment Free Flyer, Shinko-Building 2-12 Kanda Ogawamachi,Chiyoda-Ku, Tokyo 101, Japan
***: Fujitsu Ltd., 1015, Kamikodanaka Nakahara-ku, Kawasaki 211, Japan
****: Fujitsu Laboratories Ltd., 1015, Kamikodanaka Nakahara-ku, Kawasaki 211, Japan

KEY WORDS AND PHRASES

Robotics, robot hand, space robot, teleoperation, autonomy.

ABSTRACT

The Advanced Robotic Hand System(ARH) is a precise telerobotics system with a semi-dexterous hand for future space application. The ARH will be tested in space as one of the missions of the Engineering Tests Satellite VII (ETS-VII) which will be launched in 1997. The objectives of the ARH development are to evaluate the capability of a possible robot hand for precise and delicate tasks and to validate the related technologies implemented in the system. The ARH is designed to be controlled both from ground as a teleoperation and by locally autonomous control.

This paper presents the overall system design and the functional capabilities of the ARH as well as its mission outline as the preliminary design has been completed.

INTRODUCTION

The necessity of highly efficient, dexterous and versatile robot hands increases its importance for complicated and precise tasks of unmanned space facilities. To evaluate and validate related technologies of this kind of system, the Ministry of International Trade and Industry(MITI) has started the development of the ARH, which consists of a multi-degrees-of-freedom(DOF), multi-finger and multi-sensor robot hand and its supporting equipment.

The ARH will be experimented as one of the missions of the ETS-VII, which is developed by National Space Development Agency of Japan (NASDA), in order to evaluate key technologies such as dexterity and autonomy of robot hand control as well as to evaluate the capability for prospected in-orbit precise robot tasks.

Fujitsu, under the contract of MITI and under the supervision of related organization, has completed the preliminary design of the ARH. This paper reports the overall system design and functional capabilities of the ARH as well as its mission outline.

ARH SPACE EXPERIMENTS

The objectives of the ARH mission are to develop and evaluate the key technologies required for a next generation space robot which will be in charge of precise space tasks, and to validate the experimental robot system in which these technologies are implemented[1],[2]. To be more concrete, objectives are as follows:

1) Evaluate the capability of a multi-degree and multi-sensor robot hand dedicated to precise tasks required for unmanned systems or extra-vehicular activities(EVA).
2) Validate the space environment durability of mechatronic parts/devices for a space robot hand.
3) Master teleoperation skills and techniques under the communication restrictions such as limited communication capacity and time delay via a data relay satellite.
4) Acquire expertise of space robot control and operation in such space environment as weightlessness and visual monitoring restriction.

The concept of the space experiment system

164

is shown in Fig.1.

The characteristic of the space experiments is that the ARH will perform experiments for about one and a half years in the exposed space environment and will be engaged in precise tasks with a multi-fingered robot hand. Although there was a similar mission, Rotex, it was a several day experiment in the pressurized module, and its hand is a 1-DOF gripper. So the ARH may be the first space robot hand for precise tasks in EVA.

The space experiments of the ARH currently planned are divided into two categories; one is the experiments performed by the ARH system only, the other is the experiments in which the hand is attached to another robot arm(ERA) that will be mounted on the same satellite and be developed by NASDA. The experiments of the ARH only are as follows: 1) electric connector mate/demate, 2) fastening and loosening a bolt, 3) capturing of a floating object, 4) solar cell and thermal blanket expansion and handling,5) electric wire manipulation. The experiments by the hand attached to the ERA are as follows: 1) electric connector mate/demate, 2) inspection and handling of a experiment sample.

SYSTEM DESCRIPTION

The resource assignment of the ARH is very limited because ETS-VII has other main missions such as rendezvous-docking experiment and the ERA experiment. Therefore, the ARH is required to realize and perform above mentioned experiments within the assigned resources of about 44 kg weight and 500 x 480 x 500 mm envelope. In accordance with these restrictions, the preliminary design of the ARH has completed. The picture of the functional model is shown in Fig.2. The system configuration and system specification are shown in Fig.3 and Table 1 respectively. The flight segment consists of a hand, a control unit, a mini-arm, a task board and a task panel.The ground segment consists of workstations, a hand operation device, and monitor displays which show computer graphic images and real TV camera images.

The system has three operation mode. One is a teleoperation mode. Another is an onboard semi-autonomous mode, in which the hand and arm are controlled by an onboard program with the position correction using various sensor data. As a third mode, shared control between teleoperation and autonomous operation is also tried in the experiment.

The Hand

The basic design requirements of the hand are 1) to enhance the dexterity and versatility by employing a multi-finger/ multi-DOF hand , and 2) to increase onboard autonomy using multi-sensors. The hand designed is shown in Fig.4.

Considering the first requirement, the hand is designed to have three fingers with three DOF. One of them is a linear driven finger, the other two are rotary joint fingers. An object is grasped by three fingers. One of the fingers has an adaptable mechanism on its surface. A passive compliance device is installed to absorb position errors of manipulator arm. These mechanism will enhance the handling versatility, reliability and operability while decreasing processing loads of the onboard computer. According to the second design requirement, proximity range fingers, a hand-eye camera, grip force sensors, a compliance sensor and a force-torque sensor are embedded in the hand. Proximity range finders are mainly used for approach control to the task board. A CCD hand-eye camera is used to find the mark on the task board, and its image data is processed by the computer to calibrate the position errors. These sensor fusion technique will enhance the sensor based autonomy, and give a secure and flexible manipulation.

The Experiment Stage

The experiment stage consists of the mini-arm and the task board. The mini-arm with length of around 70 cm has R-P-P-P-R joints of five degrees so as to assure adequate movements in the limited mass and space resources. Each actuator has a harmonic drive which realizes 0.5 mm position accuracy. At the end of the mini-arm the tool that detaches or attaches the hand in space is equipped. The task board consists of four experiment panels where experiment parts are equipped on them. All these parts are locked so as not to be detached when it is launched, and are unlocked by the hand when the space experiments start. The hand and the mini-arm are locked separately on the base plane of the experiment stage when they are launched, and are released when the space experiments start. This hand release mechanism is also used when the ERA attaches or detaches the hand for the ERA-ARH experiments.

The Control Unit

The control unit consists of the processing computer and the power supply. The computer uses Intel 80386/387 as its MPU, which is responsible to control the mini-arm and the hand, and to process multi-sensor data as well as telemetry/command data. The computer includes a DSP board for mini-arm servo control. The sizes of ROM and RAM in it are 128KB and 256KB respectively.

The onboard software realizes or assists space experiments depending on operational modes. Its structure is shown in Fig.5. The software consists of OS, experiment program interface functions, and experiment programs. This software architecture enables experiment users to write experiment oriented programs independently from the other software while keeping the system safety. The software allows to be reprogrammed from the ground according to operational needs. As joint control parameters in space will be totally different from those of the ground due to missing gravity, in-orbit calibration using various sensors will be required. Thus these parameters are also uploaded from the ground.

The data link between space and ground is, down-link wise, the computer of the ARH, the satellite communication equipment, the data relay satellite, the ground station and the ground control facility of the ARH. The communication rate of 800 bps (4Hz) is allowed for teleoperation command. The down-link rate is 1.5 Mbps which include compressed TV image and telemetry data. A total time delay of 2 to 4 seconds is expected for the data link.

The Ground System

The ground system has the function of the supervising space robot system as well as processing telemetry and command data providing operators necessary information by a model based simulation, which compensates communication time delay and limited onboard visual information. The computer graphic simulation displays images of 3-D solid-shaded polygonal rendering. The ground system configuration is shown in Fig.6.

An operator can manipulate the master control device to control onboard mini-arm and the hand as a teleoperation control assisted by the task visualization of preview and prediction, which enhances the operation safety and efficiency. In this mode the onboard computer adjusts small errors of modeling by feedbacking hand-eye camera and proximity range finder data. In another operation mode of the ground system, an operator sends pre-programmed commands that controls the hand and mini-arm autonomously. The hybrid operation of these two modes is supported by remote-end skill and local adjustment, which will be effective to accomplish precise and complex tasks. The ground system has monitoring functions of down-linked TV camera images as well.

SUMMARY

The outline of the Advanced Robotic Hand System and its mission is presented. Its preliminary design has completed and the engineering model is under development. The ARH is a small system in size, but it includes many key technologies of sensors, mechanisms and control architectures for advanced space robot performing precise tasks.

REFERENCES

[1] Machida, K. et.al.,1993. Preliminary Design of Advanced Robotic Hand for Space Experiment on ETS-VII. International conference of Advanced Robotics '93.

[2] Machida, K. et.al.,1993. Development of Dexterous Service Robot for Space Testing(1), The 11th Conference of the Robotic Society of Japan

Table 1. ARH Specification

Total system mass	44 kg
Dimension(Envelope)	500 x 480 x 500 mm
Average Power	85 W
Hand DOF	3
Grip force	20 N
Compliance	0.17mm/N(x,y),0.08mm/N(z),2.9deg/N
Mini-arm DOF	5
Accuracy	±0.5mm, ±0.5deg
Tip force	20 N
Communication rate	
Teleoperation command	800 bps
Real time telemetry	1.5 Mbps(including video data)
Mission period	1.5 years

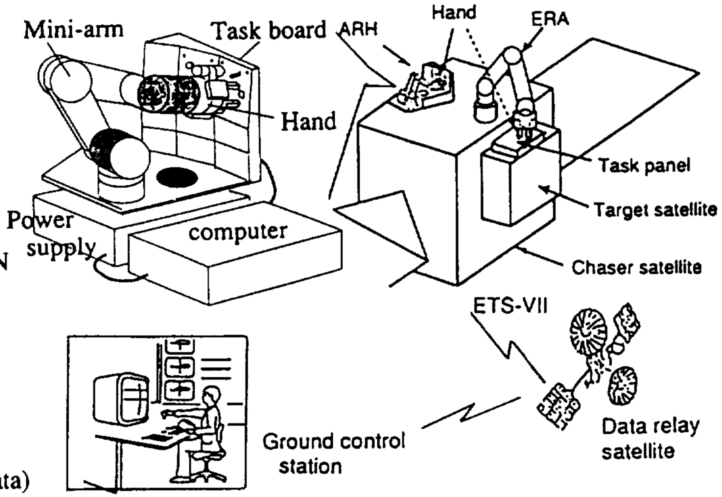


Figure 1. ARH System Concept

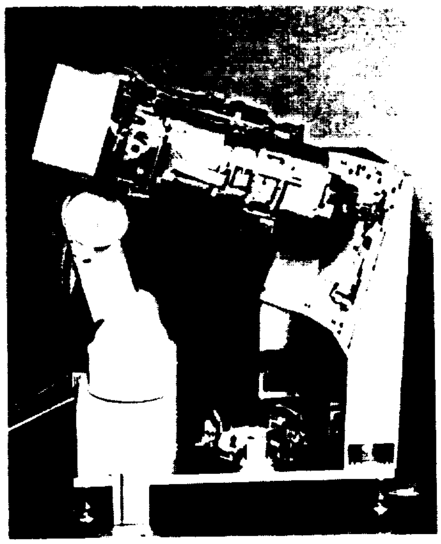


Figure 2. ARH Functional Model

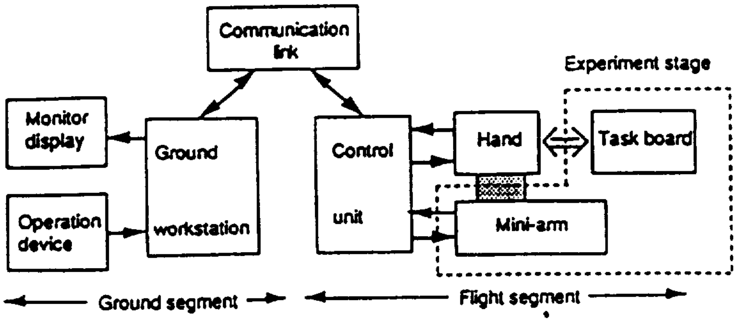


Figure 3. System Configuration

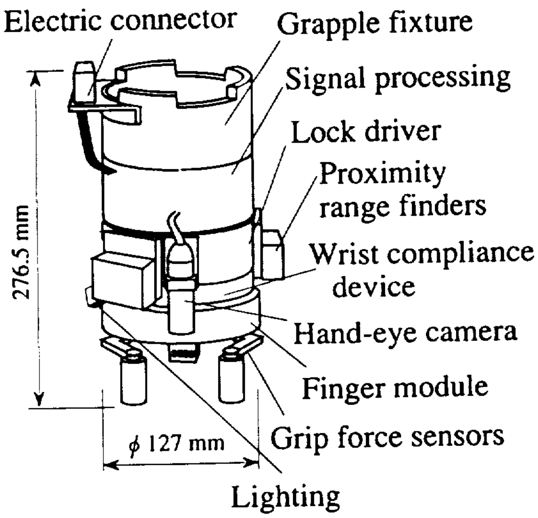


Figure 4. The Hand

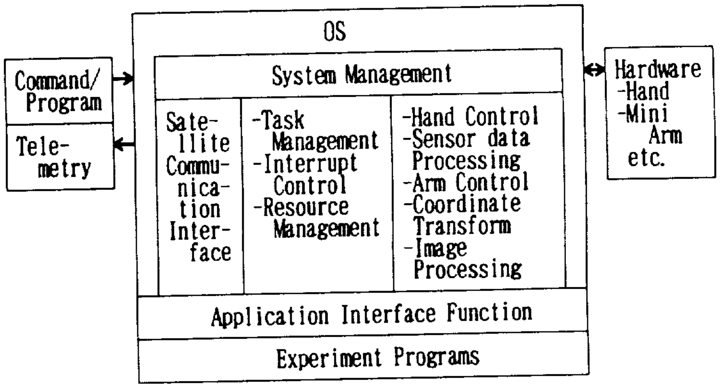


Figure 5. Onboard Software Structure

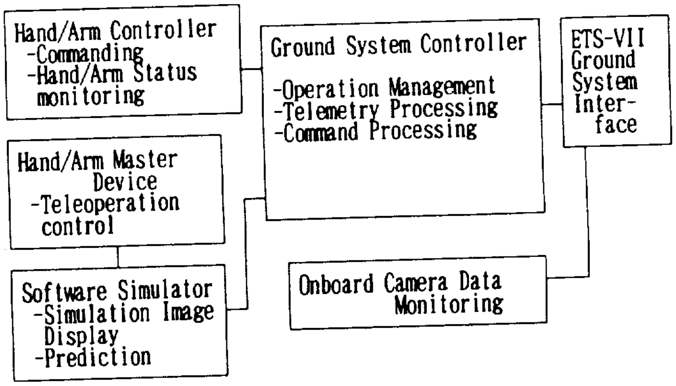


Figure 6. Ground System

[redacted]

A Three-Finger Multisensory Hand for Dexterous Space Robotic Tasks

Yuichi Murase, Satoru Komada, and Takashi Uchiyama
Fujitsu Laboratories Ltd.
1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan
Tel. : +81-44-754-2653 Fax. : +81-44-754-2582
E-mail : madonna%lucy. stars.flab.fujitsu.co.jp@fai.com

Kazuo Machida
Electrotechnical Laboratory, MITI

Kenzo Akita
Institute for Unmanned Space Experiment Free-Flyer

N95- 23705

KEY WORDS AND PHRASES

" Extravehicular robot, local autonomy, multisensory hand, space robot "

ABSTRACT

The National Space Development Agency of Japan will launch ETS-VII in 1997, as a test bed for next generation space technology of RV&D and space robot. MITI has been developing a three-finger multisensory hand for complex space robotic tasks. The hand can be operated under remote control or autonomously. This paper describes the design and development of the hand and the performance of a breadboard model.

INTRODUCTION

As an activity in space increases, robot will play a bigger role in building space stations and performing experiments. In particular, robot will have to perform delicate and complex tasks, such as arranging and servicing equipment in unmanned space facilities. A key component is a hand with dexterous and adaptable capabilities. We are developing a hand, which we call the Advanced Robotic Hand (ARH), for Engineering Test Satellite VII (ETS-VII), which is a test bed for next generation space technology. ETS-VII will be launched by NASDA (National Space Development Agency of Japan) in 1997 [1], [2]. Our hand has a multiple degree-of-freedom (DOF) mechanism, multiple sensors and onboard control. Robots have already had some success working in space. Recently, a ROTEX robot with a multisensory hand completed a space experiment in Spacelab D-2. The ROTEX works inside a space vehicle for a week. Its hand is a 1-DOF gripper [3]. The ARH is an extravehicular robot with a multi-DOF hand that works in the exposed environment of space for 1.5 years. The ARH will increase the applications of robots in space. This paper presents the design of the ARH and the performance of a breadboard model of the hand design.

ADVANCED ROBOTIC HAND SYSTEM

Figure 1 illustrates the ARH system. The hand is attached to a 5-DOF mini-arm. Parts for demonstration tasks in space were placed on a task board. This task board will be installed on the outer wall of the satellite. The hand is controlled by remote control or by autonomous control.

DESIGN OF THE HAND

If a remote robot in space is controlled from the ground, signal transmission delay lowers safety and efficiency. The ARH has multiple sensors that en-

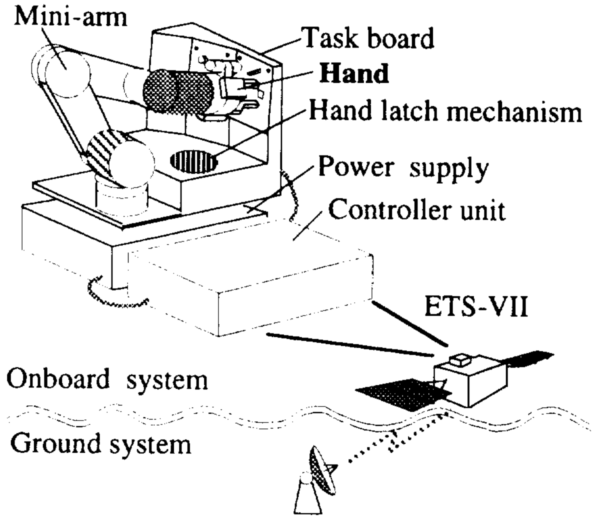


Fig. 1 Advanced robotic hand system

able it to adapt to various tasks and works autonomously. **Figure 2** shows the configuration of the hand. The finger module holds the part being manipulated and the wrist compliance device compensates for hand position errors. The signal processing module drives motors with 5-ms cycle serial control signals. It also processes sensor signals. The hand is linked electrically and mechanically to external equipment via the tool fixture. The hand is equipped with a hand-eye camera, three proximity range finders, 3-DOF wrist displacement sensor, and two grip force sensors. The mini-arm is equipped with a 6-DOF force/torque sensor at the wrist.

Finger Module

The fingers form a gripper which is simple, reliable, and finds the gripping position easily. The finger module has one linear-movement finger, A, and two rotary fingers, B and C, arranged as shown in **Fig. 3**. As the figure shows, the finger module has three degrees of freedom: α, β, and γ. The shape profiler on the linear-movement finger ensures that the surface of the finger fits various profiles, and makes grasping more stable. The finger has multiple pins arranged in a grid that move linearly on the surface of the finger, as shown in **Fig. 4**. Each pin is pressed down a maximum of 3 mm by the gripped object. Prestressing springs equalize the grasping force and press the object and grip it firmly. The grip force sensor attached to the rotary finger is designed with isotropic output characteristics. This enables proper gripping force sensing, irrespective of force direction changes of finger B and C. Strain gauges detect bending moments in two directions of the L-shaped link, as shown in **Fig. 5**. Let V_1 and V_2 be the detection voltages from the strain gauges, F be the object grasping force, and K_1, K_2, and K_3 be the output characteristic coefficients. The detection voltages are then as follows :

$$V_1=K_1F\sin\beta$$
$$V_2=K_2F\cos\beta+K_3F\sin\beta$$

The grasping force F is given by the following equation :

$$F=\frac{\sqrt{(K_2V_1)^2+(K_1V_2-K_3V_1)^2}}{K_1K_2}$$

Table 1 lists the grip capability.

Compliance Device

Figure 6 shows the extremely thin compliance device. It contains flat springs which deform to compensate for positional deviations in the four degrees of freedom X, Y, Z, and θz. Displacements in X, Y, and θz are monitored using the signals from strain sensors attached to the surfaces of flat

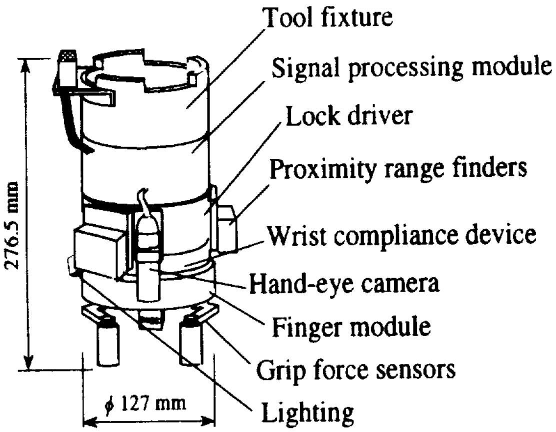


Fig. 2 Configuration of the hand

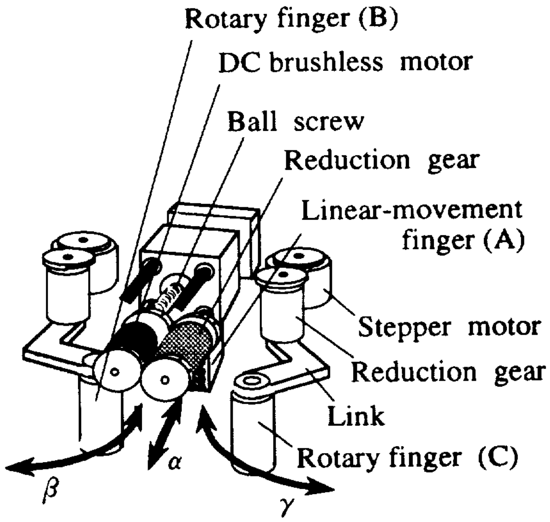


Fig. 3 Finger module mechanism

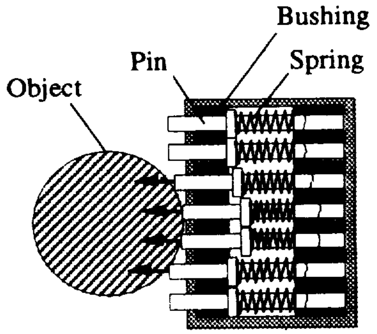


Fig. 4 Shape profiler

PERFORMANCE OF A BREADBOARD MODEL

Figure 8 is a photograph of the breadboard model. The grip force sensor gave a uniform output for all directions of a 19.3 N load (see **Fig. 9**). Variation of ±5% in range of 0 to 260 degrees is acceptable. **Fig. 10** (a) and (b) show finger position and grip force data for fingers in the two-finger coordination mode respectively. Finger A and a pair B/C move the object right and left while grasping it. Fingers B and C position the object in position control mode, and finger A grasps the object in force control mode.

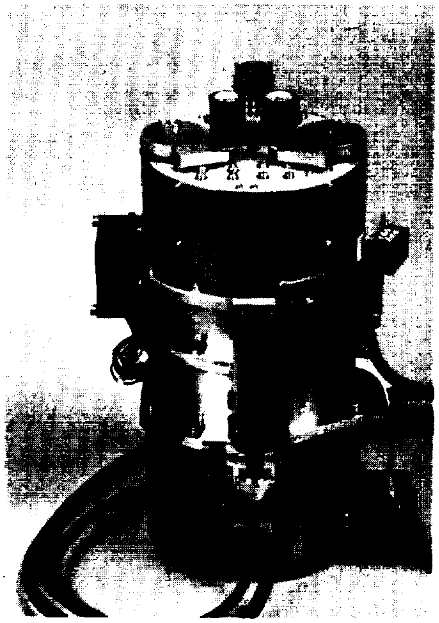


Fig. 8 Breadboard model

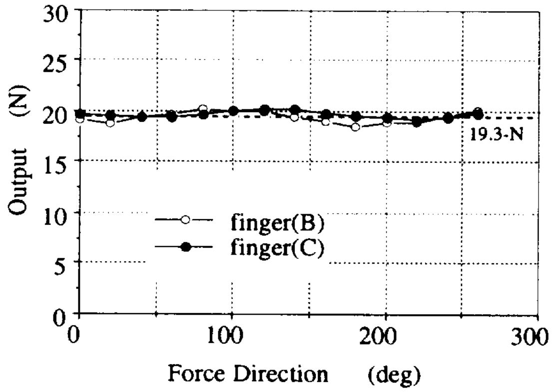


Fig. 9 Directional characteristics of grip force sensors

SUMMARY

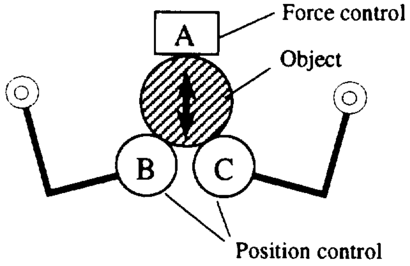
We developed an experimental gripper-type 3-finger 3-DOF hand with multiple sensors for space applications. The hand features a hand-eye camera, three proximity range finders, two grip force sensors, and a wrist displacement sensor. The multiple sensors make autonomous operation possible.

REFERENCES

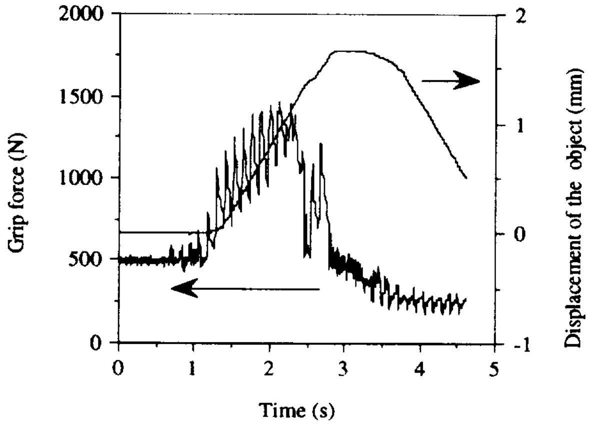
[1] Oda, M.; Wakabayasi, Y.; Imai, R.; and Anzaki,T., 1992.ETS-VII, The World First Telerobotics Satellite(Misson and Its Design Concept). *i-SAIRAS'92*

[2] Machida, K.; and Ono, S., 1992. Space Experiment Plan of Advanced Robotic Hand for ETS-VII. *i-SAIRAS'92*

[3] Brunner, B.; Hirzinger, G.; Landzettel, K.; and Heindel, J., 1993. Multisensory Shared Autonomy and Tele-Sensor-Programming Key Issues in the Space Robot Technology Experiment ROTEX. *Proc. of the IEEE Conf. on Intelligent Robotic and Systems*, Yokohama, July 26-30



(a) Control of the fingers



(b) Results of tests

Fig. 10 Two-finger coordination motion

springs. The compliance device has a locking mechanism operated by torque from the lock driver. The locking mechanism prevents the finger module vibrating when it moves. **Table 2** lists the specifications of the wrist compliance device.

Sensor Based Control

Multiple sensors are used in sensor based control to perform accurate and reliable work, as shown in **Fig. 7**. Three proximity range finders measure the distance to an object in the proximity area within 80 mm with an accuracy of 1 mm. The sensors are used for approach control and orientation control of the hand to face a task board. A CCD hand-eye camera is used for object recognition by image processing. The hand sets the local work coordinate with these functions of the proximity range finders and hand-eye camera. Noncontact sensing is effective for recognition over a wide area, and is used to navigate the hand. Grip force sensors are equipped to control the grip force. The sensors also detect the position of the object accurately with a touch-and-identify strategy. A 6-DOF force/torque sensor is equipped to measure the external force applied to the hand. An external force is always observed so that the tasks are carried out safely. Wrist displacement sensors measure small position errors of the hand with high accuracy. Another important role of the sensor is to measure the external force with the stiffness data of the wrist compliance device. This measurement is more sensitive than that of the wrist force/torque sensor. This function is especially useful for delicate tasks such as parts assembly.

Table 1 Grip capability

Finger	Grip force : Max 20 N
	Object size : ϕ 8~50 mm
Sensor	Range : Max 25 N
	Accuracy : 0.7 N

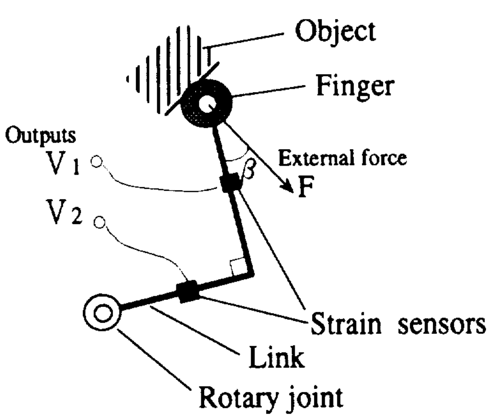


Fig.5 Grip force sensor

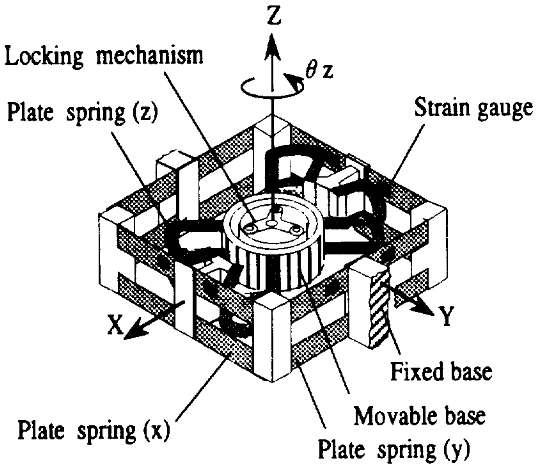


Fig.6 Wrist compliance device

Table 2 Compliance device specifications

Movement range	x, y, z : ±1.5 mm
	θz : ±2 deg
Compliance level	x, y : 0.17 mm/N
	z : 0.08 mm/N
	θz : 2.9 deg/Nm

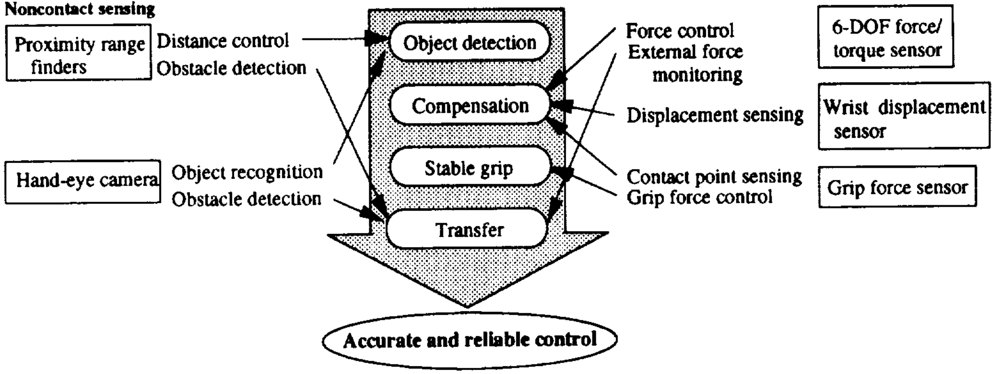


Fig. 7 Sensor based control

N95- 23706

HARDWARE INTERFACE UNIT FOR CONTROL OF SHUTTLE RMS VIBRATIONS*

Thomas S. Lindsay, Joseph M. Hansen,
Davoud Manouchehri, and Kamran Forouhar.

Rockwell International Space Systems Division
12214 Lakewood Boulevard, M/C DA37
Downey, California 90241
TEL (310) 922-2477 FAX (310) 922-0537
e-mail lindsay@hercules.dny.rockwell.com

KEY WORDS AND PHRASES

End point control unit, robotics, vibration isolation.

ABSTRACT

Vibration of the Shuttle Remote Manipulator System (RMS) increases the time for task completion and reduces task safety for manipulator-assisted operations. If the dynamics of the manipulator and the payload can be physically isolated, performance should improve. Rockwell has developed a self-contained hardware unit which interfaces between a manipulator arm and payload. The End Point Control Unit (EPCU) is built and is being tested at Rockwell and at the Langley/Marshall Coupled, Multibody Spacecraft Control Research Facility in NASA's Marshall Space Flight Center in Huntsville, Alabama.

INTRODUCTION

Robot manipulators with long flexible links, such as the Space Shuttle RMS, are susceptible to unwanted vibrations. These vibrations increase task completion times and reduce task safety. To reduce the vibrations, many arm controller architectures have been presented, including input shaping, adaptive control schemes, and fuzzy logic control. These methods show improved manipulator performance. This may be an acceptable solution for ground-based manipulators. For the Shuttle RMS, however, redesign of the controller would require a re-certification for space flight of both RMS software and hardware, an extremely expensive proposition.

Instead of redesigning the manipulator controller, a hardware device located between the arm and its payload can be used to physically decouple the system dynamics. Improved performance can be realized because the dynamics of the payload cannot adversely affect the manipulator arm, and vice-versa. A complete decoupling of the manipulator from the payload is not desired: the payload must still respond to desired motion of the arm. An intermediate level of isolation is desired.

Rockwell has developed an end-effector for the NASA robotic tile

---

* This work was partially funded by NASA Langley Research Center Contract NAS1-19243 to Rockwell Space Systems Division and was monitored by Dr. Raymond Montgomery of LaRC Spacecraft Controls Branch.

processing system (RTPS) which decouples the dynamics of the elevation arm from the force applied to the Shuttle tiles during rewaterproofing operations[1]. The sections presented below briefly describe the RTPS end-effector and its derivative, the EPCU.

THE RTPS END-EFFECTOR AND THE EPCU

The RTPS end-effector isolates the damping of the RTPS elevating arm from the shuttle itself. In this way, the necessary constant force can be maintained on the Shuttle tiles. The end-effector uses a stepper motor for one degree of linear actuation, and an encoder and a 6-axis force/ torque sensor for control feedback. This end-effector has been tested under numerous adverse conditions such as attempting to maintain a constant contact force with a Shuttle tile while being subjected to vibratory inputs [2]. Because the tests illustrated the utility of this device, the next generation end-effector (the EPCU) was designed to meet a variety of vibration isolation and dynamic decoupling problems.

The current design of the EPCU is shown in Figure 1. The unit has four major subcomponents: 1) stepper motor drive mechanism, 2) constraints for linear motion, 3) force and position feedback sensors, and 4) controller hardware and software. The control software is described below. The EPCU is a completely self contained unit and needs no inputs from the manipulator controller.

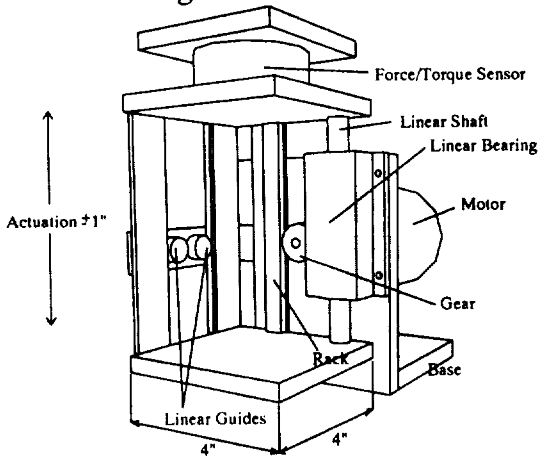


Figure 1: The Rockwell EPCU

THE EPCU DECOUPLING CONTROL PROBLEM

The EPCU control configuration is illustrated in Figure 2. Unlike the classical control problem of reducing the effects of the disturbance signal, the current system needs to react to certain disturbance signals (the undesired manipulator arm motions) yet allow other signals to pass to the payload. The disturbances to be suppressed are those close to the natural frequency of the RMS which are caused by structural bending or flexibility of the manipulator links.

The controller in Figure 2 consists of two inputs and one output. Input signals are from the force sensor and the shaft encoder. The output of the controller is a position or velocity signal which commands the stepper motor driving the EPCU.

Four control algorithms have been developed and tested with the hardware on a small-scale test bed in Rockwell's Robotics Laboratory. The first controller uses the force feedback to compute a desired EPCU deflection based upon a given spring stiffness value K for the unit. The current EPCU position is subtracted from this desired position to compute an error signal, which is multiplied by a gain to command the EPCU

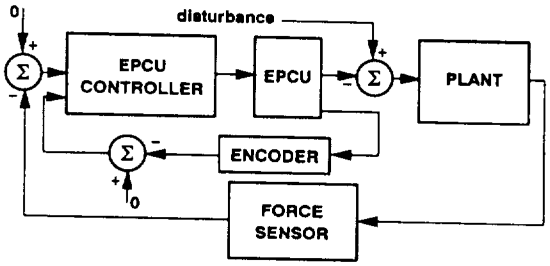


Figure 2: Control Structure Block Diagram

motor as a velocity signal. The second controller uses this same algorithm but conditions the force feedback signal by subtracting the current value from the mean of the previous n values. This eliminates the effects of both static forces and force sensor drift. The third controller conditions the force feedback signal with a bandpass filter and a delay filter to allow only those frequencies deemed to be "problem" frequencies to be controlled by the EPCU. This feedback is used in a standard PD controller. The fourth controller uses fuzzy logic to determine the velocity output from force and position feedback signals.

Each of these controllers suppressed the undesirable vibrations and disturbances. Some gain adjustments were required for different payload weights. The adaptivity of the controllers can be managed via gain scheduling.

ROCKWELL TESTBED TEST RESULTS AND CONCLUSIONS

Figure 3 shows the results of a typical EPCU test run at the Rockwell testbed, illustrating the resultant payload motion from a 1 Hz vibration input. The payload motion response to the vibratory input is reduced by over 50% by using the EPCU. Other tests show the favorable response to a sinusoidal input superimposed with a constant velocity input. Test run data shows that the motion characteristics of the payload are improved, and that the unit can partially decouple the system dynamics, demonstrating a successful implementation of the EPCU to reduce the effects of unwanted system vibrations upon system performance. After testing at Rockwell, the unit was integrated into the Langley/Marshall flat-floor testbed, which is described below.

THE LANGLEY/MARSHALL TESTBED

The NASA Langley/Marshall Coupled, Multibody Spacecraft Control Research Facility contains a 2-link, 3-joint planar manipulator supported by air-bearings on a flat-floor test facility [3](see figure 4). The manipulator payload is a large sled supported by air bearings, controlled by onboard control motion gyros (CMGs) and air reaction jets. The system represents the Shuttle RMS docking with a controlled structure such as the space station. The links are 9.75 feet in length, and are designed to have a resonant frequency approximating that of the RMS. The payload weighs approximately 3700 lbs. Currently, the system shows vibrations and unwanted transient motions for a typical motion test run.

The EPCU is inserted between the manipulator arm and the payload, as shown in Figure 4, and the same tests which showed unsatisfactory behavior were performed to determine if the EPCU unit improves the system performance. A fifth controller was developed to augment the first controller with the additional input of the desired velocity

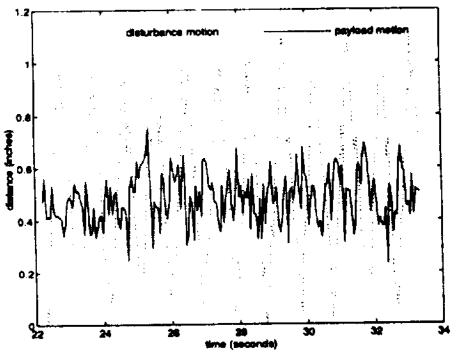


Figure 3: 1Hz Vibration Input and EPCU Motion in Rockwell Testbed

trajectory of the manipulator arm in the direction of EPCU actuation. Based upon this input, feedforward signals were generated to help the EPCU react faster to discontinuities in the velocity profile. Test results at the flat floor facility show improved performance both in terms of reduced forces on the payload and improved position tracking.

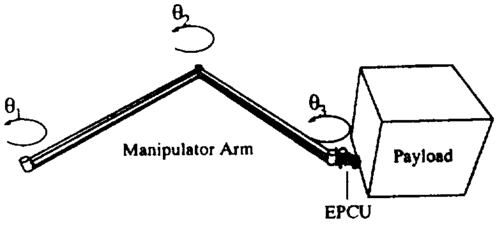


Figure 4: EPCU Integrated at Langley/Marshall Testbed

APPLICATIONS AND FUTURE WORK

The utility of the interface unit on the Shuttle RMS is evident from the above discussions. Further space applications of the device include isolation of antennas and solar panels from a satellite and isolation of payloads from Shuttle vibrations during ascent flight. Similar vibration isolation/control and dynamic system decoupling is needed for other applications of long manipulator arms, including Department of Energy waste cleanup as well as industrial uses. The device may also prove beneficial for reducing vibrations and impact forces in devices with less accurate control, such as cranes and winches. The implementation of a self-contained or near- self-contained active interface device for these applications will improve system performance without intelligent human interaction or advanced system control techniques. An addition-al application is in the suspension of automobiles, providing for a smoother ride and improved performance of the vehicle for a large range of loads.

REFERENCES

[1] Manouchehri, D.; Hansen, J.M.; Wu, C.M.; Yamamoto, B.S.; and Graham, T., 1992. Robotic End-effector for Rewaterproofing Shuttle Tiles. In *Proceedings of SPIE OE/Technology '92 Conference*, Boston, MA.

[2] Hansen, J. M.; and Manouchehri, D., 1993. *Positioning System/Active Compliance*, NASA 10-11500 Provisioning Order 1088, WBS 11 Task 2 Final Report, Rockwell International Document # SSD93F0045.

[3] Montgomery, R.C.; Tobbe, P.A.; Weathers, J.M.; Ghosh, D.; and Garrison, J.L., 1993. A Testbed for Research on Manipulator-Coupled Active Spacecraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, CA.

ROBOTIC EXPERIMENT WITH A FORCE REFLECTING HANDCONTROLLER ONBOARD MIR SPACE STATION

N95- 23707

M. DELPECH, Y. MATZAKIS,
CNES
18 Avenue Edouard Belin
31055, TOULOUSE Cedex, France

E. ROUCHOUSE
Matra Marconi Space
31, rue des Cosmonautes
31077 TOULOUSE Cedex, France

ABSTRACT

During the French CASSIOPEE mission that will fly onboard MIR space station in 1996, ergonomic evaluations of a force reflecting handcontroller will be performed on a simulated robotic task. This handcontroller is a part of the COGNILAB payload that will be used also for experiments in neurophysiology. The purpose of the robotic experiment is the validation of a new control and design concept that would enable to enhance the task performances for telemanipulating space robots. Besides the handcontroller and its control unit, the experimental system includes a simulator of the slave robot dynamics for both free and constraints motions, a flat display screen and a seat with special fixtures for holding the astronaut.

INTRODUCTION

When robot manipulators are being used in unstructured environments, telemanipulation represents either the nominal or at least the contingency mode of operation. Kinesthetic force feedback constitutes then a classical feature to enhance task performances when time delay is not a problem.
Several constraints, however, limit the introduction of force reflecting devices for teleoperating robots in space:
- the device working area must remain small enough for accommodation reasons and this prevents the use of classical 6D anthropomorphic structures,
- the dynamics of large external manipulators such as the Shuttle RMS is much slower than the operator hand, this reduces the reflected force bandwidth and so the benefit of the device,
- the computing power necessary for achieving satisfactory performances has to be very high,
- the microgravity obliges to introduce special astronaut holding equipment.

Passive devices remain then the baseline specially after the success of the ROTEX control ball [1] which has brightly proven its efficiency when coupled to a shared control robot. Such facts force to reconsider the kinesthetic force reflecting technique from a different point of view. This paper introduces a new control and design approach that addresses some of these problems. It presents then the device developped according to this approach and the experiments that will be performed in space to evaluate the ergonomy of its utilization for robotics.

HAND CONTROLLER DESIGN APPROACH

The vast majority of robotic tasks can be represented by a sequence of elementary actions, each involving motions along at most 2 or 3 axes simultaneously.
It has been taken advantage of this property in advanced telemanipulation systems where the

operator is offered a variety of control modes that allow mobility within only a subset of the cartesian space. To perform a drilling task for instance, after adjusting the orientation and position of the driller, the operator needs to keep control along the drilling axis only, the other axes are being blocked during the operation. In such a way, computer aided teleoperation enhances task performances since the operator can concentrate his perception and actuation abilities on the most rewarding part of the job. Those remarks can trigger a discussion about the necessity to provide operators with 6 d.o.f. hand controllers when half of them are supposed to be blocked most of the time.

The alternative we are proposing consists in using 3 d.o.f. force reflecting joysticks.
The advantages of such simpler mechanisms are numerous:
- the compacity of the structure makes its accomodation more realistic for space vehicles,
- the smaller envelope prevents the operator from reaching uncomfortable positions,
- the stiffness and the dynamics can be significantly increased, thus allowing better performances,
- the computational cost of forward / inverse kinematics is reduced and alleviates the implementation requirements.

For controlling 6 d.o.f. robots, the operator is provided with a set of two complementary 3 d.o.f. joysticks: one for the translations, the other for the rotations. This system being operated with both hands enables then to control a robot in free space like any classical 6 d.o.f. serial mechanism. The performances may be even better since translation and rotation motions are decoupled. When doing constrained motions, the coupling between the two joysticks appears however in a rather remarkable way. Let us consider an operator inserting a peg in a hole by moving only the translation joystick: if there is some orientation error a resistive force will be applied by the joystick to his controlling hand and at the same time he will feel some force in its idle hand generated by the rotation joystick. He may resist to this force and then block the peg or comply and allow the orientation correction. In this latter case, one hand is the "controller" and the other one is the "follower". Our opinion that needs to be confirmed by experimentation is that the operator, after some training, will better interpret multi component forces. For that purpose, a complete telemanipulation system involving such joysticks is under development and should be ready within months.

Besides this utilization, this kind of device is specially relevant for shared control modes already described by Hirzinger [2] or Hayati [3] since it will provide force feedback in the operator controlled subspace.

HANDCONTROLLER PRESENTATION

The 3 d.o.f. active joystick presented here-below has been developped to serve two purposes:
- analysis of human neuromuscular models,
- robot telemanipulation.

since the requirements were convergent in terms of kinematics and performances. Table 1 shows the joystick present characteristics.

Axes Features	X, Y	Z (Rotation)
Working envelope	+/- 120 mm	+/-120°
Maximum force	25 N	0.6 Nm
Residual Friction	< 1N	<0.03 Nm
Maximum speed	0.5 m/s	200 °/s
Maximum stiffness	10000 N/m	200 Nm/rad

Table 1

The selected kinematics with 3 rotations (Figure 1) enables no dynamic coupling between the axes. The actuation is provided by servo-motors through Harmonic Drive gears. To cancel the residual gear friction, active compliance is implemented on the joystick controller and relies on a 3 d.o.f. force/torque sensor located beneath the handle. Joystick control is based on a 68040 CPU board and runs at a high rate.

For doing force feedback evaluation experiments, the joystick system is linked via VME bus to a simulator running on a second 68040 board (Figure 2). The typical control scheme being used for implementing force reflection is presented on Figure 3 (pure force feedback) and is achieved at a medium sampling rate for realistic simulations.

However, as long as simulation is concerned, it is possible to implement higher sampling rate systems and so increase the force signal bandwidth by running at high in the joystick controller a simple interaction model whose parameters are computed by the simulator and updated with the force at medium rate. This enables to emulate systems running at higher frequencies.

The stiffness characteristics from Table 1 have been obtained according to this method for an infinitely stiff and light robot interacting with a pure spring.

EXPERIMENT DESCRIPTION

Objective

The purpose of this space experiment involving a single 3 d.o.f. joystick is twofold:
- to evaluate the ergonomy of synthetic force reflection with and without shared control
- to assess its potential benefit w.r.t. other techniques (use of passive devices such as ROTEX control ball).

Harware description

The experimental system includes the following components accommodated inside one of the MIR modules (Figure 4):
- the astronaut seat that constitutes the structural part of the system and that is fixed in the present design to the module floor.
- the motorized joystick,
- the experiment calculator including the joystick controller, the simulator computer and a graphic board,
- a flat display screen and a optical tunnel to eliminate the visual distractions.
- a handle with switches to control the experiment.

The spaceflight model of the joystick is based on ground technology: except for specially developped power electronic boards, the other elements are only hardened to satisfy the mechanical, thermal and safety requirements.

The calculator is VME based and includes standard CPU boards (MVME 162 with mezzanine IO boards) for both joystick control and simulation/experiment management.)

Experiment protocol

Robotic task

The robotic task to be performed is a "peg in a hole insertion".that involves a simulated robot interacting with a virtual environment. The robot is a 3 d.o.f. mechanical system that enables to move its end effector within a plane (2 translations along the X, Y axes and a rotation for its orientation). Figure 5 presents the model of this task. Using the joystick, the operator has to displace the peg in front of the hole, adjust its orientation and insert it smoothly until it touches the bottom. He monitors the robot displacement by watching a 3D graphic display of the scene that is representative of an image coming from a global view camera (figure 6).

The simulation includes the following features:
- the robot dynamics is finite (represented by a second order transfer function on all axes),
- the tool (peg) is attached to the robot by some compliant interface (compliance along 3 axes),
- contact interactions such as jamming effects can be represented: the obstacle stiffness is considered infinite and the only structural deformations take place at the compliant interface.

The simulation process runs in 12 ms: the force reflecting loop is closed at 75 Hz but the joystick model based joystick control runs up to 750 Hz..

The operator is asked to insert the peg in the minimum time while keeping the contact forces as low as possible: the performance criterion is a combination of those two informations.

Modes of operation

Three modes of operation are considered:

- Velocity control with visual force reflection (Mode 1)
- Position control with kinesthetic force reflection along all axes (Mode 2)
- Position control with kinesthetic force reflection along translation axes only (Mode 3).

- Mode 1 simulates the way ROTEX manipulator was operated by the astronaut within the Spacelab module [1]. The joystick is blocked in a central position to emulate a 3 d.o.f. "control ball" and force information is displayed on the screen using 3 bars (Figure 5). The slave robot moves under shared control: active compliance is provided along the orientation axis when contact is achieved.
- Mode 2 represents classical kinesthetic force feedback where all axes are controlled by the operator.
- Mode 3 is an example of kinesthetic force reflection applied in a shared control scheme. The slave robot is controlled like in Mode 1 but now the operator feels the forces along 2 degrees of freedom (X, Y).

These 3 modes will be used for performing the insertion task with two types of simulated robots:

- a high dynamics structure corresponding to some small servicing manipulator
- a low dynamics structure representative of long external manipulators.

This will make a total of six different control configurations for the experiment.

Three astronauts will participate in the experiment during the 11 days flight mission. Each astronaut will perform a specified number of repetitions of the task in the different control configurations (a minimum of 10 repetitions is required to allow a valid statistical analysis). In order to compare the obtained results with a fair reference so that the influence of gravity can be identified, the astronauts will perform exactly the same tests on ground before the mission.

CONCLUSION

The experiment presented in this paper constitutes a first shot in the evaluation of kinesthetic force reflecting techniques for teleoperation in space. We expect to demonstrate that the technique is not only feasible but enables to improve task performances when implemented with small 3 d.o.f. joysticks. However, the main purpose is the collection of experimental data for performing ergonomic analysis. It will permit then to improve the design of a complete 6 d.o.f. system (two joysticks) and to get ready for a full scale demonstration with a real space robot.

[1] G.Hirzinger and als.
The sensory telerobotic aspects of the space technology experiment ROTEX. iSAIRAS, Toulouse, September 1992.

[2] G. Hirzinger 1989.
Multisensory shared autonomy - a key issue in the space robot technology experiment ROTEX, IEEE Conference on Intelligent Robots and Systems (IROS), Raleigh, USA, July 92.

[3] S. Hayati, S.T. Venkataraman
"Design and Implementation of a Robot Control System with Traded and Shared Control Capability". IEEE Conference on Robotics and Automation, Scottsdale, 1989.

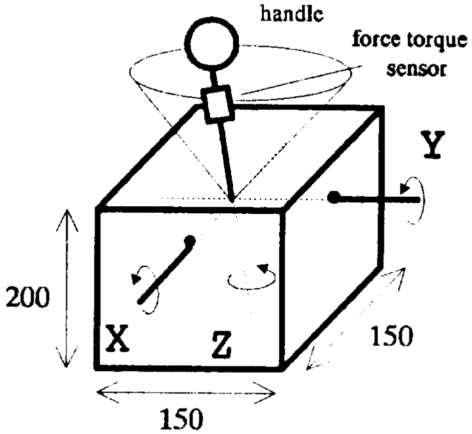


Figure 1

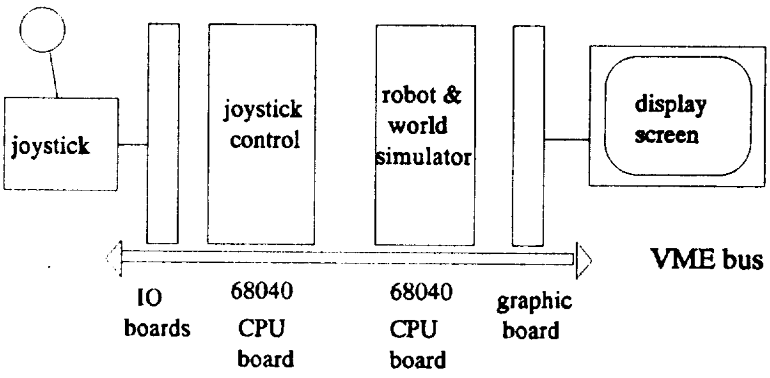


Figure 2

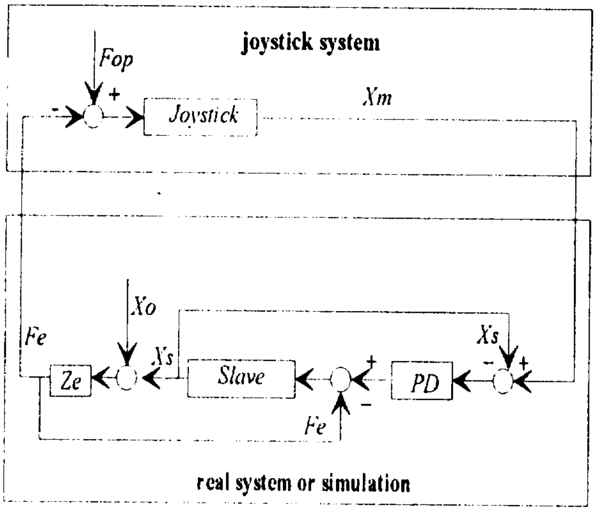


Figure 3

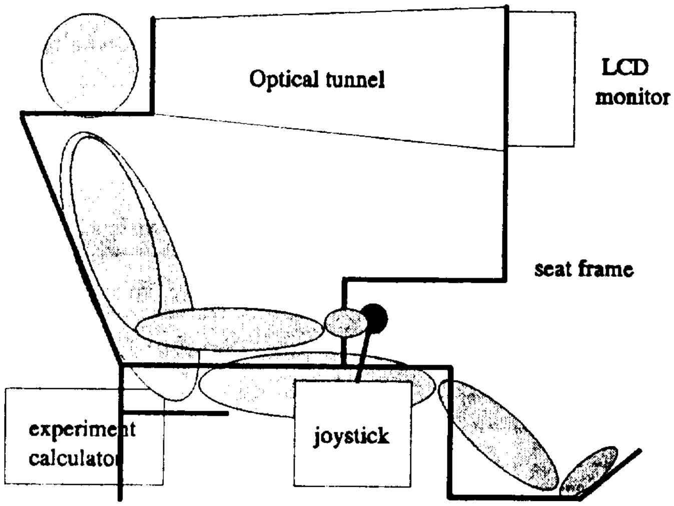


Figure 4

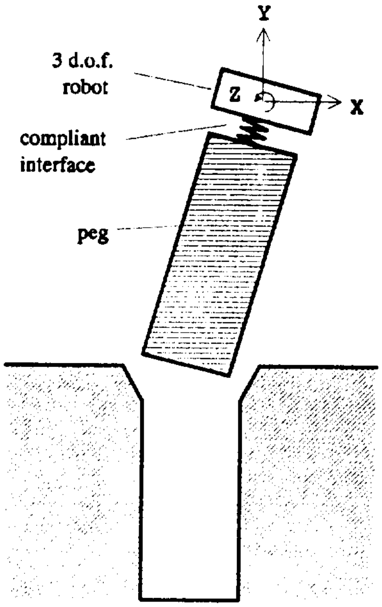


Figure 5

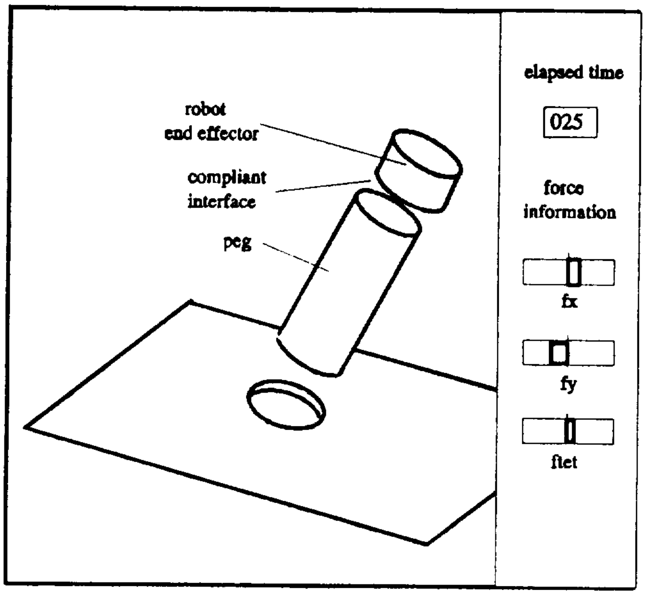


Figure 6

Development of Japanese Experiment Module Remote Manipulator System

N95-23708

Tatsuo Matsueda*, Fumihiro Kuwao, Shoichi Motohashi**, Ryo Okamura****

* : Space Station Department, Tsukuba Space Center
National Space Development Agency of Japan
2-1-1, Sengen, Tsukuba, Ibaraki, 305 JAPAN
TEL : +81-298-52-2919 FAX : +81-298-50-1480
** : Komukai Works, Toshiba Corporation
1, Komukai-Toshiba, Saiwai-ku, Kawasaki, 210 JAPAN
TEL : +81-44-548-5076 FAX : +81-44-541-1211

KEY WORDS AND PHRASES

JEMRMS, Space Robotics, Manipulator, Space Station

INTRODUCTION

National Space Development Agency of Japan (NASDA) is developing the Japanese Experiment Module (JEM), as its contribution to the International Space Station. The JEM consists of the pressurized module (PM), the exposed facility (EF), the experiment logistics module pressurized section (ELM-PS), the experiment logistics module exposed section (ELM-ES) and the Remote Manipulator System (RMS), as shown in Figure 1. The JEMRMS services for the JEM EF, which is a space experiment platform.

The JEMRMS consists of the Main Arm (MA), the Small Fine Arm (SFA) and the RMS console.[1] The MA handles the JEM EF payloads, the SFA and the JEM element, such as ELM-ES. The MA consists of three booms, six joints, a base, an end effector, and two vision equipments, as shown in Figure 2. The two long booms are made of Carbon Fiber Reinforced Plastic (CFRP) Tubes, because CFRP has the low thermal expansion, high stiffness and light-weight characteristics. The short boom is an aluminum tube. The six joints consists of two shoulder joints, one elbow joint, and three wrist joints. Each joint has a brake, which is active without electrical power. Each joint includes the joint electronics unit (JEU), which controls the angle and angular velocity of joint. The base, which consists of titanium, has a curvic coupling to separate the MA from the JEM PM by an EVA crew for maintenance. The end effector is similar to the Standard End Effector of the Space Shuttle. The vision equipment consists of the TV camera and the pun/tilt unit. Only the wrist vision equipment has the arm light. The performance of the MA is Shown in Table 1.

The SFA, which is hold on the tip of the MA during operation, is for dexterous tasks such as the replacement of EF ORUs. The SFA consists of two aluminum booms, six joints, an electronics unit, an end effector, a force/moment sensor and a TV camera. The six joints consists of two shoulder joints, one elbow joint, and three wrist joints. Each joint has a brake, which is active without electrical power. The electronics unit controls the angular velocity of the six joints. The end effector, which is called the tool, grasps the tool fixture and supplies the torque to the bolt. The performance of the

182

SFA is shown in Table 1.

The MA and the SFA are operated from the RMS console by a crew in the JEM PM. The RMS console consists of two TV monitors, passive rotation and translation hand controllers (RHC/THC), a hand controller electronics (HCEL), a remote interface panel (RIP), a laptop workstation (LTWS), a task light, a management data processor (MDP), a mass storage unit (MSU), an arm control unit (ACU), a power distribution box, a rack essential package (REP), a fire detection and suppression (FDS) panel, a hold and release mechanism electronics and a standard double size rack. One split screen capability is provided by NASA, a crew can use three video views simultaneously for the robotics operation. Due to the redesign of the space station, especially termination of the multi-purpose application console (MPAC), we are redesigning the RMS console. The concept design review will be held at the end of October, 1994.

The JEMRMS is currently in the critical design phase. The summary of design, analysis and verification plan of the MA, especially including dynamics and control, is presented.

DYNAMICS

The eigenvalue of the MA is calculated by the Finite Element Model (FEM) in both orbit and launch configurations, as shown in Table 2. The reference configuration is near extended configuration except elbow pitch joint, whose angle is 150 degree to avoid the singularity. The storage configuration is selected to reduce a heater power and to have higher natural frequency than one in the reference configuration. The natural frequencies in orbit configuration are mainly determined by the torsional stiffness of the joint, which is mainly determined by the torsional stiffness of the speed reducer. The current value of the torsional stiffness is derived from the result of the BBM joint test.

In launch configuration, the MA is stowed on the aft end plate of the JEM PM by three hold and release mechanisms (HRM). Before deployment on orbit, the forced relative displacement by the JEM PM deformation due to its pressure is induced large loads in the MA. The load relief in the boom axial direction is installed in the one of HRMs.[2]

CONTROL

The angle and angular velocity of each joint of the MA is controlled by each JEU as shown in Figure 3. The MA is operated in preprogrammed control (primary) and manual control with RHC/THC. For the SFA, the manual control is primary. The controller is designed for not the output axis of the speed reducer but the motor axis to omit the joint backlash in the closed loop. The eight sets of control parameters, which are stored in each JEU, are selected by the ACU depending upon the payload mass property and the arm configuration. The performance of the MA is determined by each joint characteristics, the extensive dynamic analysis with the controller is performed by the non-real time computer simulator.[3] The example of the result, the position error (relative to arm base) in handling the 2300 Kg payload in the preprogrammed control, is shown in Figure 4. The analytical error is smaller than the requirement in Table 1, finally the mathematical model will be updated by the function test using the flat floor.

To maintain the positioning accuracy (relative to target) for misalignments of the JEM due to assembly and thermal distortion, position and orientation of a target can be measured by the wrist vision equipment and the MDP in preprogrammed control.[1]

VERIFICATION

The tests, as shown in Table 3, are planned for the engineering model (EM) of the MA from the summer in 1995 to the summer in 1996. The function test is including the two dimensional flat floor test. The mathematical model of the MA in orbit and launch configuration will be updated by the result of the modal survey and the function test with the flat floor. The static load test and the random vibration are qualification test.

In the end-to-end system test with the MA, the RMS console and the SFA, only the extensive function test including the two dimensional flat floor test of the MA with the SFA is planned.

CONCLUDING REMARKS

The summary of design, analysis and verification plan is presented. The EM of the MA will be manufactured by the spring in 1995. The performance of the MA will be verified by the test successfully.

REFERENCES

[1]Shiraki,K.;Ozawa,K.;Matsueda, T., Goma,K.,Sugasawa,Y.,and Okamura, R., 1993. JEMRMS Development Status. In Proceedings of the Forty fourth Congress of the International Astronautical Federation, IAF-93-T-2-502. Graz, Austria.

[2]Sato,N., Shiraki,K., Kuwao,F., and Niitsu,M., 1994. In Proceedings of the AIAA Dynamics Specialists Conference, AIAA 94-1709, Hilton Head, SC., USA.

[3]Hashimoto,E., Matsueda,T., Kubo,T., and Nishida,S., 1992. Simulation for Developing JEM Remote Manipulator. In the i-SAIRAS 92 Proceedings,.

Table 1. Performance of JEMRMS.

Items		MA	SFA
Maximum payload mass	Kg	7000	300
Maximum tip velocity			
Translation	mm/sec	20^1	10^2
	mm/sec	30^3	30^4
	mm/sec	60^5	
Rotation	deg/sec	0.5^1	0.5^2
	deg/sec	1.0^3	1.0^4
	deg/sec	2.5^5	
Maximum Tip force	N	30	30
Positioning accuracy			
<Relative to	arm base>		
Translation	mm	±50	±10
Rotation	deg	±1.0	±1.0
<Relative to	target>		
Translation	mm	±50	±10
Maximum stopping distance			
Translation	mm	300	75
Rotation	deg	5	5

Note:

1 : less than 7000 Kg payload
2 : less than 300 Kg payload
3 : less than 3000 Kg payload
4 : less than 130 Kg payload
5 : less than 600 Kg payload

Table 2. Eigenvalues of the MA.

Items	Natural Frequency[Hz]	
	1st	2nd
On orbit		
<Reference Config.>		
No payload	0.32	0.34
Payload(500Kg)	0.19	0.21
Payload(7000Kg)	0.046	0.05
<Storage Config.>		
No payload	0.65	1.01
Launch	14.2	14.9

Table 3. Tests for the MA in EM Phase

Function test
Modal survey(Orbit & Launch)
Static load test
Random vibration test(TBD)
EMC
Thermal balance test

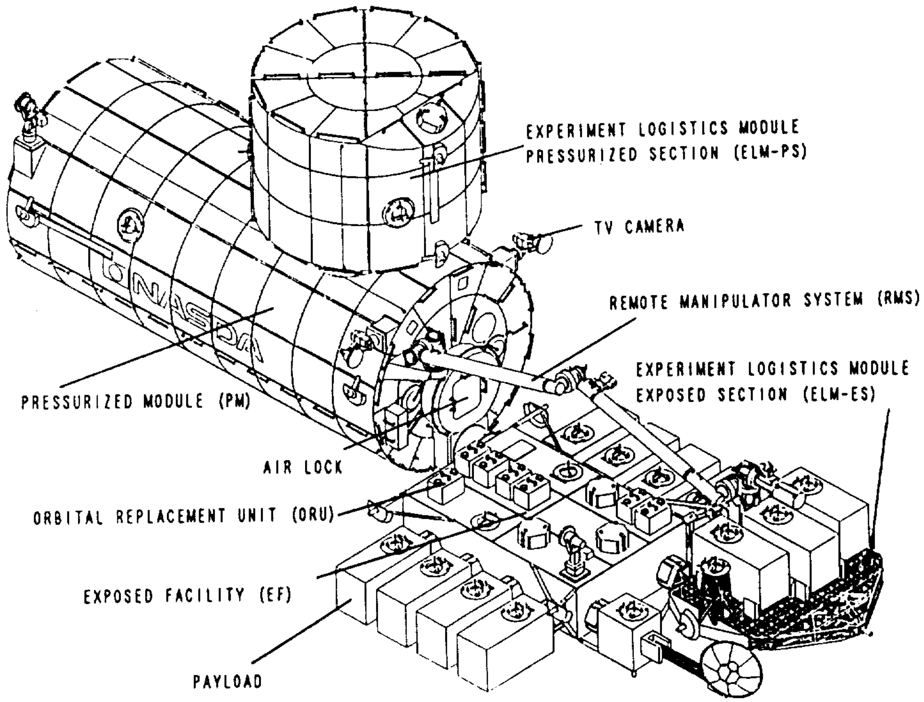


Figure 1. JEM configuration.

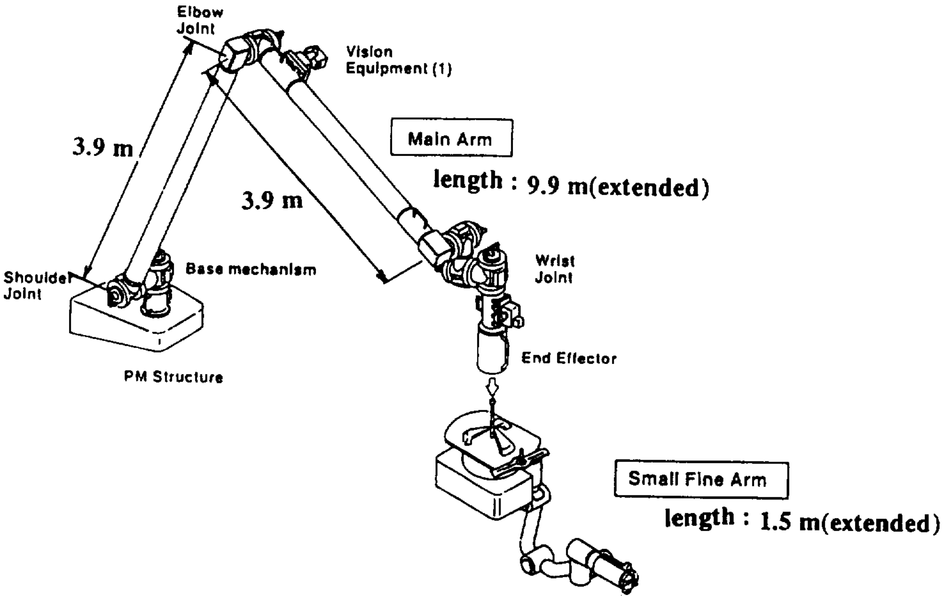


Figure 2. JEMRMS arm configuration.

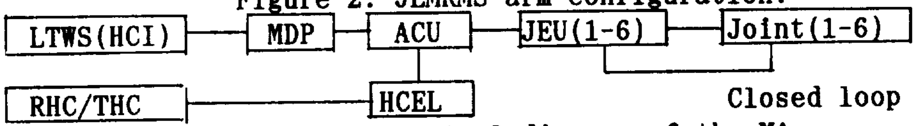


Figure 3. Control diagram of the MA.

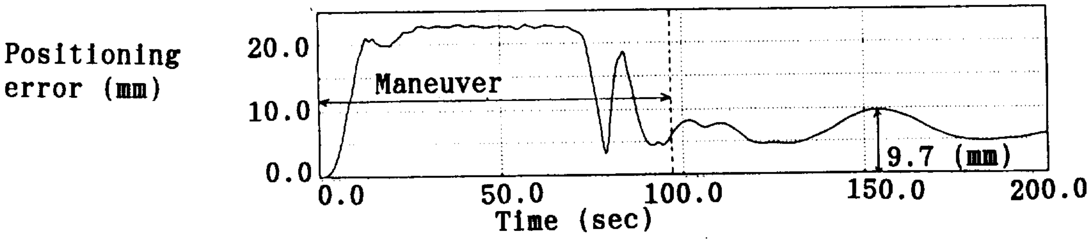


Figure 4. Position error in handling the 2300 Kg payload.

Robot Arm System for Automatic Satellite Capture and Berthing

Shin'ichiro Nishida, Hidetoshi Toriu, Masato Hayashi,
Tomoaki Kubo, Makoto Miyata
Space Systems Engineering Dept., TOSHIBA Komukai Works
1,Komuka-Toshiba-cho, Saiwai-ku, Kawasaki, 210 Japan
Tel: 81-44-548-5135 Fax: 81-44-548-5859

N95- 23709

ABSTRACT

Load control is one of the most important technology for capturing and berthing free flying satellite by a space robot arm, because free flying satellites have difference of motion rate mutually. The performance of active compliance control technique depends on the location of the force sensor and the arm's structural compliance. A compliance control technique with thinking over the robot arm's structural elasticity and a consideration for an end-effector appropriate for it are presented in this paper.

INTRODUCTION

The capture and berthing technique using space robot arm is proven of its effectiveness and its convenience by some space shuttle missions. This technique is also effective for the future unmanned space missions such as repairing, refueling, retrieving or resupplying missions for spacecrafts.

There are two themes for the future capture and berthing technique. One is the unmanned automatic technology. The other is the extension of the allowance for the difference of mutual flying motion rate.

Load applied on both satellites and arm is a key factor to extend the allowance.

This paper shows the consideration for both active and passive load control techniques , especially a joint compliance control and a joint compliance mechanism. Evaluation test result using space robot ground test facility is also mentioned.

SYSTEM ARCHITECTURE

On a capture and berthing mission, the chaser satellite with robot arm approaches to the target satellite and coincides the motion rate to the target satellite using chaser's thruster control. After the relative navigation flying, a TV camera on the robot arm end-effector acquires a view of the target marking near the grapple fixture on the target satellite and the robot arm tracks it by visual feedback control As the target marking, single dot pattern on black back plate is used. Range and direction to the Target satellite can be detected by image processing for the marking image through the arm wrist TV camera.

Detection of the target satellite attitude is no need for capturing. Because, mutual attitude error of the satellites is smaller than the allowable misalignment for the end-effector capturing performance.

On the capturing phase, the chaser's thruster control is shut off and both satellites drift each other in small rate. Impact loads caused on the target satellite capturing is dumped by the arm compliance control and the passive elasticity of the arm.

ARM CONTROL ARCHITECTURE

Load control is one of the most important technology for capturing and berthing free flying satellite by a space robot arm, because free flying satellites have difference of mutual motion rate. On capturing a satellite, the motion energy is transformed to potential energy of the arm distortion and electric energy generated by the joint motors.

The transient force/moment caused on the arm is depends on the elasticity of the arm and its active compliance control. A typical profile of the transient force on satellite capturing is shown in the figure 1. The tip elasticity of an multi-joint arm is changed as the arm portion and the direction.

The performance of active compliance control depends on the location of the force/torque sensor. However much of force controlled robot arm has a 6-DOF force/torque sensor on its wrist, arm structural flexible mode poles are in the force control loop. Single axis model using such wrist-located force/torque sensor is shown in figure 2. The pole cannot easily be cancelled, therefore the bandwidth of the loop is limited at low frequency near the pole.

Joint Compliance Mechanism

One of the solution for getting wider bandwidth of force control loop is to apply the joint torque control loop using a torque sensor on each joint. The control loop configuration is shown in figure 3.

On each joint, joint compliance control is applied. The arm structural flexible mode poles are out of the control loop. Therefore, the loop can be designed for wide bandwidth.

A JCM(Joint passive Compliance Mechanism) is installed on the actuator output shaft of each joint. The JCM elasticity reduces the impact load caused on capturing satellite. And the JCMs dumping characteristics suppress the resonance peek of the arm flexure. The JCM cross-section view is shown in figure 4.

END-EFFECTOR FOR CAPTURE AND BERTHING

For capturing satellite, large allowable misalignment is required of the arm end-effector. Considering the allowable misalignment performance, the end-effector size and mechanism feasibility, two fingers with conical guiding holder type mechanism were selected. Its fingers have suitable elasticity which reduces the impact load on initial contact to the target grapple fixture.

The target grapple fixture is a handle which has rectangular conical outer shape. Figure 5 shows the layout of the end-effector finger mechanism. The outer sleeve moves translationally forward/backward on capturing/ releasing the grapple fixture. The taking back motion on releasing achieves zero rate releasing.

Figure 6 shows the breadboard model of the end-effector. The performance of the end-effector was verified with individual testing and demonstration on the Capture and Berthing Test-bed.

EVALUATION TEST ON TEST-BED

The Capture and Berthing Test-bed is a H/W simulator for capture and berthing mission using robot arm. It consists of a 1.5m length 6-DOF robot arm, its control computer, operation console, television systems, image processor and a 6-DOF satellite motion simulator. These are shown in figure 7.

The satellite motion simulator has orthogonal laid-out 3 actuators, 3-DOF gimbal and a 6-axes force/torque sensor on its tip. The force/torque sensor detects the force/torque applied by the robot arm. The data from the sensor are integrated and used for calculation of the chaser/target sattelites motion difference. The motion differnce is simulated as the table motion.

The operation console has a console computer and a graphic work-station. It is used for predictive simulation display which compensates the signal transmission delay on manual augmented teleoperation.

Demonstration of satellite capturing using the test-bed was successfully completed. On the demonstration, evaluations for automatic satellite capture and berthing task capability, control algorithm andend-effector mechanism are performed. Quantitative variables examined included task performance time and implied load histories on the end-effector. The load histories are shown in figure 8.

Comparison with the manual tele-operated capturing is also studied. Indices of quality for mental work-load and physical discomfort to perform the task manually as a back-up mode were also employed.

CONCLUSION

For automatic satellite capture and berthing, special end-effector and control algorithm should be applied. An end-effector concept and an arm control algorithm were presented in this paper. Testing results using robot arm and motion simulator were also presented.

REFERENCE

[1] M.Oda,Y.Wakabayashi,et.al:ETS-VII,the world first telerobotic satellite In Proc. of i-SAIRAS '92

[2] H.Hashimoto,et.al:Simulation for developing JEM Remote Manipulator In Proc. of i-SAIRAS '92

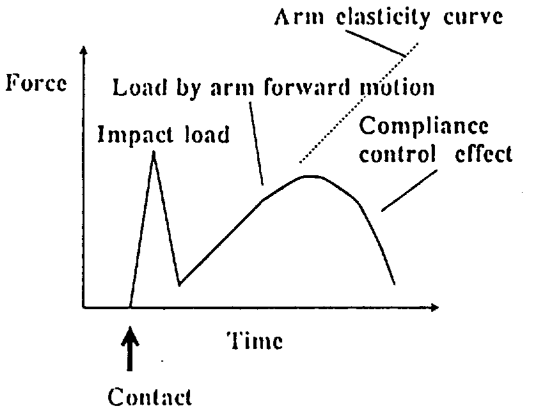


Figure 1. Typical transient force profile on capture and berthing of satellite

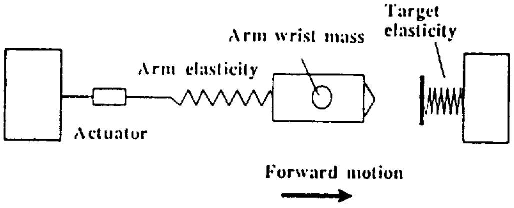


Figure 2. Force controlled arm single axis model

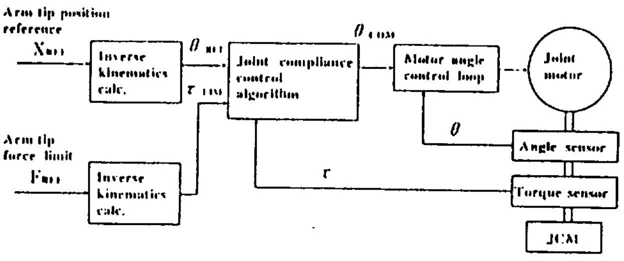


Figure 3. Joint compliance control system model

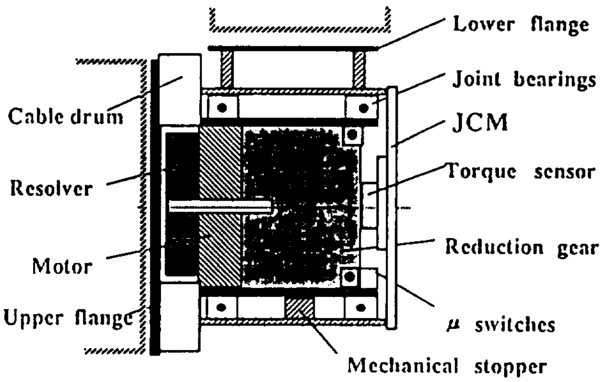


Figure 4. Joint Compliance Mechanism's cross-section

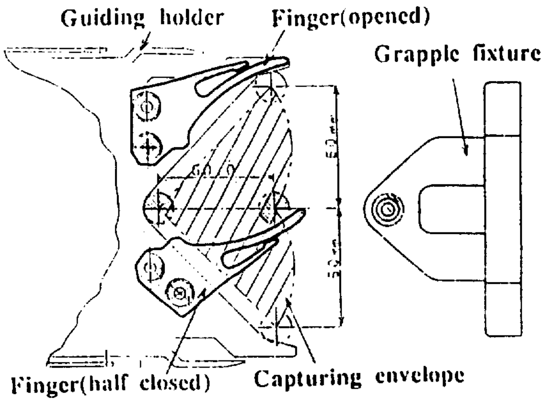


Figure 5. Capture and berthing end-effector mechanism

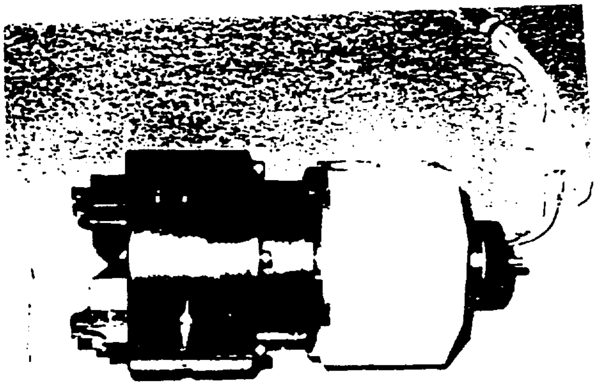


Figure 6. Breadboard model of C&B end-effector

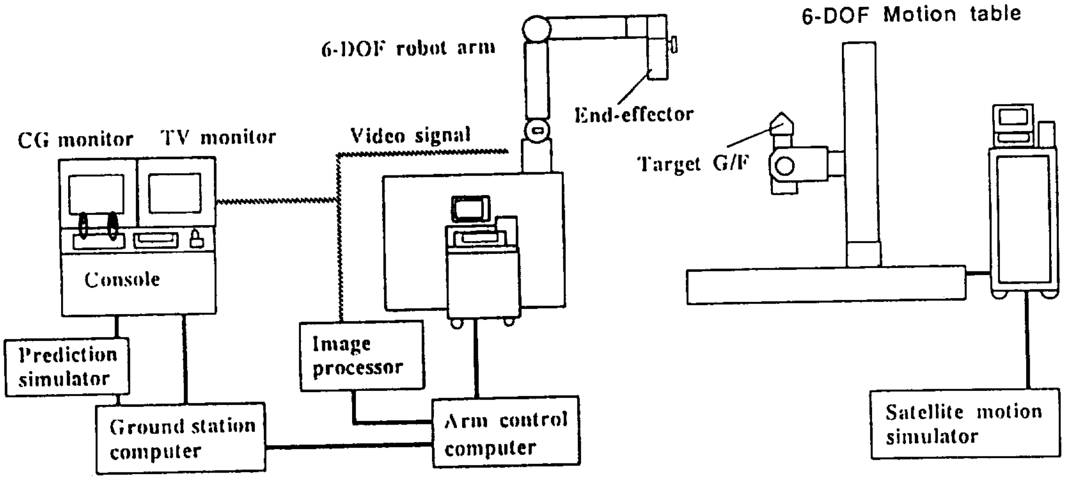


Figure 7. Configuration of C&B Test-bed

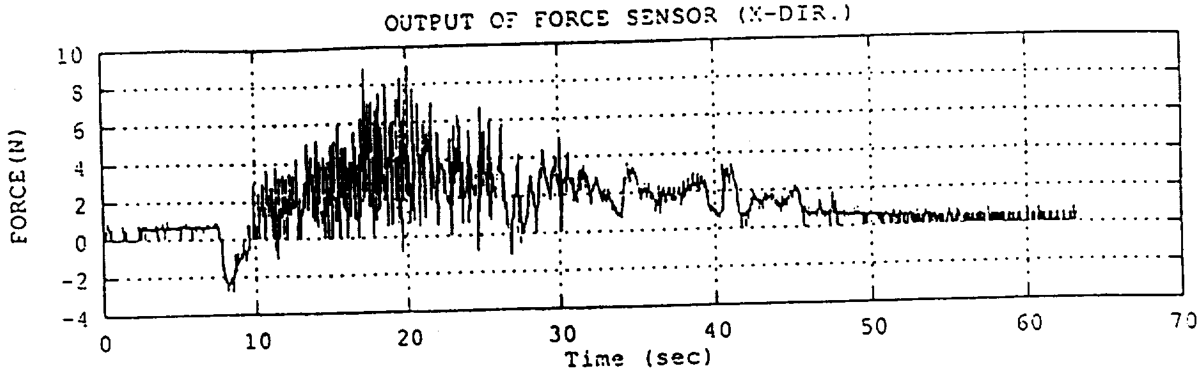


Figure 8. Load history at arm tip on C&B demonstration using C&B Test-bed

Robust Control of Multi-Jointed Robot Arm with a Decentralized Autonomous Control Mechanism

N95- 23710

Shinichi Kimura, Ken Miyazaki and Yoshiaki Suzuki
Communications Research Laboratory,
Ministry of Posts and Telecommunications,
4-2-1, Nukui-Kitamachi, Koganei-shi, Tokyo 184, Japan
TEL (0423)27-7514 FAX (0423)27-6699
E-mail shin@crl.go.jp

KEY WORDS AND PHASES

Decentralized autonomous control mechanism, robust to partial damage.

ABSTRACT

This paper presents a decentralized autonomous control mechanism applied to the control of three dimensional manipulators, and its robustness to partial damage was assessed by computer simulation. Decentralized control structures are believed to be quite robust to time delay between the operator and the target system. A 10-jointed manipulator based on our control mechanism was able to continue its positioning task in three-dimensional space without revision of the control program, even after some of its joints were damaged. These results suggest that this control mechanism can be effectively applied to space telerobots, which are associated with serious time delay, between the operator and the target system, and which cannot be easily repaired after they have been partially damaged.

INTRODUCTION

Teleoperating space robots presents two essential problems for the current control theory because of the long distance between the operator and the target system.

Since the target and the operator are separated by a great distance, not only physically but also within the telecommunication network, there is a serious transmission delay between operational commands and feedback information. The transmission delay in the communication link between a ground station and a space telerobot in low Earth orbit is expected to be as long as 2 to 8 s. System operation is quite difficult with a transmission delay in the order of several seconds. Although some attempts have been made to apply approximation techniques to the control mechanism [1,2], such techniques are limited when the target system is rapidly changed and under uncertain conditions, occurring when the transmission delay is much larger than several hundred milliseconds. Therefore, the space telerobots should be intelligent in order to produce operational information from limited teleoperation commands.

Since the target system is far from its earth base, it will incur high cost and present a danger when repairing space robots after they have been partially damaged. Therefore, telerobots in space should be able to work even when they have been partially damaged. To ensure that the telerobot system can adapt to partial damages, it should have redundant degrees of freedom. On the other hand, space robots are limited in weight, size, and cost limiting the ability to give space robots a redundant degree of freedom. Under normal conditions, space robots that can effectively use this redundant degree of freedom, will be highly adaptable. Even though some efforts have been made to solve automatically the inverse-kinematics of redundant manipulators using a pseudoinverse matrix [3-7] or quadratic programming [8,9], these manipulators will have problems in time performance where the degree of freedom is much larger.

The control mechanisms of motion in living organisms were previously studied [10,11], and it has been proposed that the decentralized autonomous control mechanisms found in biological control systems may be effective in dealing with the problems associated with the teleoperation of space robots. [12] Namely, biological control systems produce rapid and appropriate adaptation according to various external conditions. They are also quite robust in response to partial damage. In these studies it is found that a decentralized autonomous control mechanism is essential for biological adaptation and robustness. Since these control systems are constructed in a hierarchical manner on a foundation of decentralized control, they can adapt to changes in external conditions much more rapidly than if the adaptations had been determined one at a time in the higher center. Furthermore, since the operational information in biological control systems is generated in real time using autonomous control elements according to local and global information in a decentralized manner, the systems are robust to changes in their architecture due to partial damage.

This paper presents a decentralized autonomous control mechanism into a three-dimensional manipulator. It has been successfully demonstrated that the manipulator using the robustness of our control mechanism was able to continue its positioning task in two-dimensional space without revision of the control program, even after 2 (non-adjacent) of its 5 joints were damaged. This paper also describes how this control mechanism can be easily extended to a three-dimensional multi-jointed manipulator.

DECENTRALIZED CONTROL MECHANISM FOR A MULTI-JOINTED MANIPULATOR

Using our theory, the following control mechanism was applied to a two-dimensional 10-jointed manipulator (Fig. 1). The control mechanism proposed here can be easily extended to a three-dimensional manipulator. (1) Each of the joints contains an Elemental Information Processor. (2) The Operator, which corresponds to a remote operator, such as in a control center on Earth, sends information about the target location. (3) Each of the Elemental Information processors calculates a set of possible joint angles based on the two strategies for pointing at the object described below. (Feed forward) (4) The Elemental Information processors exchange these sets of possible joint angles with each other through a Global Information Bus, and select the best set based on a particular cost function. (Consensus-making) (5) Each Elemental Information Processor then applies this Consensus set to itself by sending torque information to its own actuator according to the magnitude of the desired angle change. (Motion) (6) The processes of Feed forward-Consensus-making - Motion are looped until the manipulator points the desired object.

The following benefits can be achieved using this control mechanism: (1) Once the object-related commands are given by the operator, the operational information can be autonomously generated in a decentralized (on-board) system using real-time feed-forward and feed-back information. Therefore, the entire system will be robust to the transmission delay between the target system and the operator. (2) Since all of the elemental processors calculate their own strategies in parallel, and exchange this information with each other, if some of these elements are damaged, the manipulator will still be able to point at the object employing the strategy generated using the undamaged elements (particularly using the following Non-Redundant Strategy).

The 2 two-dimensional strategies can simply be extended to a three-dimensional system in the following manner. [12] The "Equally Shared" strategy, which is based on constraints that depend upon the number of joint angles that can bend simultaneously, can be applied to a three-dimensional system by supposing that the base joint and the objective lie within the same imaginary plane. Namely the needed angle is equally shared by the pitch and/or yaw angles of joints on the plane which is parallel to the pitch or yaw axis.

For the "Non-Redundant" strategy, the non-redundant degrees of freedom are simply extended from two to three. Namely one pitch and two yaw angles, or one yaw and two pitch angles are adjusted to solve inverse kinematics as if it were nonredundant manipulator.

Cost Functions for Selecting from Among Possible Joint Angles

Various cost functions can be used to select a set of joint angles from among the possible solutions calculated using the previous strategies, according to desired performances of the manipulator. For example, if the manipulator is to move in the most energy-efficient manner, a cost function is used that measures energy consumption, which may correspond to the sum of the changes of all of the joint angles.

A simple cost function that selects the set in which the maximum change of all of the joint angles is less than those in the other sets. This cost function is believed to select the solution which enables the manipulator to point at the target location in the shortest amount of time.

To prevent the system from attempting a particular set of joint angles that cannot be achieved because of damage, a valuation scheme is introduced. In this scheme, a set of joint angles is considered invalid if the torque values of all of the joint angles are less than a certain value and the manipulator cannot access the target location.

In the next report, a comparative study of this cost function will be done.

RESULTS FROM COMPUTER SIMULATION

Computer simulation was used to verify the merits of our decentralized autonomous control of a 10-jointed manipulator. The arm lengths between the joints are identical.

Accessing Path According to Initial Condition

A cost function was selected which will achieve the fastest positioning. Therefore, the time required to point at each target location was expected to be optimized under various conditions. As shown in Fig. 1, the manipulator changed its accessing path according to its initial position. This result indicates that decentralized control mechanism effectively uses a redundant system parameter for time performance. In this decentralized control

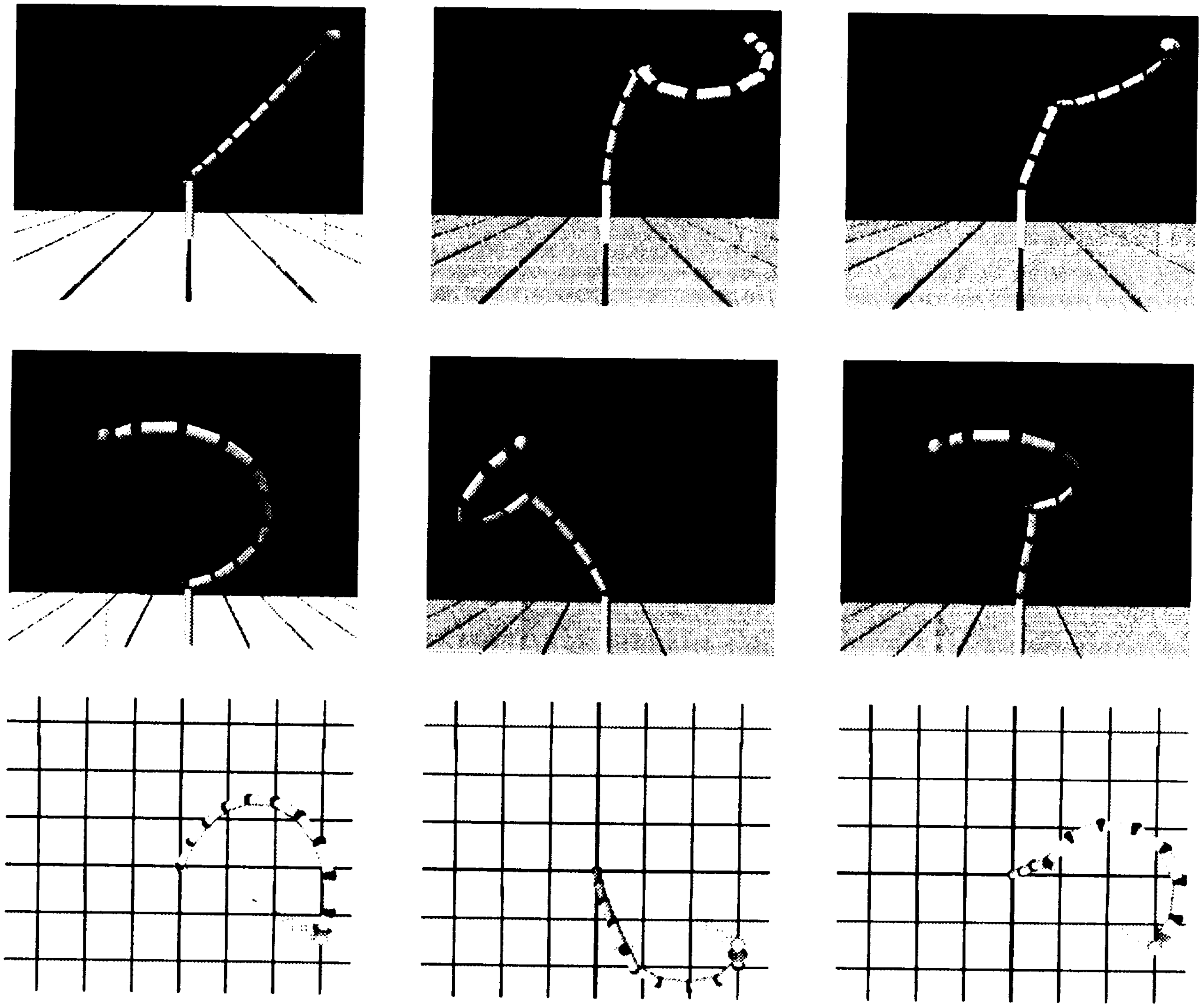


Fig. 1: Kinematic views of the access path of a 10-jointed manipulator based on its initial position. The target location is all identical.

mechanism, the number of joints hardly affects the control performance, since calculations are performed in local processors. Therefore, this result suggests that, under this control mechanism, as the number of joints in the manipulator is increased, the manipulator becomes more dexterous and faster, if permitted by the constraints of the hardware and the communication within the Global Information Bus.

Robustness to Partial Damage

The robustness of this redundant manipulator to partial damage is assessed by fixing one or two joints at a certain angle. Even after one of the ten joints was frozen at a certain angle while the manipulator was accessing the target location, the manipulator successfully pointed at the target location by autonomously changing its strategy without any external assistance or additional information. Figure 2 shows how this manipulator accesses to the target after the fifth joint was frozen when the manipulator was at its initial position. The percent of the dead area is not increased when one of the ten joints is frozen at an angle of 0°. These results show that the manipulator can solve the inverse-kinematics of every location it can physically. As shown in Figure 3, this manipulator can autonomously adapt to the condition when some joints are simultaneously damaged. (The other conditions are the same as Figure 2.) These results show that the manipulator based on our control mechanism has high adaptability to its partial damage.

DISCUSSION

This section discusses how to design a control system based on our control mechanism from a generalized viewpoint. The designing principle can be summarized using the following five points.

The first is to give "local processors" to each element of the system. The "local processor" discussed in our scheme can be imaginarily achieved using a single processor employing a program module. Of course it is better for performance and robustness if a small and independent processor for each joint is used to construct a control system.

The second is to provide an information path so that these local processors can communicate with each other. If an independent processor is used for each joint, an information path must be given between local processors. Especially for space robots, wireless communication will be advantageous since wireless communication is less sensitive to structural constraints and can broadcast information.

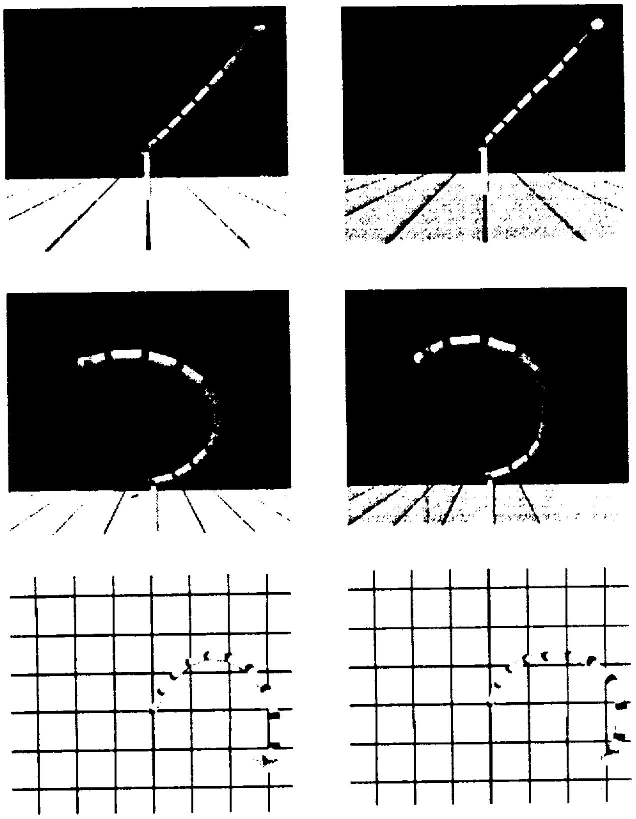


Fig. 2: Kinematic views of the access path after the fifth joint was frozen.

The third is to have algorithms show how the local processors calculate an operational-information candidate independently of one another (Strategy 1 and 2, in the case of this manipulator).

The fourth is a cost function regarding how to select the operational information from the candidates (a cost function which selects the set in which the maximum change of all of the joint angles is less than those in the other sets, as in the case of this manipulator).

Points (3) and (4) are the most important points for designing a control system. These algorithms should be created according to the function of plant. For example the cost function that selects a candidate to minimize energy consumption may suit one plant, another cost function that selects the candidate which

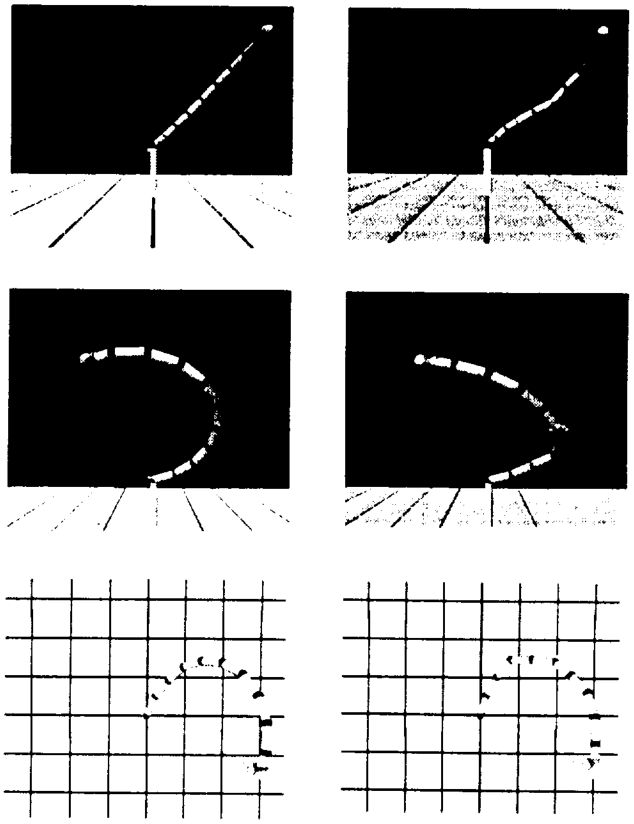


Fig. 3: Kinematic views of the access path after the third, fifth, seventh and tenth joints were simultaneously frozen.

optimizes fluency may suit another plant. The system function should be implemented into algorithms from the viewpoint of the local processor.

Point five is to give time constant for the calculation-process calculation process loop. In this control mechanism, it is essential to loop locally and globally real time. If the time constant is small, precise control is attainable and if the time constant is large, control systems have good time performance. Therefore, optimum time constants should be given according to the function of the plant.

According to the generalization of the designing principle of control systems, our control system can be extended for various plants, and an adaptive decentralized autonomous control system can be achieved.

CONCLUSION

This paper presents our decentralized autonomous control mechanism applied to the control of three-dimensional manipulators and its robustness to partial damage was assessed by computer simulation. Decentralized control structures are believed to be quite robust to time delays between the operator and the target system. A 10-jointed manipulator based on our control mechanism was able to continue its positioning task in three-dimensional space without revision of the control program, even after some of its joints were damaged. These results suggest that this control mechanism can be effectively applied to space telerobots, which are associated with serious time delays between the operator and the target system, and which cannot be easily repaired after they have been partially damaged.

REFERENCES

[1] Vemuri B. C.; and Skofteland G., 1990, Motion Estimation from Multi-Sensor Data for Tele-Robotics, In *IEEE Workshop on Intelligent Motion Control*, 20-22.

[2] Hirzinger G.; Heindl J.; and Landzettel K., 1989, Predictive and Knowledge-Based Telerobotic Control Concepts, In *IEEE Conference on Robotics and Automation*, 14-19.

[3] Liegeois A., 1977. Automatic Supervisory Control for the Configuration and Behavior of Multibody Mechanisms, *IEEE Trans. Sys. Man Cyber.* 7:868-871.

[4] Hollerbach J. J.; and Suh K. C., 1985, Redundancy Resolution of Manipulators through Torque Optimization, In *Proc. 1985 IEEE Int. Conf. Robotics Automation.*, 1016-1022.

[5] Salisbury J. K.; and Abramowitz J. D., 1985, Design and Control of a Redundant Mechanism for Small Motion, In *Proc. IEEE Int. Conf. Robotics Automation*, 323-328.

[6] Maciejewski A. A.; and Klein C. A., 1985, Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, *Int. J. Robotics Res.* 4,: 109-117.

[7] Kelin C. A.; and Blaho B. E., 1987, Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators, *Int. J. Robotics Res.* 6: 72-83.

[8] Cheng F. T.; and Orin D. E., 1990, Efficient Algorithm for Optimal Force Distribution - The Compact-Dual LP Method, *IEEE Trans. Robotics Automat.* 6: 178-187.

[9] Cheng F. T.; Chen T. H.; and Sun Y. Y., 1994, Resolving Manipulator Redundancy under Inequality Constraints, *IEEE Trans. Robotics Automat.* 10: 65-71.

[10] Kimura S.; Yano M.; and Shimizu H., 1993, A Self-Organizing Model of Walking Patterns of Insect, *Biol. Cybern.* 69: 183-193.

[11] Kimura S.; Yano M.; and Shimizu H., 1994, A Self-Organizing Model of Walking Patterns of Insect, (II) The Loading Effect and Leg Amputation, *Biol. Cybern.* 70: 505-512.

[12] Kimura S.; Miyazaki K.; and Suzuki Y., 1994, Application of a Decentralized Autonomous Control Mechanism for Space Robotics, In *Proceedings of 19th International Symposium on Space Technology and Science*, in press, Yokohama, Japan.

N95- 23711

A Development of the Dynamic Motion Simulator of 3D Micro-Gravity with a Combined Passive/Active Suspension System

Kazuya Yoshida * Shigeo Hirose Tadashi Ogawa

Dept. of Mechano-Aerospace Engineering
Tokyo Institute of Technology
2-12-1, O-okayama, Meguro
Tokyo 152, JAPAN
E-mail: *yoshida@mes.titech.ac.jp*

*Currently with Dept. of Mechanical Engineering
Massachusetts Institute of Technology

KEY WORDS

Micro-Gravity Testbed, Hybrid Simulation, Pendulum System, Free Body Dynamics, Collision Dynamics

BACKGROUND AND GOAL OF THE PAPER

The establishment of those in-orbit operations like "Rendez-Vous/Docking" and "Manipulator Berthing" with the assistance of robotics or autonomus control technology, is essential for the near future space programs. In order to study the control methods, to develop the flight models, and to verify how the system works, we need a tool or a testbed which enable us to mechanically simulate the micro-gravity environment; but it's not that easy.

There are a lot of attempts to develop the micro-gravity testbeds, but once the simulation goes into the docking and berthing operation that involves mechanical contacts among multi bodies, the requirement becomes suddenly critical. The testbed must move in 3D space with a very high frequency response to follow the impact or collision, which class of motion is very difficult to be modeled and numerically simulated by a computer, the hardware simulation or experiment by a testbed therefore is the only way to study.

A group of the Tokyo Institute of Technology has proposed a method that can simulate the 3D micro-gravity producing graceful, smooth respose to the impact phenomena with relatively simple apparatus. Recently the group has carried out basic experiments successfully using a prototype hardware model of the proposing testbed.

This paper will present our idea of the 3D micro-gravity simulator and report the results of our initial experiments.

MICRO-GRAVITY TESTBED OVERVIEW

In order to study real mechanical dynamics and demonstrate the practical validity and effectiveness of a control system using actual sensors, computers and mechanical assemblies, we need experiments with a laboratory model. However, reproducing the micro-gravity environment is not an easy task because we cannot obtain natural 3D zero-gravity or perpetual free-falling environment on earth. In general, the following methods could be available for emulating psuedo-zero-gravity:

1. Do experiments either in an airplane flying along a parabolic trajectory or a free-falling capsule. In this case, we can observe the pure nature of micro-gravity, but the cost of such experiments are very

high. In addition, this environment is very inconvenient and accommodates only very short duration experiments.

2. Do experiments in a water pool with the support of neutral buoyancy. This is especially good for the training of astronauts' activities, but from a micro-gravity dynamics point of view, water current and drag forces disturb the dynamic motion.

3. Suspend an experimental model by tethers to cancel the vertical gravitational motion. In case active counter-balancing is employed, the design of a quick response, vibration free and simple suspension control must be a key issue.

4. Support an experimental model by air-cushions or air-bearings. This is the simplest method; however, the motion is restricted to a horizontal plane.

5. Calculate the motion which should appear in zero-gravity environment based on a mathematical model, then force the corresponding mechanical model to move according to the calculation. This method is called as a '*hybrid*' simulation, a combination of mechanical and mathematical models. A testbed develped at the MIT comprising a 6DOF Stewart platform and a PUMA manipulator is classified into this category [1]. In the system, the platform provides base vehicle motion in a simulated micro-gravity dynamics based on the reaction torque sensing between the PUMA and the platform. This class of method is useful especially for 3D kinematic motion, but for the dynamic simulation, the computation and servo-control bandwidth becomes critical.

A group of the TIT has developed the air-bearing type of testbed named EFFORTS (Experimental Free-FlOating RoboT Satellite simulator) and got excellent experimental results for many years [2][3]. The advantage of this testbed is that we can observe the nature of mechanical system in a frictionless floating environment, but the drawback is the limitation to 2D motion.

For the 3D simulation, we decided the tethered hanging system, but paying attention that we should keep the advantage to observe the nature of pure dynamics; the solution to this requirement is to combine a passive mechanical system and an active control system. The detail shall be presented in the following section.

OUR PROPOSING 3D MICRO-GRAVITY SIMULATOR

Basic Principle

Imagine that a body is suspended by a spring, like **Fig.1**. If the spring is very long, we have a very small (close to zero) pendulum force for a small horizontal displacement.

$$\mathcal{F}_x = mg\frac{\Delta x}{\ell} \cong 0 \qquad : \text{if } \ell \text{ is large.}$$

And if the spring has a very low stiffness, the spring force is almost constant (just equals to the gravity force) for a small vertical displacement around the equilibrium point.

$$\mathcal{F}_z = mg\cos\theta + k\Delta z \cong mg \qquad : \text{if } \theta, k \text{ small.}$$

This simple mechanical system will accommodate a passive suspension to cancel the gravity with almost zero disturbance for small displacements to all directions.

If the displacement of the body reaches not a small amount, we should actively move the top of the spring to follow the body motion, however this active motion is not necessary so fast, just to follow and absorb vibration.

In conclusion, the proposing suspension system should be composed by a long and compliant spring and an active tracking mechanism to move the top of the spring. The passive spring will be transparent and give almost no disturbance to the high frequency force or impact dynamics of the body, and simultaneously, the low frequency gross motion is followed by the active tracker at the top.

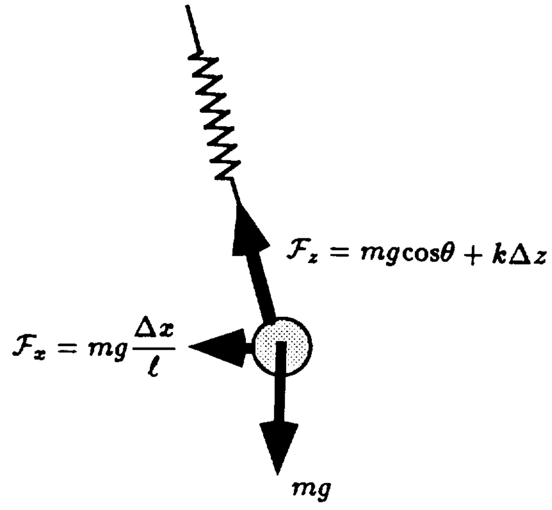


Fig.1 Basic principle of the suspension system

Combining a passive mechanism into the system, we can relieve the servo-controller from the requirement of high frequency response, which point is always a critical key for "hybrid" type of micro-gravity simulators. This is the advantage of the proposing Combined Passive/Active suspension system.

Hardware Design and Control

The developed prototype system is shown in **Fig.2**.

To provide the 3D (x-y-z) active motion of the top of spring, though any types of tracker will be available, we employed a three-tether system for the convenient installation and lower cost per working area than a Cartesian-type linear motion table.

In the figure, points A, B, C, and P form a pyramid with a regular triangle base ABC on the horizontal ceiling. Controlling wire lengths AP, BP and CP with each reel over the ceiling, we can arbitrary position P in the Cartesian 3D space.

The micro-gravity simulated object (a plate in the figure) is supported by a passive gimbal system to allow the natural rotation, and hooked by a long and compliant spring with the driving system at the top.

The motion of the object (plate) is measured by a real-time 3D vision analysis system (named "Quick Mag" developed by OKK Inc, Japan), and when it detects the displacement of point Q, the driving system translates the point P exactly above Q in the distance of the equilibrium spring length.

From a control system point of view, this is a non-collocate flexible system. However, our control goal is not fine positioning of the tip, but gross motion tracking and vibration absorption. Simply speaking, if the control system responses smoothly and quickly in higher frequency than the nature of the spring-pendulum system, the simulation system works. The long and low stiffness spring suspension contribute, again, to make the mechanical natural frequency lower, henceforth the control system implementation easier.

EXPERIMENTS AND CONCLUSION

The developed system works very well to follow the natural micro-gravity motion of the supported body. We have carried out collision experiments that the body hits an fixed wall, or two supported bodies collide each others. We measured the whole sequence of the collision, then estimated the micro-gravity environment that the system can accommodate.

The mass below the spring is $m = 0.3$ [kg] and the spring compliance about $k = 0.3$ [kg/m], then the natural frquency of the spring-mass system for vertical vibration ω_v is

$$\omega_v = \sqrt{\frac{k}{m}} \cong 1.0 \text{ [Hz]}.$$

The total length of the spring about $\ell = 2.0$ [m], then the natural frquency as pendulum system for holizontal swing ω_h is

$$\omega_h = \sqrt{\frac{\ell}{g}} \cong 0.45 \text{ [Hz]}.$$

Comparing with these frequencies, the servo-controller frequency for the driving system is much faster (higher than 20 [Hz]), the system then follows the object motion smoothly with undesirable vibration or swing well damped.

As the result of the collision experiments (where a metal ball supported by the developed combined active/passive system hits an aluminum plate supported by the passive hanging system with a gimbal. See **Fig.3**), we identified the disturbance acceleration of the developed mechanical simulator as about 0.01G when the object moves 0.1 [m/s] and 0.1G when the object moves 1.0 [m/s], the loss of momentum among the bodies through the collision is just 3.3% in average of more than ten times experiments.

In this paper, we proposed a new type of 3D micro-gravity testbed with a combined passive/active suspension system. The key was the introduction of a very compliant and low frequency mechanically passive part into the testbed. Such a passive suspension accommodates very small disturbance for the higher frequency impact dynamics, as well as relieves the active gross-motion tracking system from the high frequency response requirement.

Although the developed prototype testbed was very primitive, it worked relatively well showing good performance especially for the study of free body collision dynamics.

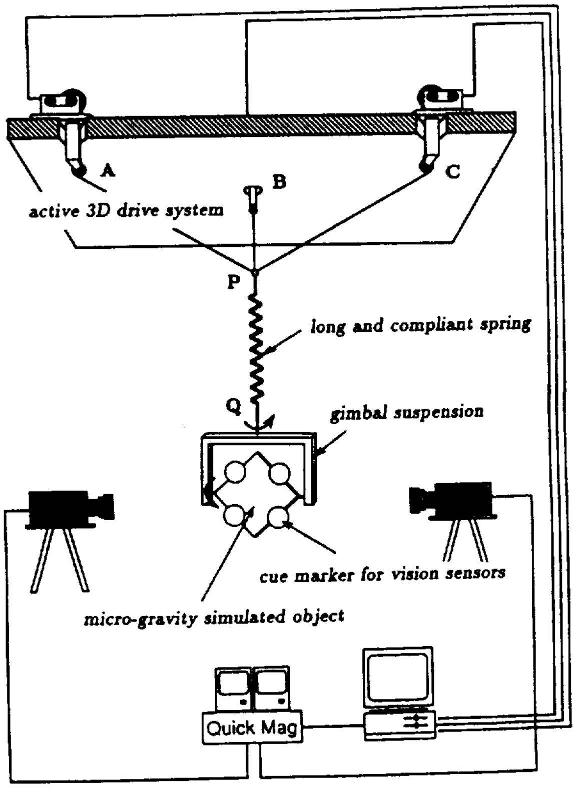


Fig.2 The developed Dynamic 3D Motion Simulator with a Combined Passive and Active Suspension System

References

[1] S.Dubowsky et al; "A Laboratory Test Bed for Space Robotics: The VES II," *Proc. of IEEE/RSJ Int. Symp. on Intelligent Robot System (IROS'94)*, Munich, Germany, 1994 (to appear).

[2] Y.Umetani and K.Yoshida: "Experimental Study on Two Dimensional Free-Flying Robot Satellite Model," *Proc. of NASA Conf. on Space Telerobotics*, vol.5, pp.215-224, Pasadena, CA, 1989.

[3] K.Yoshida: "Space Robotics Research Activity with Experimental Free-Floating Robot Satellite (EFFORTS) Simulators" *Proc. of 3rd Int. Symp. on Experimental Robotics*, Kyoto, Japan, 1993 (to be published).

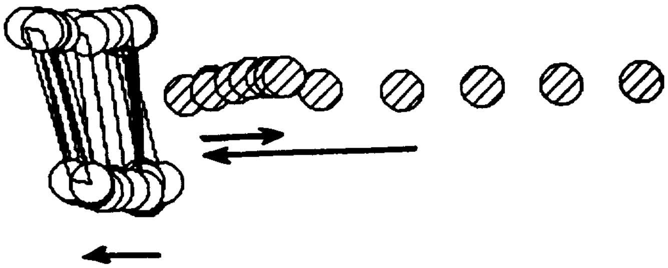


Fig.3 An experiment of collision between the suspended plate and ball

Small Image Laser Range Finder for Planetary Rover

N95-23712

Yasufumi WAKABAYASHI,Masahisa HONDA
Tsukuba Space Center,National Space Development Agency of Japan
2-1-1,Sengen,Tsukuba-shi,Ibaraki-ken 305,Japan
Tel 0298-52-2241 Fax 0298-52-2247
Tadashi ADACHI,Takahiko IIJIMA
Aerospace Division,Nissan Motor Co.,Ltd.
21-1,Matoba-Shinmachi,Kawagoe-shi,Saitama-ken 350-11,Japan
Tel 0492-31-1113 Fax 0492-31-1116

Abstract

Quite a few tasks remain to solve a variety of technical subjects for planetary rover navigation in the future missions. The sensors to perceive the terrain environment around the rover will require critical development efforts. The image laser range finder (ILRF) discussed here is one of candidate sensors because of many advantages to directly provide range data to be required for its navigation.

The authors developed a new compact-sized ILRF which has a quarter in volume size of those conventional ones. Instead of current two directional scanning system comprised of nodding and polygon mirrors, the new ILRF is equipped with a new concept of direct polygon mirror driving system, which successfully made its size compact to accommodate to the design requirements. The paper reports design concept and preliminary technical specifications established in the current development phase.

1. Introduction

The onboard sensors for Lunar or Mars rovers which will be planned in the future missions are one of the most critical elements to sense the terrain environment around the rovers. A stereo type three dimensional sensor and ILRF are most possible for the navigation system. The stereo-type three dimensional sensor system generates three dimensional terrain data from the image of onboard CCD cameras. The CCD cameras provide many advantages in various design aspects to build up the onboard sensor system. But this system needs to manage many data of the image to percept three dimensional terrain feature, so high performance computer should be required. While, the ILRF is with excellent capability to obtain three dimensional terrain geometric data within a short period of one second, which will not be influenced by the surface condition. The currently developed ILRF is equipped with a mechanical scanner for laser beam in two dimensional directions, therefore, further development efforts must be devoted to improve its design features such as reliability, weight, size, power consumption to be onboard space hardware. In practical missions, the ILRF may be utilized in cooperation with the stereo system.

2 System Outline

The ILRF sensor onboard the rover will be utilized to collect the terrain information in front of the rover, which will require high accuracy and high frame rate performance. This requirements lead us to the conclusion to a methodology of phase-comparison system utilizing intensity-modulated CW laser. The concept of this methodology is shown in Figure 1.
The laser beams are radiated to the objects, which will reflect the beams with some phase shift proportional to distance up to a radiated point. And then the resultant amount of phase

shift shall be obtained to compare with the reference to determine the distance between the rover and targeting points.

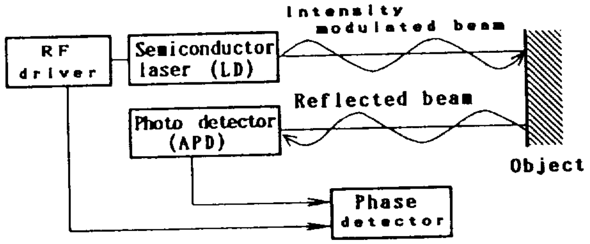


Figure 1 Concept of range measurement

Thus optical scanning in two dimensional directions helps to obtain three dimensional terrain geometric data. The optical system consists of a polygon mirror and two collecting mirrors. The object of our efforts is to realize the compactness without degrading measuring performance. The resultant concept of the optical system is shown in Figure 2.

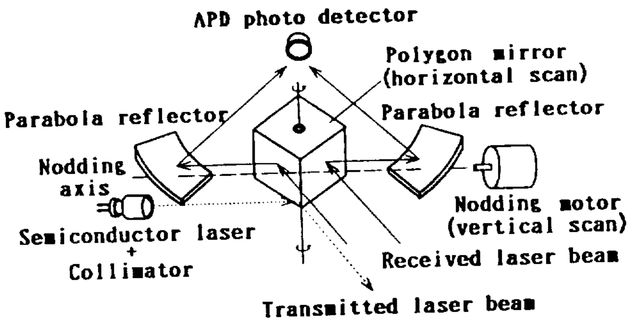


Figure 2 Concept of the optical system

Downsizing effort on polygon mirror was made by means of receiving the beam at the two facets of the polygon mirror (with four facets) for horizontal scanning instead of one facet and the nodding mirror which was eliminated by directly driving the polygon mirror in vertical direction. An incident laser beam is split into two directions by the two facets of polygon mirror, and each beam is directly focused on a detector by parabola reflectors set up symmetrically. Thus, the collecting lenses can be eliminated, which enable to minimize the optical system as a design feature. The standard sensing circuit is used in our ILRF as shown in Figure 3. The intensity of the semiconductor laser is modulated in a 10.7MHz frequency, the reflected beam from the object received by an APD detector. The output signal received by the detector passes into the detection circuit of range and reflectance. The range is determined by the phase difference of measuring signal and referential signal. In order to obtain sufficient sensitivity and accuracy, nearly 100 waves are respectively integrated for each range data. The data of range and reflectance are converted into digital data by an A/D converter. All of these data are sent to a signal processor for the terrain perception.

3. Performance improvement

The conventional laser range finders are still with problems with respect to its performance. The efforts in improving its performance were devoted to sensor as follows.

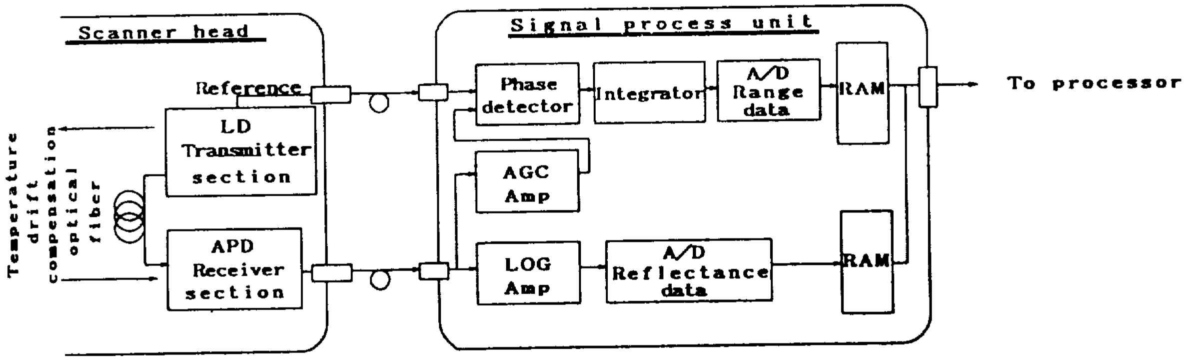


Figure 3 Block diagram of sensing circuit

(1) Protection of internal reflection of laser beam

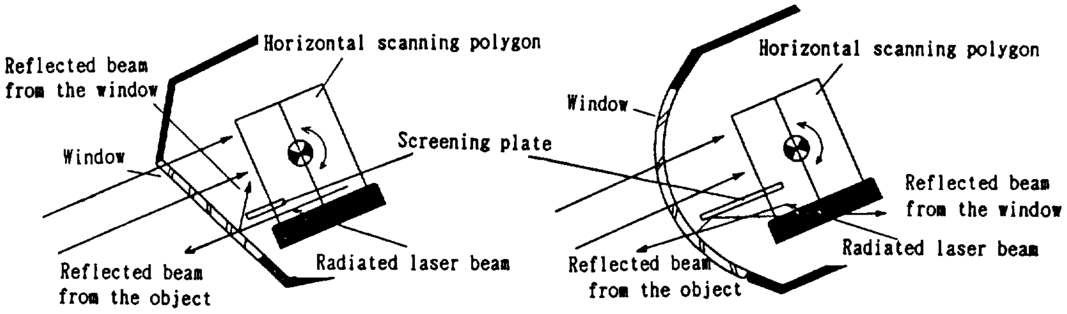
The transmitted laser beams in some directions are reflected at the window as shown in Figure 4(a) and these make the ghost in the range and reflectance images. Essentially, it is difficult to eliminate this reflection since the reflected laser beam at the window is larger than return signal from the object. We set the cylindrical window with the same surface curvature center with the pitch gimbal pivot for vertical scanning, and a complete screening plate as shown in Figure 4(b) can be set to split the laser beam transmitting part from the receiving optics for eliminating the ghost image.

(2) Compensation of range drift caused by temperature changes

The measured ranges might be mostly unstable due to changes of environment temperature with respect to a conventional laser range finder. The newly developed ILRF was no exceptional for this drift phenomena. In order to compensate the possible drift, the laser beam is led to the APD detector through a optical fiber cable with a reference length in no outside sensing period. This enables to measure drift rate in reference to the predetermined length and then, the compensation for correcting the data is executed with respect to all measurement range data. The schematic diagram of processing is illustrated in Figure 5.

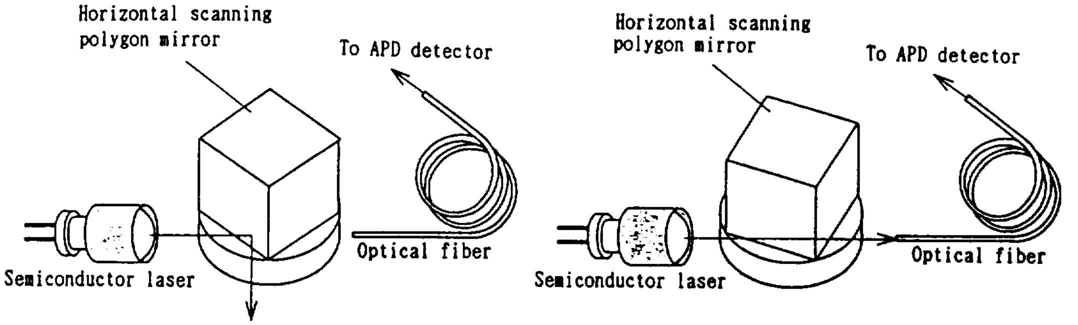
4. Characteristics of ILRF

The current ILRF configuration is given in Figure 6, and the characteristics in Table 1. Figure 7 shows ranging and reflectance image to be obtained through the ILRF sensor.



(a) Before improvement (b) After improvement

Figure 4 Protection concept for laser beam inner refection



(a) Laser is transmitted to object (b) Laser is passed through optical fiber

Figure 5 Compensation method of the range drift by temperature change

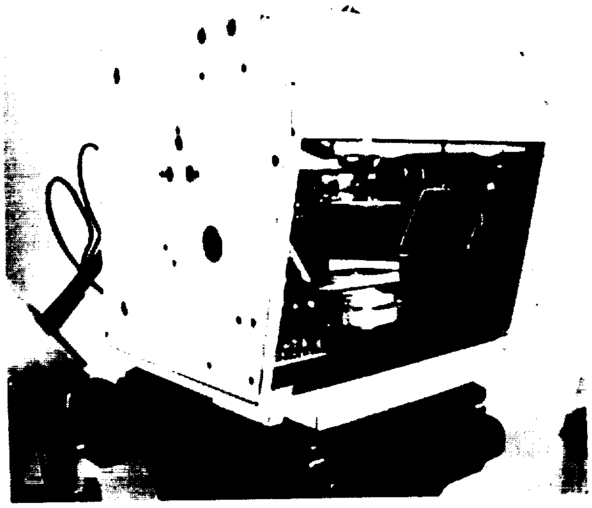


Figure 6 Configuration of the ILRF

Table 1 Characteristics of ILRF

Data details	2D Reflectance Image 2D Range Image
Field of view	80° (horizontal)×40° (vertical)
Space resolution	256(horizontal)×64(vertical)
Range	1.5m~14m
Modulation frequency	10.7MHz
Frame interval	1 sec
Size of scanning head	150×150×150 mm^3
Weight of scanner head	about 3kg
Range resolution	12Bit
Intensity resolution	12Bit
Laser power	60mW
Consumption power	80W (24V Input)

5. Conclusion

The image laser range finder reported in this paper still require improvement in order to secure more sufficient reliability especially for a mechanism of the scanner. However, the authors have successfully achieved to develop a more compact-sized laser range finder with satisfactory function and performance in the current development phase. The state-of-the-art proven in the report is believed to enhance the design-in and design-out efforts for more practical terrain sensor in the very near future.

Our ILRF has been developed in cooperation with Corporate Research Division Olympus Optical Co.,LTD.

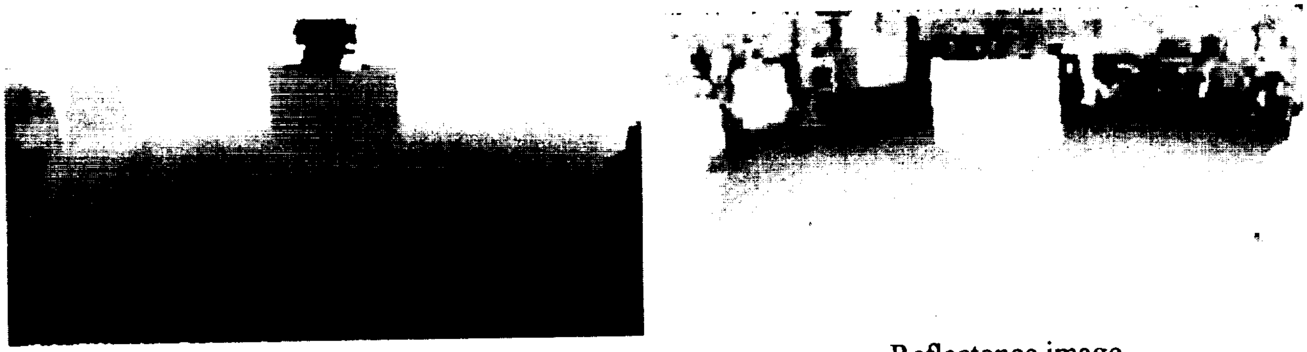
Reference

[1] Max E.Bair et al.,"Three-dimensional image and applications" SPIE Vol.726 Intelligent Robot and Computer Vision: Fifth in a Series PP264-274, 1986.

[2] In So Kweon et al., "Experimental Characterization of the Perception Laser Rangefinder" Technical Report CMU-RI-TR-91-1 Robotics Institute, Carnegie Mellon University January 1991.

[3] Lu Zukong et al.,"Range Imaging for Auto-Vehicle Guidance Applying an optical Radar" SPIE Vol.1831 Mobile Robots VU PP456-456,1992.

[4] J.Richard Kerr et al.,"Real time imaging rangefinder for autonomous land vehicle" SPIE Vol.1007 Mobile Robots III PP349-356 1988.



Ranging image

Reflectance image

Figure 7 Range and reflectance image pair of ILRF

Automatic Satellite Capture and Berthing with Robot Arm (ASCABRA)

N95- 23713

Noriyasu Inaba, Yasufumi Wakabayashi
Systems Engineering Department
Office of Research and Development
National Space Development Agency of Japan (NASDA)
2-1-1 Sengen, Tsukuba, Ibaraki Japan
TEL: +81-298-52-2261 FAX: +81-298-52-2247
E-Mail: NINABA@RD.TKSC.NASDA.GO.JP

ABSTRACT

The NASDA office of R&D is studying an automatic technique to capture and berth free-floating satellites using a robot arm on another satellite. A demonstration experiment plan with the Japanese engineering test satellite ETS-VII is being developed based on the basic research on the ground. The overview and key technologies of this experiment plan are presented in the paper, and future applications of the automatic capture technique are also reviewed.

INTRODUCTION

The technique to capture and berth a satellite with a robot arm for on-orbit servicing is widely used in the US shuttle missions (payload supporting missions) such as European Retrievable Carrier (EURECA) and Hubble Space Telescope (HST). However, all these were performed manually by on-board crew members.

One way to achieve effective and economical on-orbit activity is to use unmanned on-orbit servicing systems. The automatic capture technique is one of the most important techniques for realizing this unmanned system. This technique will be used to develop the On-orbit Service Vehicle (OSV) and Geostationary Service Vehicle (GSV).

Following the basic study to develop the automatic capture technique, the NASDA office of R&D developed a capture and berthing experiment plan using Engineering Test Satellite-VII (ETS-VII), which will be launched in 1997. The implementation plan for this additional experiment will be determined in the near future considering the ETS-VII development schedule and operational resources.

This paper presents an overview of the capture & berthing experiment plan and key technologies of the experiment (Figure 1). It also covers future applications of this technique.

EXPERIMENT AI PLAN USING ETS-VII

System Overview

ETS-VII is a NASDA's test satellite for verifying rendezvous docking (RVD) and space robot (RBT) technologies. The RVD system consists of a GPS receiver, rendezvous radar, proximity CCD sensor (PXS), docking mechanism (DM) and on-board guidance computer. RVD experiments will be performed by a 2.2-ton "chaser" satellite and a 0.4-ton "target satellite". A 2-meter, 6-DOF robot arm is attached to the chaser. The Japanese data relay satellite COMETS will be used for nominal RVD and RBT operations.

For the ETS-VII capture and berthing experiment, the chaser satellite will stay in front of the target satellite using PXS data and thrusters. The attitude control of the target will be terminated after the relative stability of the two satellites is obtained. An on-board visual feedback technique is used to guide the special arm effector towards the grapple fixture on the target, while the PXS monitors the relative movement of the two satellites. For the capturing phase, thrusters of the chaser will be inhibited to avoid potential coupling between the reaction control system and the arm control systems. After the capture, the arm moves the target satellite to DM attached to the chaser satellite and berths the target to the DM.

Sequence of Events

The sequence of events (SOE) during the ETS-VII capture berthing experiment is designed considering the following constraints:

- All operations from release to capture of the target satellite must be finished within 30 minutes, which is equal to one pass of COMETS coverage.
- Appropriate lighting conditions must be provided to execute relative navigation using the RVD proximity sensor and RBT visual feedback control of the arm.

Table 1 shows the draft SOE during the experiment. Data link and lighting condition of the SOE is shown in Figure 2.

Load Control in the Capture and Rigidization Phases

To avoid excessive load on the arm and the target satellite, the grapple fixture and the effector were designed with the proper stiffness, and an arm control method will be developed. The effector is designed to have a wide capturing area. The effector's two fingers close and capture the grapple handle of the target quickly, and a sleeve of the effector moves forward relatively slowly to rigidize the effecter to the fixture (Figure 3). A compliance control method using force and moment sensor data is also used to relieve the arm load during the phase.

Visual Feedback

A visual feedback technique is used to guide the effector to the grapple fixture on the target. The software in the robot mission on-board computer (RMOC) calculates the relative position (and orientation: option) between the hand camera and the target mark just beside the grapple fixture on the target satellite at the frequency of approximately 2 Hz. Hand camera image are changed to B/W images to measure center positions and sizes of circles on the target marks, from which relative position and orientation are determined. The threshold for on-board B/W images can be changed by ground command to accommodate lighting condition on orbit. The effector is then guided precisely to the fixture.

Ground tests were conducted using hardware equivalent to flight models. The test results show that the tracking performance of the visual feed back control is precise with relative to the effector's capturing area. Figure 4 shows an image of the target mark taken by the hand camera in ground test configuration.

Safety Control

Automatic failure detection and recovery functions are considered for this crucial experiment to avoid a collision of the two satellites. The RBT and RVD subsystems detect their own faults. In case of failure, a coordinated malfunction procedure will be implemented. The malfunction procedures differ depending on when the failure occurs during the experiment.

The general concept of each case is as follows:

- Approaching phase
  The RBT quenches its movement, and RVD starts the collision avoidance maneuver (CAM).

- Capture phase
  The RBT quenches its movement. The CAM will be inhibited to avoid collision.The ground controller will take over the operation.

- Berthing phase
  The RBT quenches its movement, and RVD stops the drive of the docking mechanism (DM).

FUTURE APPLICATIONS OF ASCABRA TECHNIQUE

The following applications are being considered using the technique of automatic satellite capture and berthing with a robot arm (ASCABRA) for the coming on-orbit servicing era.

- Capturing a supply satellite and vehicle on orbit

Rendezvous and docking is one way to execute the above task. ASCABRA is another way, and it has the merit that fewer active guidance, navigation and control systems are necessary on supply satellites or vehicles. In addition, a robot arm can be used for different kinds of tasks; for example, to transfer on-board replaceable units (ORUs) from satellite to satellite. Instead of preparing many dedicated subsystems, it has the advantage of using a manipulator for many purposes.

- Capturing of on-orbit satellites requiring service

  A Geostationary Service Vehicle (GSV) or On-orbit Service Vehicle (OSV) have been proposed in several agencies and companies to repair, refuel, reorbit, and deorbit satellites. It is not practical to require special and complicated equipment on customer satellites for GSV or OSV to rendezvous and dock. In addition, many customer satellites rotate or tumble on orbit.

Considering these facts, the ASCABRA technique is the most promising way to execute the task.

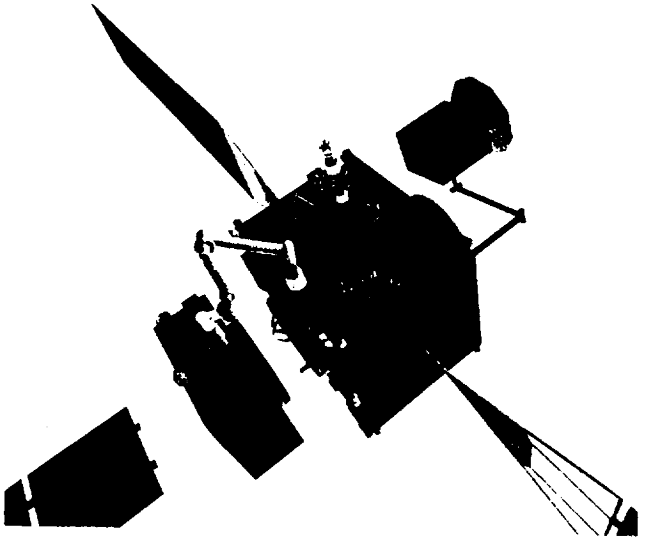


Figure 1: ETS-VII Capture and Berthing Artist's Image

CONCLUSION

Automatic satellite capture and berthing with manipulator is considered a key technology for future on-orbit servicing systems.

The results of the studies to develop the technique and an experiment plan using ETS-VII were presented in this paper.

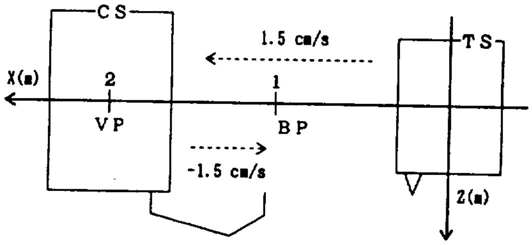
ACKNOWLEDGMENT

This work is supported by NASDA contracts with TOSHIBA Co., Mitsubishi Electric Co., and Kawasaki Heavy Industry Co.

REFERENCES

[1] Inaba N, 1993, "Current Capture Berthing Techniques and Future Applications", RS-U93049, NASDA Office of R&D.

[2] Inaba N, 1994, "ETS-VII Capture and Berthing Experiment Plan", GAS-94005, NASDA Office of R&D.



NO	ACUMULATED TIME (MM:SS)	DISTANCE CS to TS (m)	EVENT
①	00:00	0.35	TARGET SATELLITE (TS) SEPARATION
②	03:00	2	VICINITY POINT (VP) ARRIVAL
③	05:00	2	VP DEPARTURE
④	07:00	1	BERTHING POINT (BP) ARRIVAL
⑤	09:00	1	ARM MORTION START (PROGRAM)
⑥	10:00	1	ON-BOARD VISUAL FEEDBACK START (TS ATT. CONT. OFF)
⑦	12:00	1	FUNCTION CHECH FROM GROUND. CS THRUSTERS INHIBIT.
⑧	14:00	1	FINAL APPROACH TO GRAPPLE FIXTURE
⑨	14:10	1	CAPUTURE BY TOOL HEAD FINGURES
⑩	14:14	1	RIGIDIZATION
⑪	14:30	1	STABILIZATION
⑫	15:30	1	MOVE TS TO DOCKING MODULE (DM) POSITION BY ARM
⑬	17:50	0.42	DM DRIVE TO BERTH. ARM COMLY
⑭	19:40	0.38	TOOL HEAD OPEN. ARM WITHDRAW
⑮	21:40	0.38	FINAL BERTHING BY DM
⑯	22:00	0.35	EXPERIMENT OPS. TERMINATION

Table 1: SOE during the ETS-VII Capture and Berthing

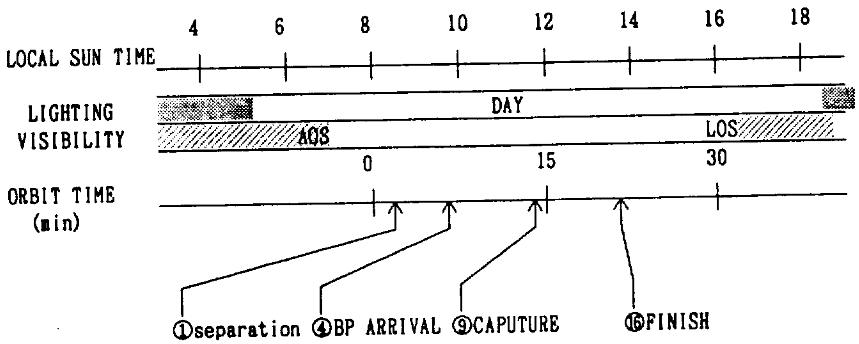


Figure 2: SOE (Data Link & Light Condition)

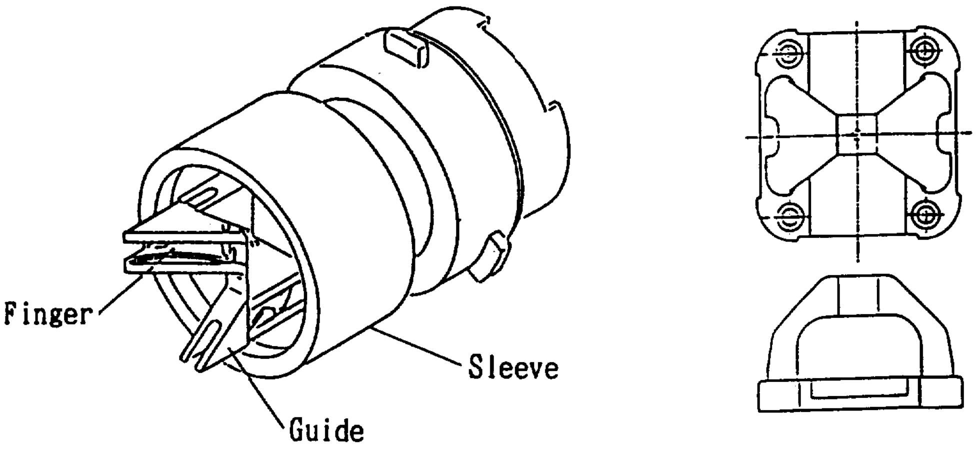
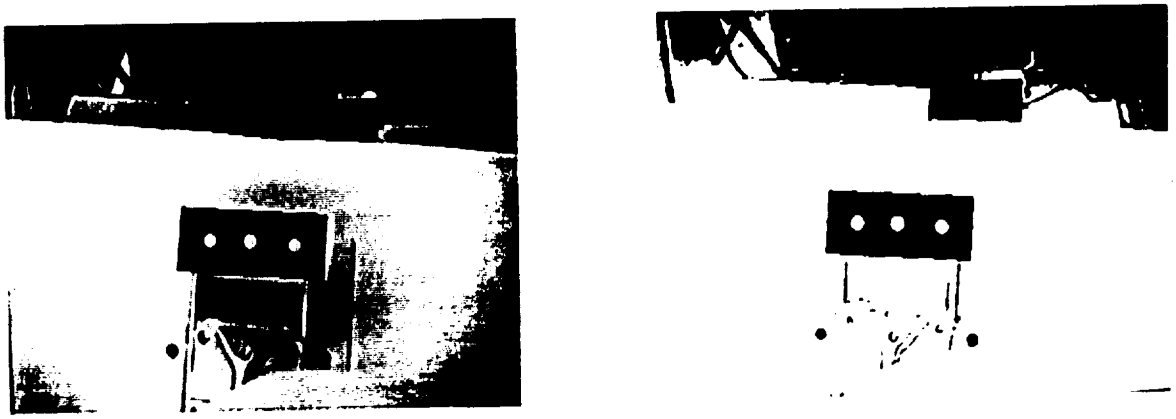


Figure 3: Caputuring Effector and Fixture



Original Image

B/W Image

Figure 4: Target Mark Images (Hand Camera View)

POINTER: Portable Intelligent Trainer for External Robotics

N95- 23714

Hilbert Kuiper, Patrick J. Rikken
TNO Physics and Electronics Laboratory
PO Box 96864, 2509 JG The Hague, The Netherlands
Telephone +31 70 3264221, Fax +31 70 3280961
E-mail: Kuiper@fel.tno.nl or Rikken@fel.tno.nl

KEY WORDS AND PHRASES

In-orbit training, intelligent tutoring systems, refreshment training, robotic arm.

ABSTRACT

Intelligent Tutoring Systems (ITSs) play an increasing role in training and education of people with different levels of skill and knowledge. As compared to conventional Computer Based Training (CBT) an ITS provides more tailored instruction by trying to mimic the teaching behaviour of a human instructor as much as possible and is therefore much more flexible.

This paper starts with an introduction to ITSs, followed by the description of an ITS for training of an (astronaut) operator in monitoring and controlling robotic arm procedures. The robotic arm will be used for exchange of equipment between a space station and a space plane involving critical and accurate movements of the robotic arm.

The ITS for this application, called Pointer, is developed by TNO Physics and Electronics Laboratory and is based upon an existing ITS that includes procedural training. Pointer has been developed on a workstation whereas the target platform was a portable computer. Therefore, a lot of attention had to be paid to scaling effects and keeping up with user friendliness of the much smaller user interface. Although the learning domain was the control of a robotic arm, it is clear that use of Intelligent Training Technologies on a portable computer has many other applications (payload operations, operation control rooms, etc.). Training can occur at any time and place in an attractive and cost effective way.

INTRODUCTION

Intelligent Tutoring Systems (ITSs) emerged with the advent of Artificial Intelligence research. Conventional Computer Based Training (CBT) methods were of limited flexibility and had a number of disadvantages: every step in the learning process had to be pre-programmed, domain knowledge was hidden in learning material and initiative was mainly taken by the computer. This often resulted in boring learning material presentation.

An ITS tries to mimic a human instructor as much as possible by combining Artificial Intelligence techniques with Computer Based Training. These techniques enable reasoning on domain knowledge which is the basis for adaption to student behaviour and answering student questions about the learning domain.

In general, ITSs are built according to an architecture that contains five modules: domain expert module, tutorial module, interface module, student model module and control module.

In the late 80s TNO Physics and Electronics Laboratory gained experience with Intelligent Tutoring Systems by developing an ITS for a message handling domain. The good results of this ITS, that includes procedural training, encouraged us to continue research and development in the field of intelligent training technology. Based upon the existing ITS framework for message handling TNO Physics and Electronics Laboratory has now developed an ITS, called Pointer, for the European Space Agency (ESA). The Pointer-project has the objective to investigate the feasibility of an intelligent system for training complex robotics operational procedures.

The uniqueness of Pointer lies in the fact that an existing operational application for operating a robotics arm is combined with (the functionality of) an existing ITS. This feasibility study has made the requirements for robotic training systems explicit.

AN ITS FOR SPACE ROBOTICS

Learning domain: robotic arm

The domain to be addressed for the ITS is the operation and control of an External Robotic Arm (ERA). This manipulator arm is used to provide ESA with a robotic in-orbit space plane servicing capability. It is used for the exchange of supplies or equipment boxes generally called Orbit Replaceable Units (ORUs) between e.g. a space plane resource module and a space station.

The robotic arm, which is still under development, has 7 joints and 6 degrees of freedom. Its length is 9.09 meters and it can manipulate objects from several kilos up to 20 tons with a high accuracy. It is to be operated in a 0-G space environment, either from the inside of the space station (MIR) or in EVA (Extra-Vehicular Activity). Monitoring of arm operations takes place via three cameras that are connected to the arm.

Robotics tasks

ERA is a priori meant to be used for several mission types or even space operation scenarios. In this section we describe the type of robotic tasks foreseen for the MIR2 space station. However, it is evident that the design of ERA has quite a generic value.
The typical tasks foreseen for MIR2 are:

- assembly of truss elements;
- mounting of station bulky elements;
- re-docking of station modules;
- installation/removal of orbit replaceable units (ORUs);
- inspection.

In the MIR scenario, certain tasks such as ORU or payload transfer require the presence of an astronaut in EVA to perform proximity operations such as final placement of objects attached to the robot.

Also in this scenario, ERA's base is installed on a trolley which can move along a truss structure. Strictly speaking, operations of the trolley are not considered part of the ERA operations, although an astronaut would have to learn to operate it.

Operation and control of the robotic arm

To supervise, or perform, the tasks, the operator has two interaction devices depending on his location:

- inside the vehicle, he will use a portable unit, called ERA Portable Brain (EPB);
- outside, in EVA, the astronaut will use a simpler device, called the EVA panel, which is based on the direct view the astronaut has on the operations.

The EPB, in its current implementation, is a portable high-performance workstation which includes:

- a synoptic area for displaying status information and arm movements in a graphical manner;
- commands, acknowledge and stop switches;
- choice of mode, control gains, procedures and graphical views;
- 2 video screens, external to the workstation but attached to it, to display the camera views as seen from the robot.

The EVA panel is a much more simpler device allowing automatic and manual modes with only a numerical display and selection switches.

The tasks of the operator will be:

- in automatic mode, to supervise and acknowledge transitions; eventually hold or stop the robot arm (emergency) while it is following pre-defined procedures;
- in manual mode, to control degrees of freedom, one at a time, but also according to a pre-defined procedure.

The complexity, criticality and required accuracy of the tasks to perform, make high demands upon robotic arm operators. Therefore, training facilities that guarantee the education of personnel that is well-qualified to perform the job, are of essential importance.

Training issues

Based upon the tasks mentioned in the former paragraph, the high level objective of the training is learning to control and monitor a robotic operation through a portable monitoring workstation.

To achieve this objective, both the interactions with the system and the procedures to operate the ERA must be trained. Each procedure is split up in a number of segments. At the end of each segment the operator must perform status checks and use acknowledge functions in order to continue in automatic mode. Errors in the automatic progress must be recognized and if necessary the operator must stop the progress and take corrective actions. These corrective actions often require the ERA to be piloted. Piloting the ERA is a complex activity where the operator must constantly be aware of physical limits (e.g. accuracy and speed), time limits and power consumption limits of the ERA.

In summary, the operator should learn:

- a great variety of foreseen tasks;
- to recognize and deal effectively with non-nominal situations as well;
- to perform an enforced procedure concept;
- all elementary functions, which are the building blocks of all operations;
- operational rules concerning safety procedures, EVA/ERA cooperation, communications with the ground;
- the physical limitations of the arm with respect to kinetic and dynamic behaviour, accuracy, speed, etc.

Constraints. The operator is supported by a number of displays. However, the portable computer poses a number of constraints on the Man Machine Interface:

- because displays can be called up one at a time, the operator should only request these displays and information pages when he really needs them and with a given purpose in mind.
- scarce MMI resources in the EPB impose a certain slowness in the operations;
- since not all MMI information is available at hand, the interaction with the workstation is also part of the operator procedure. Thus training will cover these aspects also.

Motivation for a Portable ITS

The need for an intelligent and portable training system originates in:

- a requirement for self-training and refresher training. In the long duration mission foreseen for MIR2, robotic tasks will have to be relearned by astronauts on their own. Thus the ITS will need to take over parts of the instructor role. In particular, procedural training is addressed here and the need for a student model which allows monitoring of the trainee performance was identified early.
- a requirement for a portable unit because the EPB itself is a portable unit. Thus the attached training system would have to be also portable.

Pointer: the ITS-solution

Because one aspect of the existing ITS was aimed at procedural training and because it has a domain independent framework, this ITS was chosen as the starting point for the robotics application. The same architecture was used, however extended with a simulation and an EPB-application interface.

The tutoring system is composed of three main parts:

1. a simulator of the ERA
2. the EPB application itself connected to the simulator
3. a tutoring environment which includes the EPB application

Two ITS modules, domain expert module and interface module, are described:

Domain Expert module. The expert module consists of a formalized domain knowledge base and an interpretation mechanism to reason about this knowledge. Reasoning occurs when a student asks for support while learning to perform a procedure. The student can ask what the next action is he should perform (forward), ask for a hint what to do next (hint), let the system predict what will happen if he performs a certain action (what if ...), ask to evaluate his behaviour (evaluate) or request why certain actions were wrong (explain). This expert functionality allows the student to take initiative in the learning process himself and

makes the learning process much more flexible.
In the domain expert module, clear separation of domain knowledge and interpretation mechanism has been taken into account. This enables reusability of the interpretation mechanism and easy maintenance of domain knowledge. This is an important benefit as compared to conventional computer based training because updates of the domain only have to be imported in the knowledge base instead of in all hardly traceable places of the courseware that are affected by such an update.

Interface module. The delivery system is a portable workstation with a small screen. It therefore imposes some constraints upon the implementation of the user interface. So, the user interface has to be very efficient.

The solution that was chosen to deal with the small size of the user interface is based on the principle of sliding windows. We created a virtual screen that is exactly twice as large as the display of the portable computer. The left side of the virtual screen holds windows of Pointer, the right side holds windows of ERA Portable Brain. By sliding the pointing device from left to right, the student can switch from learning environment to the application he is learning about. This solution ensures that it is always clear to a student which windows belong to Pointer and which to the EPB.

The Pointer side of the virtual screen contains a button area, an area for lectures, questions, tasks and feedback and an area for pictures and animations. The user interface is user-friendly, intuitive and consequent.

Learning procedures. Procedures are learned in small parts, called topics. The student level determines the size of these parts. For every topic, a short lecture with text, pictures, animations and examples precedes questions that are asked to check if the student is ready to perform a task. While performing a task, the student is supported by expert functionality to deal with uncertainties about how to go on or to satisfy his curiousity. After mastering all parts of a procedure segment, a task is generated to perform this segment. Pointer adapts the learning process to the student level. The student can take initiative, is aware of his progress and feels confident.

Development and delivery environment

The system was developed on a commercially available workstation and is delivered on a portable workstation. Programming language is C, MOTIF is the look and feel and DataViews and X-Designer were used to create the user interfaces.

Further areas of research and development

In this study we have proven the applicability of intelligent tutoring techniques towards procedural training, also taking into account the specific means of interaction of the operator with the robot arm. Further work will concentrate on contingency training, system level training, integration with virtual reality and ground operator training.

CONCLUSION

This paper has focussed on Pointer, an Intelligent Tutoring System for training complex operational procedures. Based upon the positive experiences with a former developed ITS, this teaching technology offered promising results that have been caught up with a new (portable) application for a robotics trainer for space applications. The new ITS is an important resource for establishment of requirements for intelligent robotic training systems.

Although the chosen domain was the robotic arm ERA, it is clear that the use of Intelligent Training Systems on a portable computer has many other applications (e.g. payload operations and operation control rooms), everywhere there is a particular need for refresher training and self training. By using an Intelligent Training System on a portable computer training can occur at any time and place, finally in an attractive and cost effective way.

SESSION SA

Science Assistant Systems

Automation and Crew Time Saving in the Space Experiment

N95- 23715

Kohtaro Matsumoto, Tsuyoshi Suzuki
Invited Scientist
Space Experiment Department, Office of Space Utilization Systems,
National Space Development Agency of Japan (NASDA)
2-1-1, Sengen Tsukuba, Ibaraki 305, Japan 81-298-52-2764, FAX:81-298-52-1597
NLMATSUMOTO@rd.tksc.nasda.go.jp

Kouichi Funaya
Functional Devices Research Laboratories, NEC Corporation
1-1, Miyazaki 4-Chome, Miyamae-ku, Kawasaki 216, Japan 81-44-856-2037,FAX:81-44-856-2210

Takeya Kawamura
Mechatronics Equipment Development & Design Sec., Information Processing & Electronic Systems Dept.
Kobe Shipyard & Machinery Works, Mitsubishi Heavy Industries, Ltd.
1-1-1, Wadasaki, Hyogo-ku, Kobe 652, Japan 81-78-672-2811 FAX:81-78-672-2985

Masayoshi Sonobe
Space Systems Design Center, Hitachi Works, Hitachi Ltd.
1-1 Saiwaicho 3, Hitachi, Ibaraki 317, Japan 81-294-21-1111 FAX:81-294-24-1464

ABSTRACT

In this paper, we will describe the preliminary results of the feasibility study of the automation and crew workload saving in the space experiments. Four apparatuses have been selected as the study case. In this paper, three results will be summarized. The fourth result will be described in other paper.[1]

1. INTRODUCTION

During the restructuring and the re-design efforts and during the user integration work for the JEM, it has been revealed that crew work might be too short on the space station.

The common experiment apparatuses for the initial utilization of the JEM have been under development already. Some automation functions have been studied for the devices, that can be automated within a single rack and without major impacts for the development process and costs.

In addition to such automation, in our study, we assume the following premises to develop new concepts;

(1) Applicable as the second generation apparatuses.
(2) Maximum reduction of the crew workload.
(3) Automation between racks including the storage.

2. CONCEPTS FOR THE AUTOMATION AND CREW TIME SAVING (A&C)

In this chapter, three A&C concepts will be described for the material processing furnace, the life sciences experiment apparatus, and the cleanbench. During the study, most of the effort were devoted to keep the rationality of the experiment itself with the minimum non-scientific degradation by the automation and/or robotics concepts.

2.1 A&C FOR THE MATERIAL PROCESSING EXPERIMENT FURNACES

(1) Tasks for A&C

The current design for the furnaces closes to its automation inside the experiment module itself. The effort for A&C focuses on the exchange of the sample holders for the logistic operation.

In the JEM, most of the logistic materials will be stored in the pressurized logistic module, that will be connected vertically to the JEM through the connecting hatch. The A&C system transfers and or hand-overs the sample holders through the hatch.

Specimen cartridges are covered with the packing or shock-absorbing materials. Unpacking and attaching cartridges to the sample holder will be the most time consuming crew works. The A&C system saves the crew workload by automating these tasks.

(2) Major elements of the automatic re-supply device for the furnaces

Fig. 1 shows the system structure and its hand-over operation through the hatch. Proposed A&C system consists of apparatuses explained below.

(a)Sample holder for launch environment: Multiple specimen cartridges are held in a holder. Each holder is designed to simplify unpackage and package tasks and hand-over tasks.

(b)Carrier mechanism: Wire driven planar carrier mechanism is used both in the logistic module and in the pressurized module. When no logistic module is used, storages are placed in the pressurized module.

The proposed design improves safety of the crew-carrier co-working. The wire driven mechanism

114

enables driving actuators to be placed in a fixed and remote area from crews. Because moving part weight is reduced, safety against the possible collision with crews improves. Also the planar implementation reduces the interference volume between crews and carrier mechanisms.

(c)Handover mechanism through the hatch Two hand-over mechanisms are installed close to the hatch. One is in the pressurized module and the other is in the logistic module. Each mechanism hand-overs one sample holder between the carrier and the another hand over mechanism. These mechanisms are folded and placed outside the hatch to keep the hatch clear when not used.

(d)Sensors, cameras: Proximity sensors are installed to detect crews and unexpected obstacles on the carrier's moving path. System cameras in the JEM or a new inspection camera are used to supervise carrier motion and to check sample cartridge defects.

(3) Merits and Problems

Proposed A&C system significantly reduces the crew workload. Though visual inspection by the crew or by the operator on the ground is still needed when new sample holder is unpacked, time for inspection is small compared with the whole time to complete sample exchange.

To keep the safety level high, carrier mechanism motion is restricted to relatively slow. This inefficiency can be improved by operating in more large velocity, when the crew is absent from JEM.

(4) Safety Investigation

There are two safety issues to be considered. One is the crew safety when carriers are moving. Another is the system safety against unexpected environment changes.

To assure the crew safety, A&C system is designed with the safety guideline shown in Table1. The carrier design fits this guideline as stated above.

The system safety is improved with sensors installed on the moving part. Even if the crew leaves something on the carrier path, sensors detects it and stops the motion. Obstacles can be removed afterwards by crew or by the teleoperation from the ground.

Before the full co-working operation, several development steps shall be considered to assure the safety functions completeness. Table 2 shows the possible development steps.

(5) Future Subjects

Items listed below should be investigated further.

-Detailed interfaces with JEM system:
electrical and mechanical
-Carrier mechanism performance:
accuracy, compliance, driving power, etc.
-System control method:
autonomous control and/or manual control.
-Testbed experiment of proposed design:
feasibility test and reliability test through on Telescience testbed experiments on the ground.

2.2 A&C FOR JEM LIFE SCIENCE EXPERIMENT

(1) Tasks for A&C

The current design for JEM incubator and cell culture devices covers only the crew operation. The A&C concepts will be required in accordance with growing needs of experiments and much less availability of crew works. For JEM incubator and cell culture devices, following factors shall be considered:

(a) Automation of experiment devices
(b) Automation of observation devices
(c) Implementation of inter rack/device operation support system (IRDOSS)

(2) Automation scheme for JEM life Science

(a)Experiment devices: In the current design, number of devices have been modified from the Spacelab's devices to improve operability, instead to enable the automation, because the automation of each devices would result in so bulky design of them. For example, the elastic bungee to fix samples in the incubator, and polyethylene soft bag to securely contain samples, both adopted on Spacelab's incubator, required much crew involvement. So, the rail sliding tray that can securely load samples, and hard cases with transparent window are adopted on JEM incubator to replace bungees and soft bags.

(b)Observation devices: On Spacelab, high quality observation devices, such as camcorder, camera, and microscope were used to adapt to the variety of experiments. The problem was that they are originally designed for commercial use and required much crew involvement during the observation.

On JEM experiment, standardization of sample size and remote command to the observation devices (commanding of zoom, focus, exposure, and so on) are considered to reduce crew involvement. The automation of exchanging lenses and films remain critical to achieve unmanned observation.

(c)Implementation of IRDOSS : In order to achieve unmanned sample exchange across the racks, automated handling system should be implemented into the existing devices. We call this system inter-rack /device operation support system (IRDOSS). Conceptual design of IRDOSS is shown in Fig.2. The system is composed of sets of articulated manipulators and planar positioning mechanism. The positioning mechanism can be attached to the seat tracks on each experiment rack, and the manipulator on it can reach every experiment equipment on the rack, open the door, and fetch bio-sample from the incubator or the stowage container.

(3) Merit and Problems

Those devices designed for unmanned operation would contribute to reduce crew involvement during manned operation. However, some of the devices such as hard cases, are likely to be too bulky or massive to be handled and stowed.

(4) Crew Safety in IRDOSS

For IRDOSS, safety is most important, because IRDOSS cannot avoid working with crew. For this matter, the smooth surface of the manipulator, or the proximity sensor to detect the crew shall be considered .

(5) Future subject

Ground test for IRDOSS will be demonstrated.

2.3 A&C FOR THE TELEOPERATION OF THE CLEANBENCH

(1) Tasks for A&C

Tasks inside the cleanbench are those such as exchange the medium of the culture cell, procedure to preserve samples, micro-manipulation, observation using the phase-contrast microscope. In this study, those typical task for the life science experiment are subjected to A&C.

(2) Concepts for A&C

The cleanbench A&C is achieved by automation and teleoperation of the following tasks listed below.
(a)Preparation of the Cleanbench: Uplink Experiment Process Managing Program (EPMP). Temperature, cleanliness and other condition of the cleanbench is controlled by the execution of EPMP.
(b)Transfer of samples : For the handling operation, two types of handling manipulators will be utilized. One will serve for the handling between the cleanbench and the other devices such as incubators and refrigerators. Another tiny manipulators will serve handling within the cleanbench. (Fig.3)
(c)Sterilization in the Airlock : In the airlock, equipment that goes through will be sterilized by 70% ethanol. Spray the ethanol, removal of the ethanol, monitor the concentration of the remaining alcohol shall be automated or teleoperated.
(d)Task performed in the cleanbench's working chamber : The culture cell is handled by the tiny manipulators. The cell is positioned to the Automated Sample Manipulation System (ASMS) and exchanging the medium of the culture cell and preservation of the samples are executed. Mircomanipulators and the phase-contrast microscope is able to be controlled by joystick and keyboard from the ground.

(3) Merit and Problems

(a)Merit

(i)Saves Crew time.
(ii)On certain task, PS participation will not be needed. Automated task may be able to perform more precise work than the PS.

(b)Demerit

(i)Total weight of the cleanbench increase.
(ii)On board computer is preferred to be multitask and high-performance.
(iii)Automated cleanbench may need major remodeling when the crew stays on orbit permanently and some automated part of cleanbench becomes obstacle.
(iv)Consumption of electrical power may increase.

(4) Crew Safety Assurance

(a)When equipment are teleoperated or operated automatically, crew are prohibited to enter the working area.
(b)The sensor shall be attached to the equipment. The sensor system avoids collision of crew and equipment.

(5) Future Subject

(a)To ensure crew's safety, when equipment are teleoperated or operated automatically.
(b)Recovery strategy from the handling error, such as release anomaly.

3. CONCLUDING REMARK

At present, the first steps were taken to the A&C evaluation. Those three results described here have each depth of its concepts and also have variety of positions to the A&C implementation. The integrated concept will be needed in the next step of A&C evaluation.

Also for the next step of A&C feasibility study, in addition to the follow on study of the above subjects, a couple of demonstration experiments using Telescience testbed will be investigated in this year.

REFERENCES

[1]Yamawaki, T., Shimoji, H., Abe, T., 1994. Robotic Servicing System for Space Material Experiment. In Proceedings of i-SAIRAS'94.

Table 1 Safety Guideline of Co-working System

1. Low mass property of the movable part.
2. Smooth shape without protuberance.
3. Manual operability in anomaly situation.
4. Monitoring of non-safety action.

Table 2 Possible Development Step of the A&C Safety

1. Ground test:
   - Functional Test of mechanics & software.
   - Long term validation operation using test devices.
   - Exhaustive test of obstacle sensors.
   - Emergency shutdown test for various situation.
2. On-orbit test:
   - With Crew : Functional test monitored by crew.
     Emergency stop by crew.
   - Without Crew: Programmed and teleoperated test.
3. Unmanned Operation:
   - Ground Checkout against the damage of specimens.
   - Autonomous recovery for partial emergency.
   - Recovery operation by the ground teleoperation.
   - Recovery by crew for serious anomaly.
4. Manned Operation
   - Crew checkout against the damage of specimens.
   - Ground monitoring for safety operation.
   - Effective recover operation by crew.
   - Consideration of the anomaly induced by crew.

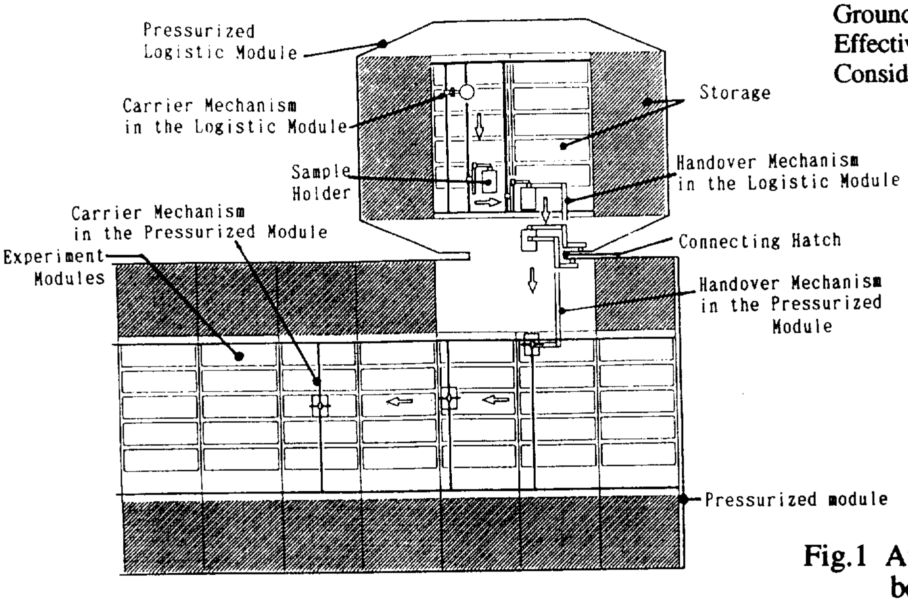


Fig.1 Automatic Re-Supply System between Modules.

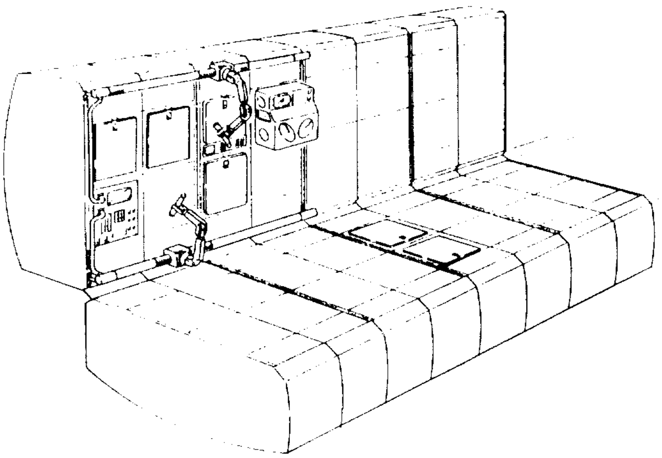


Fig.2 Inter-Rack/Device Operation Support System (IRDOSS)

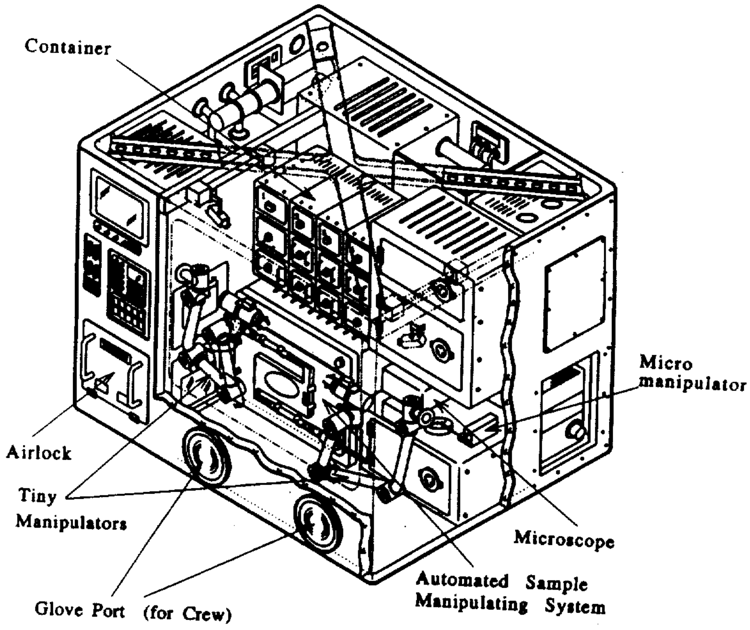


Fig.3 Automated and Tele-Operable Cleanbench

N95-23716

A Toolbox and Record for Scientific Models

Thomas Ellman
Department of Computer Science, Hill Center for Mathematical Sciences
Rutgers University, Piscataway, New Jersey, 08855, USA
Phone: (908) 445-4184, FAX: (908) 445-5530
Email: ellman@cs.rutgers.edu

Key Words and Phrases

Artificial intelligence, scientific computation, automated software design, knowledge representation, approximation.

Difficulties of Scientific Programming

Computational science presents a host of challenges for the field of knowledge-based software design. Scientific computation models are difficult to construct. Models constructed by one scientist are easily misapplied by other scientists to problems for which they are not well-suited. Finally, models constructed by one scientist are difficult for others to modify or extend to handle new types of problems. Existing knowledge-based scientific software design tools, such as SIGMA (Keller & Rimon 1992), provide only limited means of overcoming these difficulties. For example, SIGMA facilitates model construction by providing scientists with high-level data-flow language for expressing models in domain-specific terms. Although SIGMA represents an advance over conventional methods of scientific programming, it supports only certain aspects of the model development process. In particular, SIGMA focuses mainly on automating the process of assembling equations and compiling them into an executable program. Construction of scientific models actually involves much more than the mechanics of building a single computational model. In the course of developing a model, a scientist will often test a candidate model against experimental data or against a priori expectations. Test results often lead to revisions of the model and a consequent need for additional testing. During a single model development session, a scientist typically examines a whole series of alternative models, each using different simplifying assumptions or modeling techniques. A useful scientific software design tool must support these aspects of the model development process as well. In particular, it should propose and carry out tests of candidate models. It should analyze test results and identify models and parts of models that must be changed. It should determine what types of changes can potentially cure a given negative test result. It should organize candidate models, test data and test results into a coherent record of the development process. Finally, it should exploit the development record for two purposes: (1) automatically determining the applicability of a scientific model to a given problem; (2) supporting revision of a scientific model to handle a new type of problem. Existing knowledge-based software design tools must be extended in order to provide these facilities.

An Artificial Intelligence Approach

We are attacking this problem using two related ideas: First, we are building a "Model Development Toolbox". The toolbox will support a set of generic model development steps that are taken by most scientists in the course of developing scientific computational models: Examples of such generic model building steps include: (1) mapping equations onto physical situations; (2) fitting models against experimental data; (3) testing models against experimental data; (4) testing applicability of models to given inputs; and (5) modification of models in response to test results. Second, we are designing a "Model Development Record". The record will contain machine readable documentation of the entire model development process. To begin with, the record will describe the goals the model is intended to fulfill. For example, this might include a representation of the questions the model is (and is not) intended to answer. The record will also describe the sequence of candidate models that were constructed in the course of developing the final model. For each candidate model, the record might describe: (1) the equations encoded in the model; (2) assumptions underlying the model; (3) fitting techniques used to instantiate free parameters of the model; and (4) tests against empirical data that were performed on the model. The record must also describe (5) the temporal sequence of candidate models as well as (6) logical dependencies between test results on early models and modeling choices made in constructing subsequent, more refined models.

Tools for checking applicability of scientific models to new problems will rely heavily on the model development record. Important applicability checks include: determining whether a proposed use of a model is consistent with the goals the model was originally intended to fulfill; determining if a new problem lies within the range of inputs for which the model was tested; and testing assumptions underlying the equations that were incorporated into the model. Each of these checks requires access to various aspects of the model development record. Likewise, tools that support model revision will also rely heavily on the model development record. Important types of model revision include: extending/modifying the model to handle a wider/different range of input parameters; re-fitting free parameters of the model to new empirical data; changing the assumptions used to model a physical process; adding/deleting physical processes to/from the model; and changing the overall purpose of the model. A model revision tool should automatically determine when a revision is needed (e.g., by determining that a new problem falls outside the range of problems handled by the original model, or by detecting discrepancies between empirical data and outputs of the model). It should suggest changes to the model that have the potential to cure the problem (e.g., by reasoning about sensitivities of outputs with respect to changes in intermediate results, or by reasoning about the effects of potential changes in assumptions on the outputs of the model). Finally the system should assist in re-validating the new model, (e.g., by suggesting new tests of validity, and carrying out and evaluating such tests.) In many cases, models may be revised by "replaying" a portion of the development record that led to the original model. Replay will require access to logical dependencies among test results and modeling choices found in the development record, using techniques similar to derivational analogy (Mostow 1989) and transformational implementation (Balzer 1985).

System Architecture

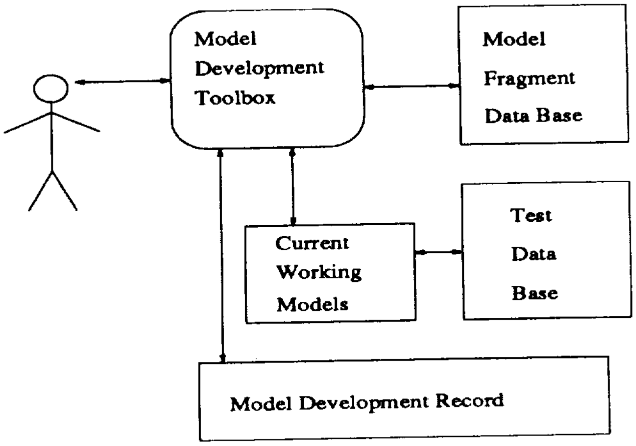


Figure 1: Model Development System Architecture

The overall architecture of our envisioned system is shown in Figure 1. The model development toolbox serves as a front end to the whole system. The toolbox interacts with a human user to build an initial model in some scientific domain. It also interacts with a user in order to revise an existing model to handle a new situation. Finally, the toolbox also includes facilities for controlling the application of scientific models. As the toolbox guides the user through a series of model building, testing and revision steps, it interacts with several data bases. The model fragment data base contains the basic building blocks of scientific models. The toolbox uses techniques embodied in the SIGMA system to combine model fragments into one or more "current working models". As working models are constructed, they are tested against test data drawn from a test data base. Likewise, as tests are run, results are incorporated back into the test data base. As the initial model development process unfolds, the toolbox leaves a structured trace of the process in the model development record. Later on, the scientist will apply the model to specific problems in which he is interested. As the model is applied to each problem, the system consults the model development record to determine whether the model is valid for the current problem. If the model fails to apply, the scientist may use the toolbox to revise the model. During the revision process, the toolbox is also guided by the model development record. The toolbox and record is being implemented as an extension to the SIGMA scientific model building system (Keller & Rimon 1992). Testbed domains for this research include planetary atmosphere modeling and ecosystem modeling problems used in development of SIGMA. Additional testbed domains include two problems under investigation in computer-aided design research at Rutgers University: modeling of jet engine nozzle performance and modeling the motion of sailing yachts.

Controlled Application of Models

Implementation of the model development toolbox and record is initially focusing on methods for controlling the application of scientific computation models. To this end, we have developed a collection of techniques that prevent users from applying scientific models to situations which violate their implicit assumptions and lead to erroneous or meaningless results. Some of these tests can be applied to virtually any scientific model. Such generic test include: (1) comparing inputs, outputs or intermediate results to fixed bounds; (2) verifying expectations about monotonicity or uni/multi-modality of computed functions; (3) validating results in comparison to simplified models. We have also defined a collection of more specialized tests, whose relevance depends on the specific idealizations, approximations or abstractions that were used to construct the model. Examples include: (4) checking nearness to the fitting point of a linear approximation and (5) verifying self-consistency of solutions obtained by decomposing systems of equations, among others.

Our applicability testing techniques require that models be represented in a manner that makes explicit what tests are required and how the tests should be applied. For this reason, we have developed and implemented a model representation language that contains applicability checking information. Our representation is an extension of the dataflow graphs used in SIGMA (Keller & Rimon 1992). The representation includes annotations that describe what applicability tests should be carried out at model execution time. The annotations are linked to the dataflow graphs in a manner that allows the system to determine the stage of the computation at which each applicability test should be carried out. We have also defined and implemented a general model execution procedure that refers to the annotations to perform the required applicability tests during the course of model execution. We have implemented and tested several versions of a jet engine nozzle performance model and a yacht velocity prediction model in the new representation along with applicability tests suitable to each.

An example of a scientific model represented as a dataflow graph is shown in Figure 2. This graph represents a model for computing the steady state velocity of a sailing yacht as a function of several geometric and physical parameters of the yacht, (e.g., vertical center of gravity (VCG), wetted surface area (WSA), longitudinal second moment (LSM), effective draft (T_{eff})), as well as inputs describing the sailing conditions, (wind-speed (V_{tw}) and heading angle (B_{tw})). The model describes a computation that proceeds in two stages. The first stage is to solve the torque balance equation $NetTorque(\phi) = 0$ which asserts that "heeling" torques (causing the yacht to heel over in the wind) are equal to "righting" torques (causing the yacht to remain upright). The solution value of ϕ is the heel angle at which the yacht will sail. The second stage is to solve the force balance equation $NetForce(v, \phi) = 0$ which asserts that "thrust" (due to the wind acting on the sails) is equal to "drag" (due to the friction caused by water). The solution value of v is the steady state velocity of the yacht. The "Torque Balance" and "Force Balance" nodes of this graph each describe submodels of the overall yacht velocity prediction model. Each submodel is itself represented by a dataflow graph that describes a process of solving an equation using a numerical root-finding algorithm (Brent's method). This yacht velocity model is only an approximation of a more accurate model of the yacht's motion. The more accurate model solves for velocity v and heel angle ϕ simultaneously using a pair of coupled torque balance and force balance equations. The coupling is due to the fact that $NetTorque$ actually depends on both ϕ and v, just as $NetForce$ depends on ϕ and v. The coupled model is generally more accurate; however, it takes longer to run. It is also more brittle than the uncoupled model since it uses a two-variable equation solver (Newton-Raphson) which more often fails to find a root than the two one-variable (Brent) equation solvers used in the uncoupled model.

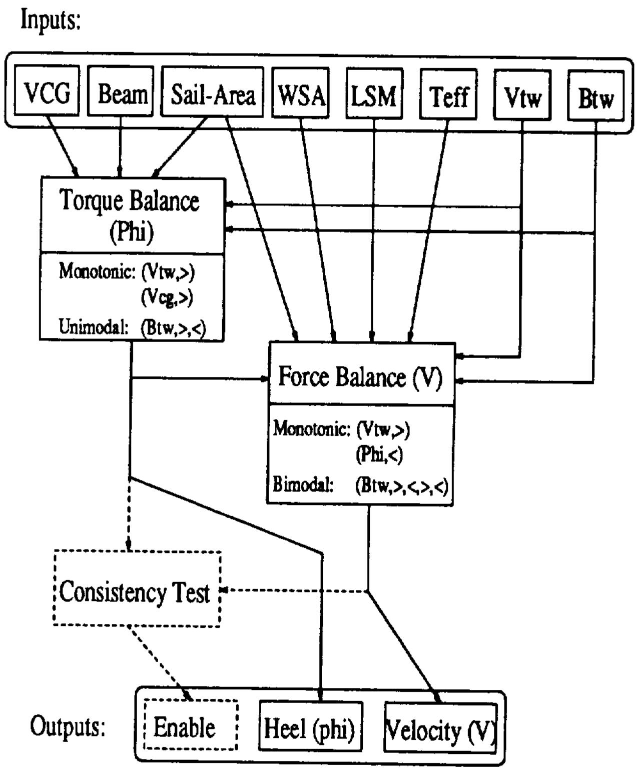


Figure 2: Yacht Velocity Model Dataflow Graph

The yacht model dataflow graph illustrates our approach to representing applicability tests as annotations to dataflow graphs. Some types of tests are general enough to apply to virtually any numerical model of a physical system. Examples of this type include testing whether a model exhibits a qualitative behavior that can be described in terms of monotonicity, unimodality or multimodality of the function computed by the model. These qualitative tests are represented as special slots appearing in each dataflow node object. For example, in the "Monotonicity" slot, an entry of the form $(Input, Sign)$ (where sign is one of $\{\geq, >, \leq, <\}$ indicates that the model's output is expected to be monotonic (increasing, strictly increasing, decreasing, strictly decreasing) in the named output. For example, the monotonicity slot in the "Force-Balance" object includes the entries $(V_{tw}, >)$ asserting that the velocity output v is expected to be a strictly increasing function of the wind speed V_{tw}. Whenever a model is executed, the execution procedure examines the monotonicity and modality slots, extracts descriptors of the expected qualitative behavior, and tests whether the current execution of the model is consistent with that behavior. The current execution is checked by examining a database of results of previous model executions and verifying that the current results bear the correct qualitative relationship to previous results.

Some types of tests are highly specialized, and apply only to a small number of models, perhaps only one model. We represent these tests as special "applicability checking nodes" that are directly wired into the dataflow graph. An example of this type is the "Consistency Test" node in the yacht model dataflow graph. The consistency test checks whether the decoupling of the torque balance and force balance equations is a good or bad approximation. It does so by evaluating the solution values of ϕ and v in the inequality $NetTorque(v, \phi) \leq K$. This test measures whether the approximate solution brings the net torque close enough to zero. By representing applicability tests as additional nodes in a dataflow graph, our system allows arbitrary computations to be used for applicability tests.

Although applicability checking nodes are represented in the same manner as the main stream of the computation, they are not handled in the same fashion by our model execution procedure. To begin with, applicability nodes are kept separate from ordinary nodes. Under the control of the user, the system can execute the entire graph, include applicability checks, (running in "restricted mode") or the system can execute only the subgraph representing the main stream of the computation (running in "unrestricted mode"). Furthermore, our execution procedure allows the applicability checking nodes to determine whether or not execution should be aborted in the event of an applicabilty failure. Outputs of applicability tests are typically routed to a special "Enable" input of other nodes. When an applicability test disables another node in the graph, all computations downstream of the test are aborted.

Initial tests of our system for controlling application of models have demonstrated two types of benefits. When provided with inputs that would previously have caused models to return erroneous results, our system returns an error condition indicating that the model is not applicable to the current input. The system thus avoids misleading the user with erroneous results. In addition, our system informs the user of which applicability tests failed and thus makes him aware of the reason the model does not apply to the current input. In the future, we plan develop tools for using such diagnostic information to support revision of scientific models to change or extend their ranges of applicability.

Summary

The model development toolbox and record is intended to support a variety of activities that occur in the course of developing scientific computation models. These activities include construction and testing of new models; controlled application of models to specific problems, and revision of models to handle new situations. The system is also expected to promote rapid development of new scientific computational models, more reliable use of scientific models among computational scientists; wider sharing of scientific models within communities of scientists; and deeper understanding among scientists of the assumptions and modeling techniques incorporated in the models they use. A more detailed description of this research is found in (Ellman 1993).

Acknowledgments

The research presented in this document is supported by NASA Grant NCC-2-802. It has also benefited from discussions with Richard Keller, Saul Amarel, Haym Hirsh, Lou Steinberg, Andrew Gelsey, John Keane and Mark Schwabacher.

References

Balzer, R. 1985. A 15 year perspective on automatic programming. *IEEE Transactions on Software Engineering* SE-11(11):1257–1268.

Ellman, T. 1993. A toolbox and record for scientific model development. In *Proceedings of the Symposium on Space Operations and Research.*

Keller, R., and Rimon, M. 1992. A knowledge-based software development environment for scientific model-building. In *Proceedings of the Seventh Knowledge-Based Software-Engineering Conference.*

Mostow, J. 1989. Design by derivational analogy: Issues in the automated replay of design plans. *Artificial Intelligence* 40:119 – 184.

N95- 23717

A New Generation of Intelligent Trainable Tools for Analyzing Large Scientific Image Databases

Usama M. Fayyad, Padhraic Smyth, and David J. Atkinson
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Tel: 818-306-6197 Fax: 818-306-6912
Email: Fayyad@aig.jpl.nasa.gov, pjs@galway.jpl.nasa.gov, Atkinson@isd.jpl.nasa.gov

KEY WORDS AND PHRASES
Machine learning, Automated Data Analysis, Intelligent Analysis Tools, Large Image Databases.

1. INTRODUCTION

In a variety of scientific disciplines two-dimensional digital image data is now relied on as a basic component of routine scientific investigation. The proliferation of image acquisition hardware such as multi-spectral remote-sensing platforms, medical imaging sensors, and high-resolution cameras has led to the widespread use of image data in fields such as atmospheric studies, planetary geology, ecology, agriculture, glaciology, forestry, astronomy, diagnostic medicine, to name but a few. Across all of these disciplines there is a common factor: the image data for each application, whether it be a Landsat image or an ultrasound scan, is but a means to an end in the sense that the investigator is only interested in using the image data to infer some conclusion about the physical properties of the target being imaged. In this sense, the image data serves as an intermediate representation to facilitate the scientific process of inferring a conclusion from the available evidence.

In the past, in planetary science for example, image databases were analyzed in a careful manual manner and much investigative work was carried out using hard copy photographs. However, due to the sheer enormity of the image databases currently being acquired, simple manual cataloging is no longer a practical consideration if all of the available data is to be utilized.

A currently familiar pattern in the remote-sensing and astronomy communities is the following: a new image data set becomes available but the size of the data set precludes the use of simple manual methods for exploration. Scientists are beginning to express a need for automated tools which can assist them in navigating through large sets of images. A commonly expressed wish is the following: "is there a tool where I could just point at an object on the screen (or even draw a caricature of it) and then have the algorithm find similar items in the database?"

Note that in this paper the type of problem being addressed differs from the types of problems typically addressed by classical work in machine vision. Machine vision work has focused primarily on image understanding, parsing, and segmentation, with a particular emphasis on detecting and analyzing *human-made* objects in the scene of interest. The focus of this paper is on the detection of *natural*, as opposed to *human-made*, objects. The distinction is important because, in the context of image analysis, natural objects tend to possess much greater variability in appearance than human-made objects. Hence, we shall focus primarily on the use of algorithms that "learn by example" as the basis for image exploration. The "learn by example" approach is potentially more generally applicable compared to model-based vision methods since domain scientists find it relatively easier to provide examples of what they are searching for versus describing a model.

1.1 TWO ILLUSTRATIVE CASE STUDIES

Using ongoing JPL projects as case studies, this paper is intended to provide motivation for the need to develop automated image analysis techniques as well as report on our initial success in the application of pattern recognition and machine learning technology to the general problem of image database exploration. The first project, the Sky Image Cataloging and Analysis Tool (SKICAT), represents an already successful application of decision-tree learning to classification in the context of a well-understood image analysis problem in astronomy. The second project represents ongoing work which targets a more ambitious problem of dealing with domains where the basic image processing itself is not straightforward: The JPL Adaptive Recognition Tool (JARtool) is being developed for use by planetary geologists on the automated analysis of the Magellan Synthetic Aperture Radar (SAR) images of the planet Venus.

2. SKICAT: AUTOMATED SKY SURVEY CATALOGING

The first case study consists of an application of machine learning techniques to the automation of the task of cataloging sky objects in digitized sky images. SKICAT has been developed for use on the images resulting from the 2nd Palomar Observatory Sky Survey (POSS-II) conducted by the California Institute of Technology (Caltech). The photographic plates collected from the survey are being digitized at the Space Telescope Science Institute (STScI). This process will result in about 3,000 digital images of roughly 23,000 x 23,000 pixels[1] each. The survey consists of over 3 terabytes of data containing on the order of 10^7 galaxies, 10^9 stars, and 10^5 quasars.

The first step in analyzing the results of a sky survey is to identify, measure, and catalog the detected objects in the image into their respective classes. Once the objects have been classified, further scientific analysis can proceed. For example, the resulting catalog may be used to test models of the formation of large-scale structure in the universe, probe galactic structure from star counts, perform automatic identification of radio or infrared sources, and so forth. The task of reducing the images to catalog entries is a laborious time-consuming process. A manual approach to constructing the catalog implies that many scientists need to expend large amounts of time on a visually intensive task that may involve significant subjective judgment. The goal of our project is to automate the process, thus alleviating the burden of cataloging objects for the scientist and providing a more objective methodology for reducing the data sets. Another goal of this work is to classify objects whose intensity (isophotal magnitude) is too faint for recognition by inspection, hence requiring an automated classification procedure. Faint objects constitute the majority of objects on any given plate. We target the classification of objects that are at least one magnitude fainter than objects classified in previous surveys using comparable photographic material.

The learning algorithms used in SKICAT are the GID3* [4] and O-Btree [5] decision tree generation algorithms. In order to overcome limitations inherent in a decision-tree approach, we use the RULER [6] system for deriving statistically cross-validated classification rules from multiple (typically > 10) decision trees. The details of the learning algorithms are beyond the scope of this paper and are therefore not covered here. For details of how rules are generated from multiple decision trees, and for other algorithmic details, the reader is referred to [6,7].

A manual approach to classifying sky objects in the images is infeasible. Existing computational methods for processing the images will preclude the identification of the majority of objects in each image since they are at levels too faint (the resolution is too low) for traditional recognition algorithms or even methods based on manual inspection or analysis. Low-level image processing and object separation are performed by the public domain FOCAS image processing software developed at Bell Labs [11,14]. In addition to detecting the objects in each image, FOCAS also produces basic attributes describing each object. These attributes are standard in the field of astronomy and represent commonly measured quantities such as area, magnitude, several statistical moments of core intensity, ellipticity, and so forth. Additional normalized attributes were measured later to achieve accuracy requirements and provide stable performance over different plates. In total, 40 attributes are measured by SKICAT for each detected object.

2.1 FAINT SKY OBJECT CLASSIFICATION

In addition to the scanned photographic plates, we have access to CCD images that span several small regions in some of the plates. The main advantage of a CCD image is higher resolution and signal-to-noise ratio at fainter levels. Hence, many of the objects that are too faint to be classified by inspection of a

---

[1] Each pixel consists of 16 bits and represents the intensity in one of three colors.

photographic plate, are easily classifiable in the corresponding CCD image (if available). We make use of the CCD images in two very important ways: CCD images enable us to obtain class labels for faint objects in the photographic plates, and CCD images provide us with the means to reliably evaluate the accuracy of the classifiers obtained from the decision-tree learning algorithms.

In order to produce a classifier that classifies faint objects correctly, the learning algorithm needs training data consisting of faint objects labeled with the appropriate class. The class label is therefore obtained by examining the CCD frames. Once trained on properly labeled objects, the learning algorithm produces a classifier that is capable of properly classifying objects based on the values of the attributes provided by FOCAS. Hence, in principle, the classifier will be able to classify objects in the photographic image that are simply too faint for an astronomer to classify by inspection of the survey images. Using the class labels, the learning algorithms are basically being used to solve the more difficult problem of separating the classes in the multi-dimensional space defined by the set of attributes derived via image processing. This method allows us to classify objects at least one magnitude fainter than objects classified in photographic sky surveys to date.

2.2 RESULTS

We were able to achieve a stable classification accuracy of 94% in classification of sky objects into four classes: *star, galaxy, star-with-fuzz*, and *artifacts* [15]. The latter class represents non-sky objects in the photographs due to film problems, satellite or airplane traces, or other problems. It is noteworthy that using the learning algorithms, we are able to classify objects that are at least one magnitude fainter than objects classified in previous comparable surveys. The SKICAT system is expected to speed up catalog generation by one to two orders of magnitude over traditional manual approaches to cataloging. This should significantly reduce the cost of cataloging survey images by the equivalent of tens of astronomer workyears. In addition, SKICAT classifies objects that are at least one magnitude fainter than objects cataloged in previous surveys. We have exceeded our initial accuracy target of 90%. This level of accuracy is required for the data to be useful in testing or refuting theories on the formation of large structure in the universe and on other phenomena of interest to astronomers.

The catalog generated by SKICAT will eventually contain about a billion entries representing hundreds of millions of sky objects. For the first survey (POSS-I) conducted over 4 decades ago, without the availability of an automated tool like SKICAT, only a small percentage of the data was used and only specific areas of interest were studied. In contrast, we are targeting a comprehensive sky catalog that will be available on-line for the use of the scientific community. Because we can classify objects that are one magnitude fainter, the resulting catalog will be significantly richer in content, containing three times as many sky objects as would have been possible without using SKICAT.

3. JARTOOL: VOLCANO DETECTION IN MAGELLAN-VENUS DATA

The Magellan-Venus data set constitutes an example of the large volumes of data that today's instruments can collect, providing more detail of Venus than was previously available from Pioneer Venus, Venera 15/16, or ground-based radar observations put together [13]. Venus is an extremely volcanic planet (volcanoes are by far the single most visible geologic feature in the Magellan data set); hence, the study of basic volcanic processes is essential to a basic understanding of the geologic evolution of the planet [10]. Central to volcanic studies is the cataloging of each volcano location and its size and characteristics. We are initially targeting the automated detection of the "small-shield" volcanoes (less than 15 km in diameter) that constitute the most abundant visible geologic feature [8] in the more than 30,000 SAR images of the surface of Venus. It is estimated, based on extrapolating from previous studies and knowledge of the underlying geologic processes, that there should be on the order of 10^6 of these volcanoes visible in the Magellan data [1,10].

Identifying and studying these volcanoes is fundamental to a proper understanding of the geologic evolution of Venus. However, locating and parameterizing them in a manual manner is forbiddingly time-consuming. Hence, we have undertaken the development of techniques to partially automate this task. The primary constraints for this particular problem are that the method must be reasonably robust and fast.

3.1 THE APPROACH

There has been little prior work on detecting naturally occurring objects in remotely-sensed images. Most pattern recognition algorithms are geared towards detecting straight edges or large changes in texture or reflectivity. While this works well for detecting *human-made* objects, approaches such as edge detection and Hough transforms deal poorly with the variability and noise present in typical remotely sensed data [3,12].

We are developing a system that consists of three distinct components: focus of attention, feature extraction, and classification learning. Figure 1 gives a block diagram of the approach. The focus of attention component is designed primarily for computational efficiency. Its function is to quickly scan an input image and roughly determine regions of interest (regions potentially containing objects similar to those specified by the scientist). Given a set of detected regions of interest, the remaining task is to discriminate between the volcanoes and false alarms. A current focus of the research is to find a useful feature-representation space --- although nearest neighbor classifiers can provide reasonably accurate results, a representation based purely on pixels will tend to generalize poorly. For the purposes of incorporating prior knowledge, the ideal feature set would be expressed in the form of expected sizes, shapes, and relative geometry of slopes and pits, namely, the same features as used by the scientists to describe the volcanoes. However, due to the low signal-to-noise ratio of the image, it is quite difficult to gain accurate measurements of these features, effectively precluding their use at present. The current focus of our work is on a method which automatically derives robust feature representations. The current method is based on performing a singular value decomposition of training images (15 x 15 pixel vectors centered at volcanoes) to find the eigenvectors of the data. In turn, the dominant eigenvectors (principal components) provide the means to translate pixels into a low-dimensional feature space. In the latter, classification learning is used to distinguish between true volcanoes and focus of attention (FOA) false alarms.

3.2 STATUS AND PRELIMINARY RESULTS

We have constructed several training sets using 75-m/pixel resolution images labeled by the collaborating geologists at Brown University to get an initial estimate of the performance of the system. The FOA component typically detects more than 80% of all the volcanoes, while generating 5-6 times as many false alarms. Using features derived from both segmentation and principal component methods [2] has resulted in accuracies of the order of 85% of the volcanoes detected by FOA. It is important to clarify that these are initial results and with further effort we hope to be able to significantly improve the accuracy. Demonstrating the general applicability of this approach to the detection of other Venusian features as well as images from other missions will be the next step. So far the emphasis has been placed mainly on developing the computer tools to allow scientists to browse through images and produce training data sets (as well as partial catalogs) within a single integrated workstation environment.

4. CONCLUDING REMARKS

Natural object detection and characterization in large image databases is a generic task which poses many challenges to current scientific analysis tasks. The SKICAT and Magellan SAR projects are typical examples of the types of large-scale image database applications which will become increasingly common --- for example, the NASA Earth Observing System Synthetic Aperture Radar (EOS SAR) satellite will generate on the order of 50 GBytes of remote sensing data per hour when operational. In order for scientists to be able to effectively utilize these extremely large amounts of data, basic image database navigation tools will be essential. Our existing JPL projects have so far demonstrated that efficient and accurate tools for natural object detection are a realistic goal provided there is strong prior knowledge about how pixels can be turned into features and from there to class categories. With the astronomy problem there was sufficient strong knowledge for this to be the case: with the volcano data, the knowledge is much less precise and consequently the design of effective object detection tools is considerably more difficult.

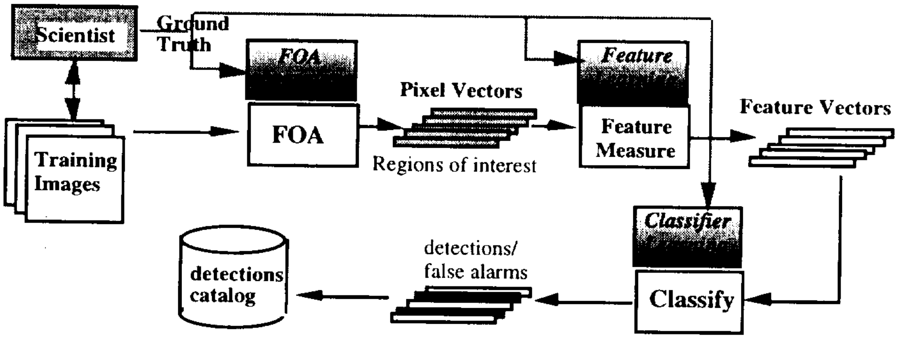


Figure 1. Block Diagram of the JARtool System

We believe that trainable tools for object recognition/cataloging will soon become a necessity. The alternative of writing purpose specific programs customized to individual problems is simply unrealistic and too constrained. The alternative of manual analysis by the scientists is no longer feasible due to the large database sizes.

ACKNOWLEDGMENTS

SKICAT work is a collaboration between Fayyad (JPL) and N. Weir and S.G. Djorgovski of Caltech's Astronomy Department. JARtool work is a collaboration between Fayyad and Smyth (JPL) and M.C. Burl and P. Perona (Electrical Engineering Department, Caltech); the domain scientists are J. Aubele and L. Crumpler (Dept. of Geological Sciences, Brown University). The research described in this paper was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] J. C. Aubelle and E. N. Slyuta, ``Small domes on Venus: characteristics and origins," in Earth, Moon and Planets, 50/51, 493--532, 1990.

[2] M.C. Burl, U.M. Fayyad, Perona, P., Smyth, P. and Burl, M.P. (1994) "Automating the Hunt for Volcanoes on Venus", *Proc. of The Computer Vision and Pattern Recognition Conference (CVPR'94),* IEEE Press.

[3] A.M. Cross, *Int. J. Remote Sensing*, 9,no.9, 1519-1528, 1988.

[4] U.M. Fayyad (1991). *On the Induction of Decision Trees for Multiple Concept Learning*. Ph.D. Dissertation, The University of Michigan.

[5] U.M. Fayyad and K.B. Irani (1992) "The attribute selection problem in decision tree generation." *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI-92*. San Jose, CA.

[6] U. Fayyad, N. Weir, and S.G. Djorgovski (1993). "SKICAT: a machine learning system for automated cataloging of large scale sky surveys." *Proc. of 10th Int. Conf. on Machine Learning*, Amherst, MA.

[7] U.Fayyad, P.Smyth N. Weir, & S.Djorgovski (1994). "Automated Analysis and Exploration of Image Databases", *J.Intelligent Info. Systems* (in press).

[8] J. E. Guest et al. (1992) *Journal Geophys. Res.*, 97, E10, 15949.

[9] J. W. Head et al. (1991) ``Venus volcanic centers and their environmental settings: recent data from Magellan," EOS 72, p.175, *American Geophysical Union Spring meeting abstracts*.

[10] J. W. Head et al. (1992) *Journal Geophysical Res.*, 97, E8, 13,153-13,197.

[11] J.Jarvis and A.Tyson (1981) *Astronomical Journal* 86:41.

[12] S. Quegan et al, (1988) *Trans. R. Soc. London*, A 324, 409-421.

[13] Science, special issue on Magellan data, April 12, 1991.

[14] Valdes (1982) Instrumentation in Astronomy IV, SPIE vol. 331, no. 465.

[15] N.Weir, S.G.Djorgovski, U.M.Fayyad, et al (1992) "SKICAT: A system for the scientific analysis of the Palomar-STScI Digital Sky Survey." *Proc. Astronomy from Large databases II*, p. 509, Munich, Germany:Euro. Southern Observatory.

Space Science Experimentation Automation and Support

N95-23718

Richard J. Frainier, Nicolas Groleau, Jeff C. Shapiro
Recom Technologies, Inc.
Artificial Intelligence Research Branch, NASA Ames Research Center m/s 269-2
Moffett Field, CA 94035-1000, USA
vox (415) 604-4776; fax (415) 604-3594
frainier@ptolemy.arc.nasa.gov

KEY WORDS AND PHRASES

Advisory Systems, Artificial Intelligence, Knowledge-Based Systems, Space Science

INTRODUCTION

This paper outlines recent work done at the NASA Ames Artificial Intelligence Research Laboratory on automation and support of science experiments on the US Space Shuttle in low earth orbit. Three approaches to increasing the science return of these experiments using emerging automation technologies are described: remote control (telescience), science advisors for astronaut operators, and fully autonomous experiments. The capabilities and limitations of these approaches are reviewed. Cost-effective automation often takes advantage of the presence of crew, regarding them as an essential component of the experiment system. Humans suffer from limitations as part of that system. However, humans have unsurpassed general purpose intelligence (common sense reasoning) and abilities as general purpose manipulators.

The US has had access to space for science experimentation for over three decades. Although the US has ventured as far as the surface of the moon with crewed vehicles, most work has been done in low earth orbit. Crewed mission series over the last two decades include Skylab, the Space Shuttle (with Spacelab and the aft flight deck lockers), and Shuttle/MIR, with Space Station Alpha anticipated by the next decade. Still, access to space for science experimentation has been sporadic. Putting people into space is a costly undertaking. Devising and building experiments suitable for use in flight is costly. Total mission payload mass and volume are carefully managed resources. Scarcer yet on Space Shuttle missions, is crew (experiment operator) time.

There are several aspects to the issue of limited crew time. First, missions have a fairly short duration. Second, the crew of a particular mission is usually identified only about a year prior to launch, leaving limited training time for a set of experiments often outside the range of expertise of a given crew member. Third, many of the Shuttle-hosted Spacelab missions are life-science investigations which often use crew as subjects. When experiment subjects, they are unavailable as experiment operators. Finally, space is a difficult working environment for humans. Crew typically suffer some disorientation in space, especially early in a mission. The disorientation limits the complexity of the tasks they can accomplish. This fourth issue is managed by scripting and rehearsing on-orbit activities. Deviations from the script are discouraged, and fixed experiment protocols are used. This major constraint severely limits the ability of an earth-bound scientist to change the course of an experiment even when the data and current situation clearly indicate that it would be scientifically more valuable to do so. Worse yet, it is sometimes the case that an experiment apparatus is damaged or is otherwise not producing valid data. The faults need to be identified and repaired. There is often an extended interval to identify, communicate, and execute needed on-orbit experiment apparatus repairs.

There are other significant features of the Spacelab task environment. One is that there can be several experiments being conducted concurrently, with different demands on uplink and downlink data transmission capability. In particular, video data may not be continuously available during an experiment session. Further, there is not continuous signal availability during the orbit of the Space Shuttle. "Loss of Signal" (LOS) occurs for perhaps 15% of a given orbit.

As a result, automation is viewed as a way of getting better return for the money invested

228

in space science experimentation. We have identified three (and tested two) conceptual approaches to employing advanced automation techniques: (1) telescience, or remote operation of experiment using a command uplink, (2) laptop-based science advice for the astronaut experiment operator, and (3) a fully autonomous science experiment system. Each of these approaches is presented in turn.

TELESCIENCE

By "telescience" we mean that the space-based experiment is instrumented sufficiently well to permit a ground-based investigator understand the experiment's progress in "near real-time" and to directly control it using command uplink. The investigator may still depend on the crew to deploy and set-up the experiment apparatus. However, the investigator has direct control of experiment parameters and is controlling the execution of the experiment protocol. The key issue is real-time datalink access. If available, it is feasible to perform reactive scientific experimentation using this approach to automate certain types of investigations.

The telescience approach to automation was used to support the Superfluid Helium On-Orbit Transfer (SHOOT) experiment. This experiment investigated physical processes associated with superfluid helium flow in microgravity. For the STS-57-hosted experiment, a ground based Macintosh computer was used to control the conduct of the experiment. A Command and Monitoring System (CMS) was developed at the NASA Ames Artificial Intelligence Research Laboratory [1]. The CMS was used for all phases of the investigation's operation from hardware test and system integration, through launch pad servicing and telecontrol of the flight experiment during the mission, to post-flight data analysis. This paper highlights the CMS telecontrol of the flight experiment during the mission.

The CMS used a modern window-oriented point-and-click interface replacing the previously typical line-oriented keyboard interfaces. Key features of the system included a macro facility, flexible data displays, and scientific data analysis.

A set of low-level commands were devised to control the SHOOT experiment hardware. The commands control valves, voltages, and establish setpoints. This is not a useful level of abstraction for the experiment's investigator. The CMS macro facility was a pre-tested set of commands constructed from the experiment hardware's low-level command set. For example, the macro "transfer port-starboard for 10 minutes at 20 volts" would call the correct sequence of a dozen low-level commands to configure valves and set a helium pump's voltage level, timer, and relay. These macros facilitated rapid and accurate control of the experiment protocol during the flight. Macros were sent directly as immediate commands. They were also called up for display and modification before execution. Editing typically involved parameter (timing or voltage) adjustments to a pre-tested macro. The interactive displays were important in assisting users through the process, especially when the experiment was not behaving as anticipated.

The CMS also offered flexible data displays that could be manipulated by the operator, as opposed to previous "canned" displays offering only fixed views of the data on a display screen. Further, CMS offered the ability to dynamically change limits associated with telemetry out-of-limit checks. Some real-time scientific data analysis was performed in the CMS: a fluid-level adjustment calculation could be performed in real-time for the operator. This feature was crucial to the success of many on-orbit helium mass gauging operations, even though it had not often been required for pre-flight laboratory helium mass gaugings.

Further work needs to be done on displays of "aged" data. It is important to indicate both the importance of the data (nominal, borderline, out-of-limit) and its currency (recent, adequate, "stale").

SCIENCE ADVISORS

There is a wide assortment of experiments performed on many Spacelab missions. According to the Marshall Space Flight Center Payload Projects Office [2], there were 20 experiments performed during SLS-1 in June, 1991. There were 78 experiments performed during D-2 in April, 1993. This is a far greater number than a 4-person crew can master in the year between assignment to a mission and lift-off. Thus, with Mission Specialists working in Spacelab now, a generalist is performing a specialist's expert task. The expert is at a remote location (the ground), and is not in ready contact with the generalist during experiment execution. A possible solution is to make a science advisor available to the astronaut conducting the experiment in space. In this case, monitoring and analysis done by the ground-based investigator is replicated on a laptop computer connected to the experiment apparatus. The astronaut and the advisor work together to understand the progress of the experiment (Figure 1). These systems can empower the user by providing a readily accessible source of expert advice. This approach is not limited to space: it can be applied to any science or technical analysis task where an operator is gathering data and needs to make high-value decisions in real-time without ready access to the technical expert.

The Principal Investigator in a Box (PI-in-a-Box) system was used to support the Rotating Dome Experiment during the Spacelab Life Sciences Mission hosted by STS-58 [3]. It was developed at the NASA Ames Artificial Intelligence Research Laboratory and had direct access to all 5 of the experiment's analog data channels. The Macintosh PowerBook-based system provided support for the key activities of reactive experimental science: assuring sensor values are data, analyzing those data against the investigator's model of the phenomenon under study, and suggesting high-value departures from the pre-planned protocol in reaction to the results of the analysis. The astronaut is in overall control of the investigation, and can act with confidence using the advice of the surrogate scientist. In flight use, the system demonstrated superior data integrity assurance, data analysis, and model validation (Figure 1). The system also demonstrated graceful degradation when training recall problems were encountered. The ability to use "degraded operation" modes with simpler interfaces was cited by the astronauts as a key success of the system. The diagnosis and troubleshooting facility did not get exercised, as there were no equipment problems encountered with the experiment inflight. The protocol management facility was used with mixed success: some operators used it to modify protocols without incident while others had difficulty with the astronaut-computer interface.

A major issue that arose with the use of PI-in-a-Box was the willingness of the astronauts to operate as reactive scientists. The current culture surrounding Spacelab operations is tuned to set up the experiment and ensure data of reasonable quality is being archived on the ground for later detailed analysis. Thus, in many cases, neither the crew nor the investigators on the ground monitoring the progress of the experiment are reacting in real-time to change the preplanned, scripted course of the experiment. With both MIR and Space Station Alpha, this style will no longer be adequate. Presently, MIR is in contact with the ground for only about 50% of an orbit. This makes telescience-based control difficult. Furthermore, some experiments are sent up to a resident MIR crew that has had no training at all on them. Even if the communication connectivity is improved, there is a clear role for science advisor systems and fully automated experiment systems in this environment.

The PI-in-a-Box experience indicates that the astronaut-computer interface needs to be made as simple and "intuitive" as possible. Mastery of computer system skills in ground simulations does not guarantee successful recall in flight.

AUTONOMOUS EXPERIMENTS

As NASA moves to MIR and Space Station Missions, it seems likely that available air-to-ground bandwidth and crew time will be exhausted before other resources such as loftable mass and volume, and available power and thermal rejection capacity. In this case, experiments that can be fully automated can be run using "leftover" resources. NASA has flown "Get Away Special" containerized experiments in the past with mixed results. These experiments are preplanned and offer no opportunity for reaction to the data. The addition of intelligence would allow a much greater range of investigation by dynamically adjusting experiment coverage parameters based on intermediate results.

A semi-autonomous system, called AfDex, was developed at the NASA Ames Artificial Intelligence Research Laboratory for control of the SHOOT experiment (mentioned above) from the Shuttle aft flight deck. AfDex successfully executed several experiment steps autonomously: the system represents a good first step at fully autonomous experiment control. AfDex has helped to define the characteristics of experiments that would benefit from this approach. It appears that some of the plasma physics and materials science investigations performed on previous Spacelab missions could have been engineered for autonomous operation.

HYBRID APPROACHES

Integration of these three approaches may result in a system superior to one based on any single approach. In Space Life Sciences for example, routine maintenance of specimen viability is best achieved through autonomous control. However, astronaut intervention may be needed for detailed problem diagnosis or for complex visual evaluation of samples that have been treated with a fixing agent. Finally, control can be exerted with more powerful ground-based workstations in real-time at critical phases of the scientific evaluation. Ground-based workstations could also establish high-level experiment goals and timelines for experiment events occurring later in the mission. These goals and timelines could be uplinked to the on-board scientific advisor during periods of low air-ground channel usage (during crew sleep periods).

A hybrid advisory system proposed by one of the authors would monitor the experiment and mediate decisions made by the automated system, the astronaut, and ground personnel. At each choice point in the experiment, the system will communicate the need for a decision to (local and/or remote) human operators and in parallel, attempt to resolve the question autonomously. Unless human intervention cancels an on-board computation, the automated scientific advisor will notify the operator(s) of its conclusion. That conclusion is implemented after a time-out period dependent upon the criticality and time-sensitivity of the decision. This mechanism ensures that, when available, a human operator can question or override the automatic science advisor.

CONCLUSIONS

Presently, there are severe restrictions on the ability of crew to operate as reactive scientists in space laboratories. Advanced automation techniques have demonstrated a level of maturity that makes their inclusion in future science missions highly desirable. The approach selected for a given experiment depends on that experiment's characteristics, in particular the space laboratory resources needed to conduct the investigation and the opportunity offered by the investigator's model to conduct reactive science.

BIBLIOGRAPHY

[1] Shapiro, J. C.; and Villere, G. L., 1994. A Description of the SHOOT Command & Monitoring System. *Cryogenics* 34(5): 407-413.

[2] Torr, M. R., 1994. Scientific Achievements of 10 Years of Spacelab: An Overview of the Missions. In *Proceedings of the 32nd Aerospace Sciences Meeting & Exhibit,* AIAA 94-0340. Washington, DC: American Institute of Aeronautics and Astronautics.

[3] Frainier R. J.; Groleau, N.; Hazelton, L. R.; Colombano, S. P.; Compton, M. C.; Statler, I. C.; Szolovits, P.; and Young, L. R., 1994. PI-in-a-Box: a knowledge-based system for space science experimentation. *AI Magazine* 15(1):39-51.

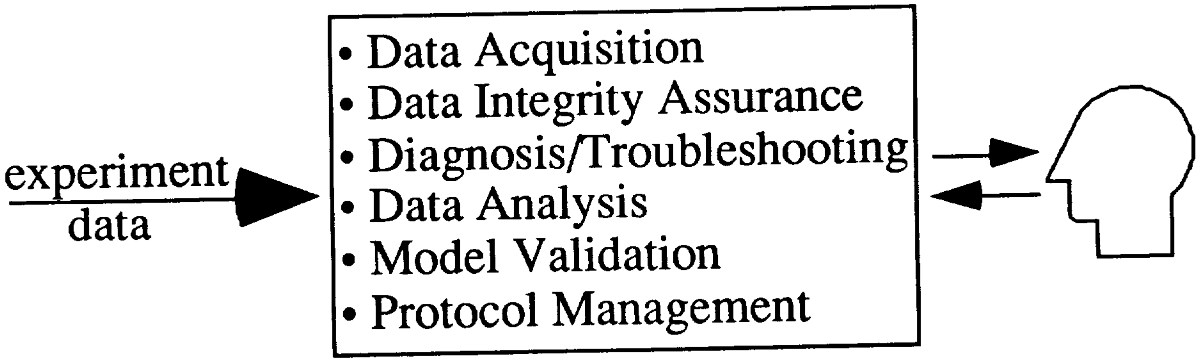


Figure 1. Advisor roles

SESSION SC

Spacecraft Control Systems

N95-23719

Autonomous Spacecraft Executive and Its Application to Rendezvous and Docking

Fuminobu Komura*, Masatoshi Furuya*, Toshiro Sasaki*
Robert L. Anderson, Roy K. Tsugawa****
*Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji, Asao-ku, Kawasaki 215 Japan
Phone:+81-44-966-9111, Fax:+81-44-966-6823
**Civil and International Systems Division,
TRW Space and Electronics Group
One Space Park, Redondo Beach, CA 90278 USA
Phone:(310)812-2608, Fax:(310)812-7296

KEY WORDS AND PHRASES

autonomy, mission planning, rendezvous and docking

ABSTRACT

Autonomy is needed for future spacecraft to solve the problems of human operator overload and transmission delay. This paper describes the autonomous spacecraft executive for rendezvous and docking. It is an onboard expert system and has decision making capability for mission planning of nominal and contingency cases. The executive has been developed and verified using a hardware motion based simulator.

INTRODUCTION

Research activities have been done to develop autonomous space systems.[1] Spacecraft autonomy is needed to avoid the overload of human operators and to overcome the delay or loss of command link. Spacecraft rendezvous and docking is a typical mission which needs autonomous operations.[2][3]

Spacecraft autonomy is attained by realizing mission planning and contingency management functions in onboard computers. The product of mission planning or contingency management is a sequence of commands to the conventional control systems of the spacecraft.[3]

AUTONOMOUS SPACECRAFT EXECUTIVE

Fig. 1 shows the architecture of an autonomous spacecraft.[3] The Autonomous Spacecraft Executive is an expert system implemented on an onboard computer that makes decisions needed for the spacecraft mission. The Executive is interfaced to the GN&C (Guidance, Navigation & Control system) and the SM (System Manager), and receives state and status from the GN&C and SM, and generates control commands and sends them to the GN&C and SM.

This architecture has the following characteristics.

(1) It is a universal modular architecture and is applicable to any spacecraft.

(2) The modules that receive the control commands don't need to know whether the commands are sent from the Executive or from a ground controller.

(3) The Executive has a vehicle dynamics simulator as a mission planning tool.

EXECUTIVE FOR AUTONOMOUS RENDEZVOUS AND DOCKING

Requirements

We consider rendezvous and docking missions where the target vehicle is a cooperative passive vehicle which is holding its attitude in a LVLH (local vertical - local horizontal) frame and has a receiver for differential GPS and reflectors on the target for a docking sensor on the chaser vehicle. The active chaser vehicle has the architecture of Fig. 1.

To complete a rendezvous and docking mission

234

many decisions must be made. The most essential decision is to plan a flight path or a velocity profile to attain the mission goals under safety, timing and consumables constraints. The plans must be made for both nominal and contingency situations. They vary depending on the phases of flight, i.e., approach from a parking orbit, proximity, dock, separation, etc. To accomplish the rendezvous and docking mission autonomously the Executive is required to create these flight plans.[3][4]

For the final stage of proximity e.g. from 1000 ft to 0 ft, the requirements for the Executive will be as follows.

modes:
- nominal approach plan
- contingency: loss of GPS lock or loss of proximity sensor lock
    - replan and try again, or
    - abort the mission

constraints:
- safe velocity profile
- safe approach corridor
- time of arrival (for lighting control, crew schedule, communications availability, etc.)

Executive Functions

The Executive has the following functions to meet the above requirements.

(1) input
- mission goals from the ground controller
- spacecraft state and status from the GN&C and SM

(2) monitor
- status of sensors
- position and velocity of chaser relative to target:

determine whether within control volume and safety limits, and if mission requirements are attainable

(3) plan

Depending on the output of (1) and (2), either of the following plans is generated from the rules.
- nominal approach based on the time of arrival requirements
- contingency plan based on the spacecraft state and status
- abort

(4) output
- control commands to the GN&C and SM

Monitoring and Planning Rules

The Executive functions of monitoring and planning can be realized by a set of decision rules which are expressed in the following form.

```
IF
      (current_control_state)(relative_position)
      (vehicle_status)(mission_requirements)
THEN
      (create new plan
        or continue
        or create contingency plan
        or station keep
        or back away
        or abort)
```

The IF part represents the monitoring, and the THEN part represents the planning. By these rules the control state of the vehicle is determined. Fig.2 shows a state transition diagram for the proximity operation.

The generation of the nominal plan "create new plan" consists of the following processes.

1. Design velocity profile for each phase

The proximity operations consist of a number of phases separated by station keeping positions. For example, station keeping positions are set at -1000ft, -300 ft, -35 ft, and -20 ft. They are needed for changing the vehicle control modes and for adjusting the arrival time at the target. A transfer is usually used from -1000 ft to -300 ft to save fuel, and an LVLH approach is preferable within -300 ft for safety. The velocity profile is computed by using mission planning tools, e.g. a vehicle dynamics simulator.

2. Select the earliest possible docking window
3. Allocate duration for each station keeping position
4. Abort if no window is attainable

With these rules the Executive can make decisions needed for the nominal and contingency operations in the proximity stage. Other set of rules are used for the autonomous operations in other stages.

VERIFICATION TESTS USING A HARDWARE SIMULATOR

Simulator Configuration

The configuration of the verification test facility is shown in Fig. 3. The Executive was implemented on a PC and it was connected via an RS422 link to the 6 DOF (Degree of Freedom) dynamics simulator and the GN&C system installed on a VAX at the NASA Marshall Space Flight Center astrionics laboratory. The mockup of a chaser vehicle with the actual VGS (Video Guidance Sensor) was mounted on the floor and the VGS was connected to the 6 DOF simulator. The GPS was simulated in the 6 DOF simulator. The DOTS (Dynamic Overhead Target Simulator) on a VAX moves a crane arm based on the output state of the 6 DOF simulator. The mockup of a target vehicle is attached on the arm end. The reflectors for the VGS are attached to the back face of the target vehicle.

With this configuration the motion of the target vehicle relative to the chaser vehicle can be simulated. The range of simulated flight covers the final approach from 50 ft to 0 ft station keeping position where the three point docking mechanism can be activated to complete the docking.

In addition to the simulations using the above setup, the software simulations were done using only the Executive on the PC and the VAX simulator. The range of flight in these software simulations are from 1000 ft to 0 ft.

Test Results

Test runs of the chaser approach were made both in hardware simulations and software simulations by changing the initial conditions and the docking windows. The contingencies were brought about by either physically disabling the VGS hardware or simulating the loss of GPS lock at an arbitrary time during approach. In all of the cases it was verified that the Executive can start the mission replanning and generate a new approach or abort profile based on ground supplied mission rules.

Fig. 4 shows an example test result of a case where VGS lock was lost and regained during the final approach. While station keeping at x = -35 ft, the Executive generated a flight plan, PLAN1, for the nominal approach. The plan drives the chaser first to the next station keeping point at x = -20 ft, and the vehicle stays for the period needed to check the vehicle status, and the vehicle resumes the approach to x = 0 ft to meet the docking window #2. But during the approach the VGS lock was lost at t = 190 sec. When the Executive detected the loss it generated the contingency plan, PLAN2. The plan forces the vehicle to back up to the safe station keeping position at x = -35 ft, and let it wait until sensor lock is regained. Because the lock is regained during this back up, the Executive generated a new plan, PLAN3, similar to PLAN1, to resume a nominal approach, but this time the earliest window available is window #3. Tables 1. and 2. show the control commands for PLAN1 and PLAN2.

CONCLUSIONS

The autonomous spacecraft executive has been developed for autonomous on board mission planning for rendezvous and docking. Its decision making capability for nominal and contingency cases has been verified by simulations.

The executive is also applicable to other spacecraft missions which need autonomous onboard decision making.

REFERENCES

[1] M.L. Montemerlo, The AI Programs at the National Aeronautics & Space Administration, *AI Magazine*, Winter 1992.

[2] H. Nakamura, et al., Study of Autonomous Rendezvous and Docking, *J. Space Technology and Science*, vol.4, No. 2, 1989.

[3] R.L. Anderson, et al., Demonstration of Automated Proximity and Docking Technology, In *Proceedings of AIAA Space Programs and Technologies Conference*, March 1992, Huntsville, AL.

[4] M. Furuya, et al., A Study on High Level Decision Making Function in Onboard Computer System for Rendezvous and Docking, In *Proceedings of the 37th Space Sciences and Technology Conference*, October 1993, Kita-Kyushu, Japan.

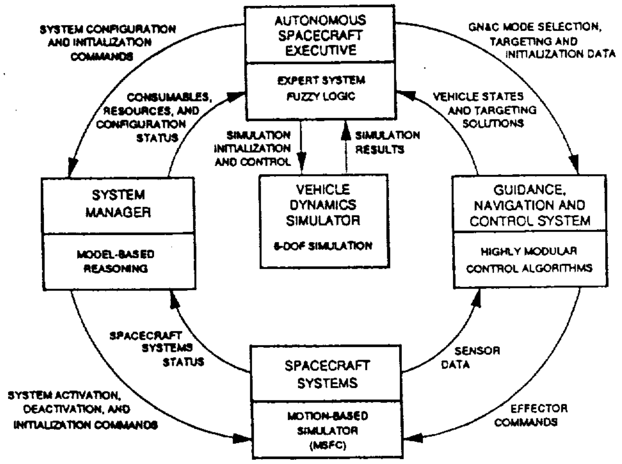


Fig. 1 Architecture of Autonomous Spacecraft

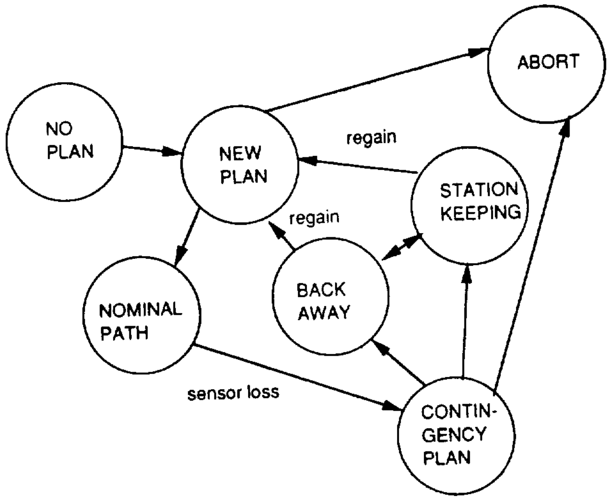


Fig. 2 State Transition Diagram for Proximity

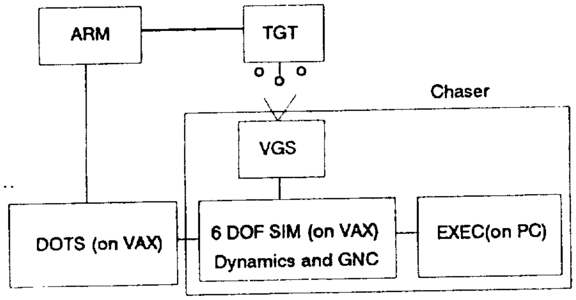


Fig. 3 Configuration of Verification Test Facility

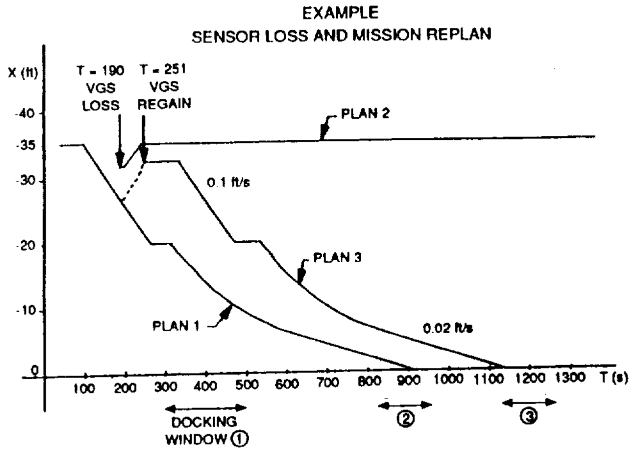


Fig. 4 Sample Test Result

Table 1. Control Commands for PLAN1

T(sec)	X(ft)	EVENT
0	-35	SET LVLH FRAME
0	-35	SET TARGET POINTING
0	-35	START STATION KEEPING
97.3	-35	START APPROACH
263.0	-20	START STATION KEEPING
273.0	-20	START TARGET BODY FRAME
273.0	-20	START ATTITUDE HOLD
313.0	-20	START APPROACH
919.1	0	START STATION KEEPING

Table 2. Control Commands for PLAN2

T(sec)	X(ft)	EVENT
191.0	-31.8	START STATION KEEPING
191.6	-31.8	SET LVLH FRAME
191.6	-31.8	SET TARGET POINTING
199.1	-31.8	START SEPARATION
240.8	-35	START STATION KEEPING

FORMALIZING PROCEDURES FOR OPERATIONS AUTOMATION, OPERATOR TRAINING AND SPACECRAFT AUTONOMY

François Lecouat, Arnaud De Saint Vincent

MATRA MARCONI SPACE
31 rue des Cosmonautes
31077 Toulouse Cedex, France

Tel: +33 61 39 69 39 Fax:+33 62 24 77 80
email: lecouat@isis.matra-espace.fr

N95- 23720

INTRODUCTION

Procedures play a key role in the space domain, since most of the activities that require commanding a spacecraft are based on procedures. Procedures permit to keep the spacecraft inside safe limits whatever happens during operations. Another important property of procedures is that they are a convenient support for bringing together various kinds of expertise in a way that facilitates validation: procedures are written in a language that can be understood by most people involved in a space project.

The generation and validation of operations procedures is a key task of mission preparation that is quite complex and costly. This has motivated the development of software applications providing support for procedures preparation. Several applications have been developed at MATRA MARCONI SPACE (MMS) over the last 5 years. They are presented in the first section of this paper. The main idea is that if procedures are represented in a formal language, they can be managed more easily with a computer tool and some automatic verifications can be performed. One difficulty is to define a formal language that is easy to use for operators and operations engineers.

Once formalised procedures have been generated for a spacecraft, they can be used by other tools for many interesting applications including generation of detailed timelines, automatic or semi-automatic procedure execution, and operators training. Such applications developed by MMS are described in this paper.

Moreover, this concept of formal operations procedures can be adapted to on-board procedures for representing the information necessary to increase spacecraft autonomy. This idea has been explored on the AMR mobile robot and is being developed on the IARES project for CNES dedicated to the development of a demonstrator of a planetary exploration mobile robot.

PROCEDURES PREPARATION

The POM tool has been developed by MMS to support the generation and maintenance of satellite ground control procedures, and to facilitate their use during operations thanks to a procedure browser. POM is now used operationally for the procedures of the Telecom 2, HISPASAT and SOHO spacecrafts. Savings that can be credited to POM during the procedure elaboration phase at MMS were estimated at 50%. Another fine result was the increase of procedure quality.

From the experience of the various procedures management tools developed in the last five years (including the POM, EOA and CSS projects [4]), MMS has derived OPSMAKER, a generic tool for procedure elaboration and validation. It has been applied to quite different types of missions, ranging from crew procedures (PREVISE system [5]), ground control centers management procedures (PROCSU system), and - most relevant to the present paper - satellite operation procedures (PROCSAT developed for CNES, to support the preparation and verification of SPOT 4 operation procedures, and OPSAT for MMS telecom satellites operation procedures).

The basic functions provided by OPSMAKER procedures preparation applications are :

- a procedure editor which supports "assisted editing" (e.g: on-line access to system data) for more efficient procedures writing;
- a procedures compiler, which generates an internal, formal representation of the procedures (and, when applicable, detects syntactic errors);
- a procedures formatter, which generates automatically a high-quality document (FOP, Flight Data File);
- a procedures checker, based on qualitative simulation, which provides for a rich set of verifications to speed up procedure development : simple errors are detected early before starting detailed simulations.

Procedures are entered with the editor in a special form with several columns and various fields. The body of a procedure is entered in a formal language that is a normalisation of the natural language usually used in operations. Quick access to system data (e.g. TM, TC, TC blocks, ground system data) is provided as well as various search mechanisms. In PROCSAT and OPSAT, procedures are saved in a relational database enabling fast search functions and safe team work: several instances of the editor can be opened at the same time (client-server architecture).

The use of a formal language for representing procedures in the Editor (operations engineers view) enables the implementation of a procedure compiler that generates an internal representation of the procedure. The formater then generates a command file for a standard desktop publishing tool (e.g. FrameMaker). Data from the database is automatically inserted in the procedures (e.g. verification TM for a TC, list of TCs for a block) to build up the operators view. The procedures can also include additional information (text and graphics). Formalisation of procedures and modelling of actions facilitate team work by guarantying homogeneous procedures manuals. Everybody works at the same level of detail, with the same language. Maintenance of procedures is facilitated since information is never duplicated and powerful search functions are provided. The use of a normalised language and a normalised presentation by the operations team, should secure the execution of operations.

Several verification mechanisms are provided ranging from simple "local" checks on the individual consistency of every statement, up to the "logical" verification of a procedure by simulating the effects of commands and checking operations constraints (e.g. TC and TC groups pre-validation checks). These verification functions work on the basis of information stored in the spacecraft database. Consistency checking of the operational data and the use of these data without possible corruption improves the consistency and quality of procedure manuals.

There are on-going studies at MMS on the adaptation of OPSMAKER to support *integration procedures* . These procedures used in spacecraft integration have a lot of common aspects with operations procedures. Common data structures and tools would significantly increase spacecraft development productivity.

Another part of mission preparation activities is devoted to the preparation of timelines, in particular for LEOP (Launch and Early Orbit Phases) and IOT (In Orbit Test) operations. An additional advantage of formal procedures is the possibility of detailed timeline generation. Once a top level timeline has been created (timed sequence of procedures) it is quite easy to explore the procedure database and print each procedure action, together with its execution time, in a detailed timeline.

PROCEDURES EXECUTION

Requirements for the improvement of operations safety and efficiency motivate the development of advanced tools to provide a real-time support to spacecraft operators during monitoring and control activities.

The Expert Operators' Associate (EOA), developed for ESOC by MMS and CRI is a prototype centered around the concept of assisted procedures execution. The EOA procedure language allows to attach to the procedure some informations which were not present in the "conventional" procedures: goal, context of applicability, and a more complete description of the execution constraints.

EOA main functions include :
- real-time monitoring of spacecraft telemetry and alarm filtering;
- on-line selection of applicable procedures, in particular in case of contingency;
- managing a timeline of procedures;
- supporting the operator for the execution of the procedure (presenting the chosen procedure to the user in both textual and graphical form, and dynamically reflecting on the display the status of execution of the procedure). Automatic execution of procedures is also possible;
- continuously verifying the validity of operations constraints.

A procedure interpreter allows safe procedure interruption and restart as well as concurrent execution of procedures. A reactive architecture ensures that appropriate response is given to user queries and incoming alarms.

The EOA has been interfaced to the ESOC Multi-Satellite Support System (MSSS), and experimented with MARECS spacecraft analysts on the MARECS simulator, and on the MARECS B2 spacecraft where an eclipse operations was executed by EOA in a completely automatic way (in parallel to the operator). This demonstrates the feasibility of a generic mechanism for semi-automated procedures. Moreover a lot of progress has been made in applications such as PROCSAT and OPSAT, to make the procedure language easy to use by operations engineers. This is a very important aspect for the maintainability of procedure and the acceptability of the tool by users.

MMS is now developing a new generation procedure execution tool that is compatible with the OPSMAKER approach for procedure generation. This procedure executor shall be easily connected to existing control centers as an add-on tool. Expected benefits include:
- improved reliability of spacecraft control thanks to pre-recorded procedures, automatic TC uplink verification, greater number of checks (constraints verification), assistance in conditional branching, timely invocation of procedures from schedule...
- improved efficiency of spacecraft control: operators can be relieved from real time monitoring for well tested procedures (e.g. eclipse procedures), fast execution of recovery procedures (e.g. payload switch-off).

With respect to ad-hoc computer programs implementing procedures, this concept shall permit:
- better observability and control
- an interactive execution mode where the operator can be fully in the loop
- a formalism to encode the operations that does not duplicate efforts and that facilitates maintenance.

OPERATOR TRAINING

Operators training in a spacecraft control center is a recurrent activity, which is going to take an increasing importance with the growing complexity and increasing life duration of modern spacecrafts. In this perspective, it appears essential to develop new training environments/tools allowing to make this task easier and less demanding on instructors availability.

This is the objective of the on-going ATIS project, carried out by CISE and MMS for ESA/ESTEC [1]. This system is applied to the case of astronaut training to the operation of a microgravity payload (RAMSES), but is based on widely applicable concepts and mechanisms which are :

- tutoring functions/modes : in these modes, the user can access to and navigate in technical / operational documentations, either in a free manner, or being guided by the system following an initially specified "training objective";
- procedural training functions/modes : in these modes, ATIS is connected to a simulator. The session is started by specifying an initial scenario (possibly a contingency case) ; the user (operator) executes an operational procedure as in "traditional" simulation session, but is constantly monitored by ATIS which in parallel tracks the procedure to be executed. In case of error, the operator is given corrective guidance. Contextual access to relevant informations is also provided.

Such functionalities could be usefully integrated to a Mission Control Center. A key point is that such a tool can reuse a large part of data and knowledge already produced by other tools during mission preparation (in particular formal procedures). Having a unique source of information for training and operations will enforce the representativity of training.

INCREASING AUTONOMY WITH ON-BOARD PROCEDURES

Many space projects can benefit from a greater spacecraft autonomy. This can be achieved by:
- performing well defined operations on-board without ground intervention
- optimizing the use of the communication link, and of ground processing by generating synthetic reports for the ground
- providing on-board anomaly handling mechanisms.

Formal procedures associated to an on-board procedure executer can help to achieve these requirements. A library of data structures representing operations procedures is stored on board. Procedures to be executed are referenced in a master timeline, and the procedure executer starts interpreting each procedure at the appropriate time. This brings many advantages with respect to dedicated on-board software or to simple on-board command sequences:
- convenient representation: a procedure is more expressive than a command sequence (it contains command verifications, branches, constraints).
- cost saving: procedures are directly written by operations engineers in a high level language, not by software developers.
- ease of validation: the control mechanism is decoupled from procedures. When a new procedure is written the control mechanism has not to be validated.
- finer control: progress of a running procedure can be monitored. Execution can be interrupted and resumed. General exception handling mechanisms can be provided.

An alternative to procedure execution for increasing autonomy would be planning. Not only these techniques are quite complex to be implemented on-board, but they may be not very well suited. Two simple facts give an idea why state of the art planners cannot replace procedures. First of all, space operations are often described with constructs not supported by planners, like loops and execution constraints. Second, the goals that underlie operations preparation are not only expressed in term of states, like in most planners, but also in term of behaviour over a period of time as described in [3] (e.g. "diagnose cause of alarm1 and alarm2 before reconfiguration").

The AMR mobile robot project and the on-going IARES project for CNES are two contexts in which MMS explores related ideas.

CONCLUSION

The formalization of operations procedures brings a lot of benefits. It facilitates mission preparation thanks to automated procedure verification and formating tools. It also makes possible new applications for operator training and operations automation.

REFERENCES

[1] A.Bertin, L.Bertotti, F.Buciol, A.De Saint Vincent, P.Caloud, J.Kaas, F.Allard, *An Intelligent Tutoring System for System and Procedural Training of Astronauts*, 4th ESA Workshop on AI and KBS for Space, Noordwijk, The Netherlands, May 1993.

[2] J.M.Darroy, F.Lecouat, J.M.Brenot, A.de Saint Vincent, *OPSWARE : A new generation of software tools for making space operations faster, better and cheaper*, 44th Congress of the International Astronautical Federation, Craz, Austria, Oct. 1993.

[3] M.Georgeff. *Reasoning about procedural knowledge*, Proceedings of the AAAI 85 Conference, 1985.

[4] F.Lecouat, M.Nielsen, K.Grue, *Knowledge-Based Assistance for Generating and Executing Space Operations Procedures*, ESA Workshop on AI and KBS for Space, Noordwijk, The Netherlands, May 1991.

[5] A.De Saint Vincent, F.Lecouat, G.Leonis, F.Allard, *PREVISE: a KBS to Support the Preparation and Verification of Crew Procedures*, 4th ESA Workshop on AI and KBS for Space, Noordwijk, The Netherlands, May 1993.

The AUSTRALIS-1 Control Architecture — A Behavioural Model for Satellite Autonomy

C. Lindley

THIS PAPER WAS NOT SUBMITTED.

N95- 23721

The Starpicker Expert System — A Problem in Expertise Capture

David A. Smith and G. Martin Hudson
Hughes Information Technology Corporation
16800 E. Centre Tech Parkway
Aurora, CO 80011, U.S.A.
Tel: (303) 344-6175 Fax: (303) 344-2903
smith@redwood.hac.com

KEY WORDS AND PHRASES

Expert system, spacecraft planning, attitude determination.

ABSTRACT

This paper describes the Starpicker expert system, a tool for spacecraft operations planning. Both programmatic and technical aspects are discussed.

BACKGROUND

The Space Precision Attitude Control System (SPACS) Star Sensor was designed and developed by Hughes for use on the HS-318 satellite bus. This is a spin-stabilized spacecraft whose purpose is to provide an accurately positionable platform in earth orbit. The Star Sensor serves as the primary attitude reference.

The function of the Star Sensor is to determine the orientation of the spacecraft spin axis in three-dimensional space, as shown in Figure 1. The sensor operates by measuring the elevation of two selected stars relative to the equatorial spin plane of the spacecraft. These stars are chosen near the spin plane and are ideally separated by about 90 degrees of rotation. Using a catalog of absolute star positions on the celestial sphere, the spin axis of the spacecraft can be accurately determined.

Two sensors are placed on the rotating portion of the spacecraft. Each sensor has a vertical field of view spanning six degrees. One sensor is centered three degrees above the spin plane and the other is centered three degrees below, resulting in twelve degrees of total coverage. The sensor in use is programmed to "open" or "gate" at fixed moments during the rotational period of the spacecraft — once for the primary star and once for the secondary star. During each gate, the elevation of any bright object appearing in the sensor will be measured.

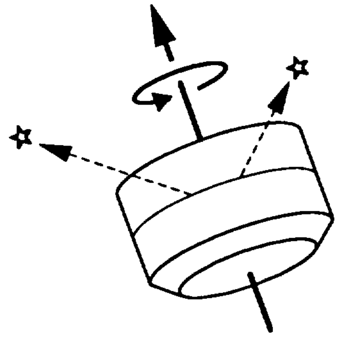


Figure 1. Spacecraft attitude determination

THE PROBLEM

It would seem that with an estimated 200 billion stars in our Galaxy, there would be plenty of stars to choose from. However, a variety of constraints combine to make this a challenging problem in operations planning:

- The sensor has programmable sensitivity — at its most sensitive, the sensor can gate on about the 300th brightest star in the sky.
- Both stars must be within the same sensor's field of view (either above or below the spin plane).
- The separation between the two stars should be between 30 and 150 degrees — the closer to 90 degrees the better.
- The sensor cannot discriminate between stars in the sensor which are less than 4 degrees apart. In this case, neither star is usable.

- The previous constraint also applies when one of the bright planets appears in the sensor — i.e., Mercury through Saturn.
- Glare generally prevents use of any star within 60 degrees of the sun in any direction, although the brightest stars are still usable somewhat closer than this. The motion of the sun by about one degree per day frequently limits the number of days that a star can be used.
- When the moon is in the sensor, glare generally wipes out any stars 15 to 20 degrees before or after the moon. This effect depends on the phase (and therefore brightness) of the moon.
- The appearance of the earth in the sensor during the spacecraft orbit may obstruct visibility of stars. The glare of the sun shining on the earth makes the affected area larger.
- Over time, the sensor becomes degraded in sensitivity, making dimmer stars unusable and reducing the glare-immunity of brighter stars.
- Some stars vary in brightness over periods ranging from hours to months, making their use problematic. Some other stars seem to yield low-quality data, presumably from the presence of nebulosity or other sources of sensor noise.

The above constraints must all be accommodated in order to achieve nominal operations. Unfortunately, there are times when not all constraints can be satisfied. In these cases it is necessary to find the best possible fall-back solution so that operations can continue.

EXPERT SYSTEM DEVELOPMENT

The Starpicker expert system was built to help choose attitude determination stars. The expert system captures both the nominal selection criteria described above and the fall-back heuristic methods.

The development of Starpicker is outlined in Figure 2. The idea grew out of a study that focused on automated capture of human operations expertise. Starpicker is the first such tool to be identified and built.

Two prototype versions of Starpicker were built using Nexpert Object on a 386-SX PC platform. The first prototype was built in the space of about 6 weeks and captured the nominal criteria for star selection. The second prototype required another 6 weeks and implemented a revised control structure. This second version was organized as a hierarchy of computational strategies so that progressively more "desperate" measures could be applied in difficult cases. These prototypes served as a credible proof of concept, but fell short of an operational capability.

The operational version of Starpicker was built using ART-IM on a Sun SPARCstation platform. Development of the operational version of Starpicker required about 15 months and resulted in 4400 lines of ART-IM code and 6800 lines of C code. The ART-IM code comprises 127 rules and 172 functions.

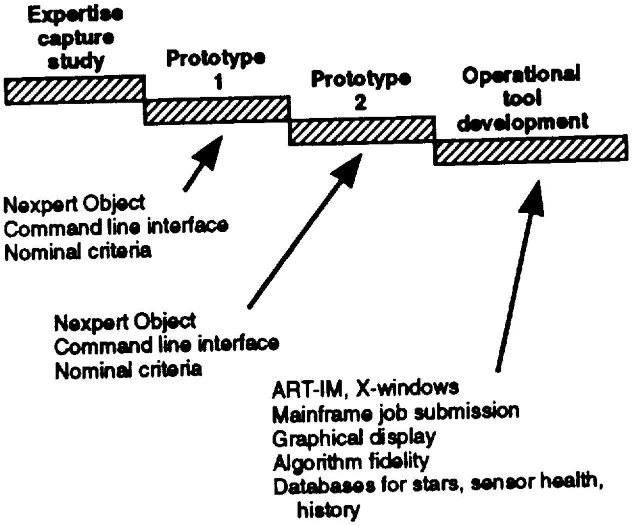


Figure 2. Starpicker development overview

IMPLEMENTATION TECHNIQUES

During the expertise analysis phase of this project it became evident that numerous rules of thumb are used by the expert — for example, estimating the range of glare interference in various situations. A design goal was to avoid discontinuities in the program behavior when a star is found to be just inside or just outside such a range threshold. To do this, a "fuzzy logic" model is used. The glare near the moon, for example, is characterized by a fuzzy region. At one edge of the region a star is considered "certainly unusable," and at the other edge it is "certainly usable." In between, the star is assigned a usability that is between zero and one. (See Figure 3.) Using this technique, the expert's heuristics are represented directly, and the system behavior is not highly sensitive to small variations in the exact values chosen. This formalism was found to be a useful knowledge representation, although only a minimum amount of "fuzzy inferencing" is done in the system.

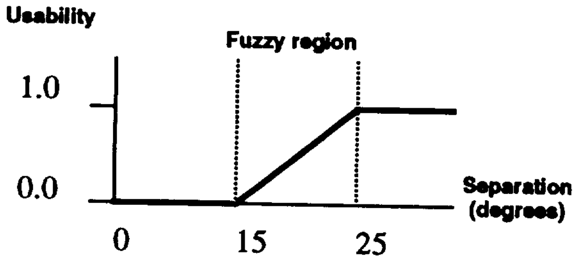


Figure 3. Example of fuzzy transition region — usability of a star in the presence of moon glare

A major concern in the Starpicker design is to prevent combinatorial explosion in the generation of candidate star pairs. To do this, two dynamic lists of stars are maintained, a list of candidate primary stars and a list of candidate secondary stars. At any given time, the currently enabled pair-formation rules generate all admissible pairs using these two lists. Membership in the two lists is gradually augmented until a desired number of pairs has been generated. This process is heuristically organized so that the better pairs are likely to be generated first. The final list of pairs is then ranked based on pair quality.

Star lists are implemented using the dynamic class-membership facilities of ART-IM. The cyclic process of adding stars and generating pairs is implemented with phased rule firings, using the ART-IM rule "salience" mechanism. ART-IM rules are organized into levels of priority or "salience," so that at each execution step the eligible rule with the highest salience is the one that is fired. In Starpicker, a low-salience rule examines the number of pairs generated so far. If more pairs are desired, the next strategy is taken from a list of strategies, appropriate rules are enabled, and the higher-salience rules are allowed to fire again to generate more pairs. Successive strategies from the strategy list will therefore be applied until enough pairs have been generated or the strategy list has been exhausted.

This architecture for the rule base is both easy to understand and easy to use. Changes in the overall problem-solving approach are easily implemented by editing the initial strategy list. This has proven to be a useful vehicle for explaining the implementation to the expert and incorporating his feedback.

A typical strategy list is shown in Figure 4. Two kinds of information are recorded in a strategy list — rule groups and parameter threshold settings. A list item with two elements, such as

```
(strategy dual-sec),
```

denotes a rule group to be enabled. When this strategy is enacted, a fact that enables a selected group of rules is added to the data base. A list item with three elements, such as

```
(pri-thresh -0.1 0.0),
```

is used to control a numeric parameter in the pair generation process. When this strategy is enacted, the specified parameter is progressively stepped (in this case by -0.1) until the specified ending value has been reached (in this case, 0.0). A strategy item of this form may therefore cause several passes through the pair generation rules, one for each iterated parameter value. (Terms in Figure 4 beginning with a question mark are global values defined elsewhere in the code.)

EXPERIENCE

A key factor in the success of this development was the availability of a domain expert who was both supportive of the goals of the project and physically available for consultation. During the development, the domain expert and the principal knowledge engineer were located in the same office area so the knowledge engineer could observe the expert's working practices and quickly resolve questions about the implementation. This close interaction with the expert may have contributed to schedule delays, but the resulting product was significantly improved.

User reaction to Starpicker has been generally favorable. The primary user, the "Star Analyst," uses Starpicker on a regular basis. This individual has extensive experience in the problem domain and has defined many of the current practices. Not surprisingly, therefore, the user does not view Starpicker as a black box for planning solutions. Instead, the user sees Starpicker as a "source of confirmation," since he frequently has a tentative solution in mind before starting to use the tool. He values Starpicker for its convenient access to pertinent information, its "conservative estimates," and the fact that it "doesn't make mistakes."

An important factor in the acceptance of this tool is that its conclusions can be overridden when necessary. The user can also easily update the external data files to reflect experience with new stars and changes in sensor health.

Equally important user feedback comes from individuals who serve as backup Star Analysts in the absence of the primary expert. The reaction from these users has also been generally positive, but it is interesting to note occasional differences in approach. For example, one user states that he is much more willing than others to "push the rules" regarding the star selection criteria. Observing these occasional users, it seems that an on-line help facility would be desirable as an alternative to the written documentation. A tutorial user mode would also be helpful.

Neither the expert nor the occasional users seem inclined to accept Starpicker's recommendations on blind faith. The users prefer to have access to as much supporting information as possible in order to evaluate for themselves the recommendations of the system. A certain degree of subjective judgement appears to go into the final choice from among the available solutions. This judgment process, which has not yet been formalized, trades off such factors as the quality of the stars versus the expected duration of the solution. The users have expressed general satisfaction with Starpicker as both a source of recommendations and supporting information, and it has become a standard resource in day-to-day operations.

CONCLUSIONS

In conclusion, we attribute the success of this program to a combination of programmatic and technical factors. The initial prototyping cycle was useful in defining the concept of the tool, establishing its scope and operation, and providing a convincing demonstration prior to development. ART-IM was found to be powerful, stable, and well suited to this project. Close physical access to the domain expert during the development and the expert's positive and helpful disposition contributed significantly to the quality and usefulness of the final product.

DISCLAIMER

None of the descriptions of commercial software products in this article should be considered an endorsement or criticism by Hughes Information Technology Corporation. These remarks are derived from experience which may or may not be directly transferrable to other applications.

```
(deffacts strategy-list
  (strategy-list
    (strategy nominal)                       ; nominal
    (pri-thresh ?*pri-delta* ?*quality-g*)   ; decrease quality by steps
    (abbrev-limit ?*abbr-delta* ?*abbr-lim*) ; permit abbreviated use
    (strategy dual-sec)                      ; permit dual secondaries during rev
    (strategy relax-sep)                     ; relax separation
    (strategy use-planets)                   ; enable use of planet
    (pri-thresh ?*pri-delta* ?*quality-p*)   ; further decrease quality
    (abbrev-limit ?*abbr-delta* ?*abbr-max*) ; further relax abbrev use
    (pri-thresh ?*pri-delta* 0.0)            ; further decrease quality to zero
    (strategy really-relax-sep) ))           ; relax separation to max
```

Figure 4. Sample strategy list

SESSION SE

Servicing

Satellite Servicing in GEO by Robotic Service Vehicle

W. De Peuter and G. Visentin

ESA
Noordwijk, The Netherlands

THIS PAPER WAS NOT SUBMITTED IN TIME FOR PUBLICATION.

Robotic Servicing System for Space Material Experiment

N95-23722

Toshihiko Yamawaki , Haruhiko Shimoji *, Toshio Abe**

National Space Development Agency of Japan (NASDA)
2-1-1, Sengen, Tsukuba, Ibaraki 305, JAPAN
Tel. Int-298-52-2772, Fax. Int-298-52-1597

* Central Research Laboratory, Mitsubishi Electric Corporation,
8-1-1, Tsukaguchihonmachi, Amagasaki, Hyogo 661, JAPAN
Tel. Int-6-497-7161, Fax. Int-6-497-7291, E-Mail shimoji@mec.crl.melco.co.jp

** Kamakura Works, Mitsubishi Electric Corporation,
325 Kamimachiya, Kamakura, Kanagawa 247, JAPAN
Tel. Int-467-47-2136, Fax. Int-467-47-1874

KEY WORDS

Robotic Servicing System, Automation, Teleoperation, Image Furnace, Japanese Experimental Module

INTRODUCTION

A containerless image furnace with a electro-static positioning device has been developed as one of material experiment facilities on the Japanese experimental module (JEM). It is characterized by heating / melting / cooling the sample whose position is kept without any contacts by actively controlled electro-static force exerted between the sample and a set of electrodes.

The experiment using the image furnace requires various servicing operations. We have been developing a robotic servicing system with an internal robot accommodated in the rack as an alternative to the crew. It aims to reduce the load of the crew by automating regular tasks and to increase the flexibility applicable to simple irregular tasks by introducing a remote teleoperation scheme.

The present robot has poor capability to replace the crew. In order to compensate it, introducing of the concept of the robot friendliness and improving the controllability of the teleoperation by the ground operator aids are essential.

In this paper, we identify the tasks to be performed by the robotic servicing system and discuss the way to compensate the capability of the robot. In addition we describe the evaluation tests using an experimental model.

SYSTEM CONFIGURATION

The total system packed into a rack as shown in Fig.1 is loaded on the pressurized module of JEM. The total system consists of an image furnace, a robotic servicing system, and a container.

The image furnace is composed of a spheroid mirror with a heating lamp inside and a vacuum equipment to process samples in a high vacuum.

The robotic servicing system is composed of a robot, CCD cameras, and a robot controller. The robot has six joints and its length is about 0.8 [m]. A force sensor and a hand camera are mounted on the wrist and a touch sensor at the hand. The robot, which is furnished in a limited volume, about 0.6x0.4x0.8[m], must have large reaching envelope in order to handle as many ampule as possible.

The container, which is a sample storage, stores about 50 ampules with a sample. Each sample is in a transparent ampule respectively so that the sample and the environment may not be

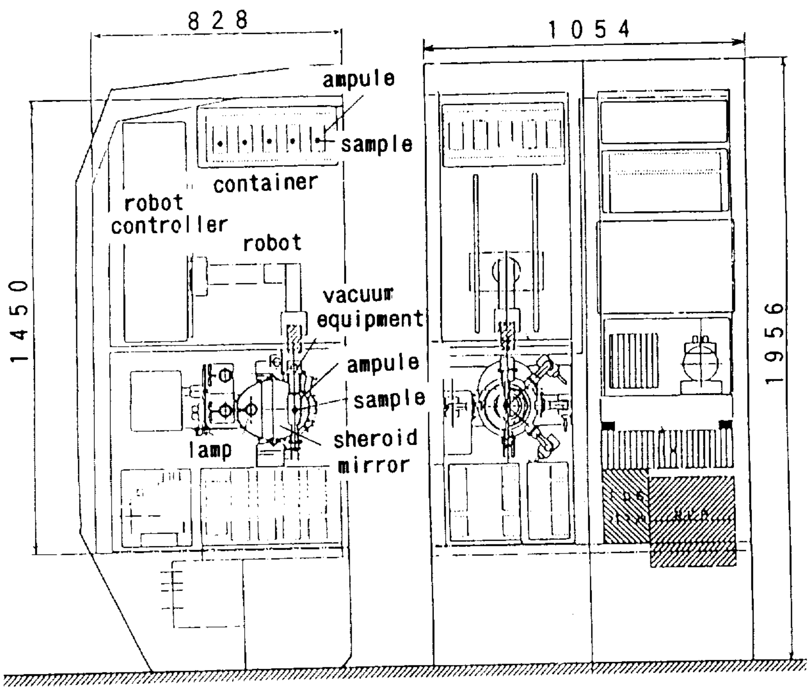


Fig.1 Total System Configuration

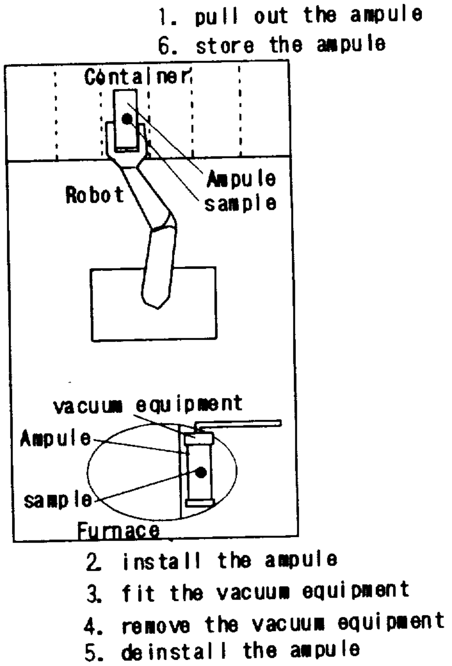


Fig.2 Regular tasks of exchanging ampule

contaminated.

ROBOTIC SERVICING SYSTEM

Tasks Performed by Robotic Servicing System

It is not necessary to automate all tasks. The tasks which are rarely required and includes complicated operations are performed by the crew.

The tasks of supporting the experiment are roughly divided into regular tasks and irregular tasks. The former is standard and pre-defined tasks and the latter is not.

The former consists of two kinds of tasks. One is the ampule exchanging task performed along the following process as shown in Fig.2. The robot pulls out an ampule from the container, installs it in the furnace, and fits it with the vacuum equipment. After the experiment, the robot removes the vacuum equipment, deinstalls the ampule from the furnace, and stores it to the container. These tasks involve various subtasks such as opening/closing the lid and handling the lever to fix/release. The other is the container exchanging task required after all samples are processed. It is intended that the former is performed by the robotic servicing system and the latter is done by the crew.

The irregular tasks are needed in the cases when the operation error occurs and when ground scientific investigators request undefined operations. In the former, such as improper gripping and the collision during transportation, the ground operator who has enough information on the facility inspects and recovers the situation by operating the robot if possible. If not, the crew do it. In the latter case, the operator controls the robot according to the investigator's request. For example, when the investigator requests to observe the processed sample, the operator controls the robot to set the ampule in various orientation in front of the camera.

Trade-off in Constructing Servicing System

In constructing the servicing system there are two items to be traded off. One is the trade-off between a servicing system with a robot and with a dedicated mechanism. We selected the former as it is superior to the latter in the following aspects; complexity of the mechanism, applicability to the

C-4.

irregular tasks, and the accessibility of the crew in case of accident.

The other is the trade-off between a robot mounted inside the rack and outside. As the experiments need the dexterous operations around the furnace, we selected the former to relax the mechanical interface requirements. In addition the latter requires severer safety level since the robot and the crew are accommodated together in the aisle.

Key Technologies of Robotic Servicing System

Obviously the capability of a present robot is much inferior to that of crew. In an aspect to execute commanded tasks, the dexterity and the tip force are insufficient. In order to compensate it the mechanical design with a concept of the robot friendliness is important.

In another aspect, that is autonomy, the robot does not have sufficient ability to cope with the irregular occasion by itself. The ground operator controls the robot with the remote teleoperation scheme to cope with it. But the remote teleoperation has limited controllability because of the communication time delay. The way to improve it with the ground operator aids using computer is important.

In the following section, these two items are discussed.

Automatic Operation

Exchanging the ampule as the regular tasks are executed automatically. The reliable execution is achieved by the following robot friendly design.

- the unification in size and shape of grasped surface
- the mechanism to fix/release the object only by pushing/pulling it, followed by the firm fixation with other methods after releasing it
- the mechanism to adjust the object position only by moving it along the guide
- the mechanism to handle the object with small force

We also designed the controller with following functions.

- to adjust the hand position by force control
- to verify the fixation of the ampule before releasing it
- to monitor the operation in real time using sensor information
- to decide whether to try again or not when the error is detected

As an example, the tasks of the ampule installation in the furnace is done as following. First, with the force control scheme, the robot pushes the ampule on the positioning guide. The ampule position is automatically adjusted. Next the latch attached on the ampule automatically hooks to the furnace. The robot can fix the ampule without additional motion while grasping it. Moreover, the robot releases it only after confirming the correct fixation by trying to pull it out, in order to prevent the ampule from freely-floating.

If the error is detected by the sensors, the robot performs the task again or requires the remote teleoperation according to the situation.

Remote Teleoperation

The rather simple irregular tasks are performed with remote teleoperation. But it has the poor controllability because of the communication time delay estimated at about 10 seconds.

In order to increase the controllability, a software simulator with following functions as the operator aids is introduced to the ground system.

- to estimate the robot motion without the time delay
- to overlay the estimated robot motion on the actual video image
- to correct the internal model by manually moving the estimated images on the actual ones
- to display the graphical image from appropriate position
- to record, edit, and play the command

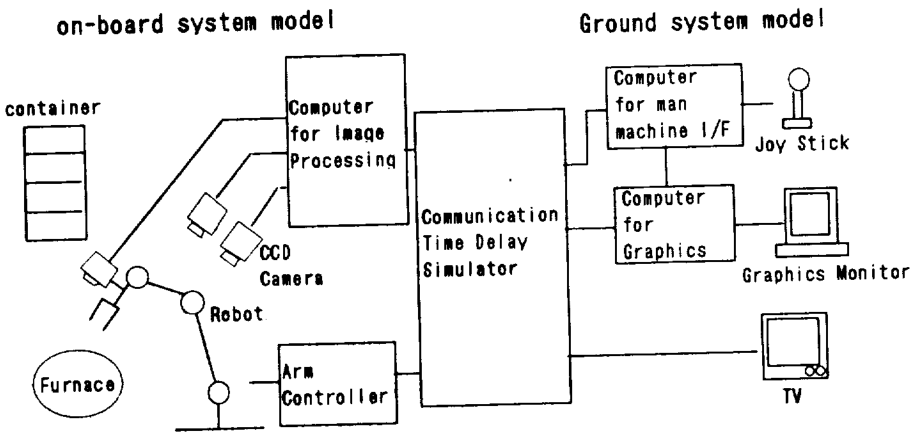


Fig.3 Block diagram of the experimental model

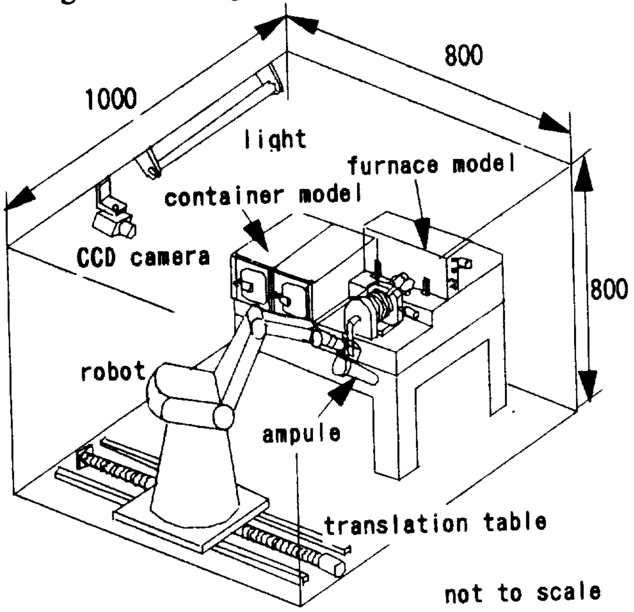


Fig.4 Outlook of the experimental onboard system

- to check and display the sensor data

Using the simulator, the operator can control the robot without feeling the time delay as he interacts with the graphically simulated environment. In addition, he can send the motion command to the onboard, only after verifying the commanded motion through the preview display. The simulator increases the reliability of the operation as well as the controllability.

EVALUATION TESTS

Evaluation tests are performed using experimental model. The objectives are to verify the feasibility of the robotic servicing system and to improve the reliability and the dexterity.

Fig.3 shows the block diagram of the experimental system. It consists of an on-board system model and a ground system model which are connected via the communication time delay simulator. Fig.4 shows the outlook of the onboard system model. It is composed of a robot, a container model, and a furnace model. The robot hand is newly designed but the robot arm is an industrial one because the robot configuration is not main issue.

Two experiments are performed. One is to perform a sequence of the ampule exchanging task automatically. The other is to install the ampule in the furnace with remote teleoperation. At the present state, the experimental results make sure that the tasks can be performed by the robotic servicing system.

CONCLUDING REMARKS

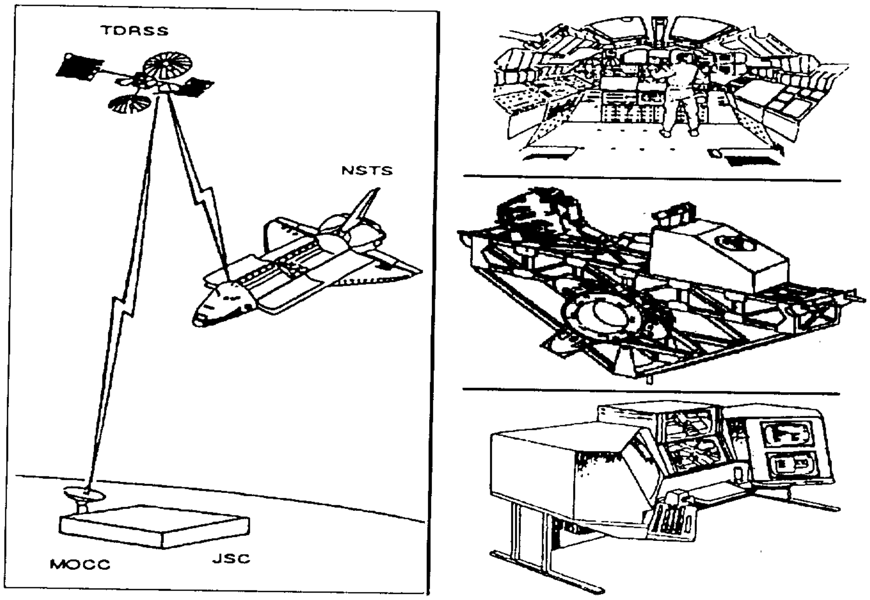
The material experiment using the image furnace needs various servicing tasks which are desired to be automatically performed without the crew. We have been developing the robotic servicing system to perform the tasks of exchanging samples and simple exceptional handling. The system can reduce much crew time. This paper identified the tasks to be performed by the system and discussed two important items to construct it; the mechanical design with the concept of robot friendliness and the way to improve the controllability by the ground operator aids. In addition we constructed the experimental model and verified the feasibility.

We are now constructing the bread-board model including a newly designed robot, and will evaluate its capability considering the effects of the limited work space and 0-g environment.

DEXTEROUS ORBITAL SERVICING SYSTEM (DOSS)

N95- 23723

Charles R. Price, Reginald B. Berka, and John T. Chladek
NASA/Johnson Space Center
Tel: (713)483-1523, Fax: (713)483-7580
e-mail: crprice@ctsd2.jsc.nasa.gov



AFT FLIGHT DECK:
Crew Operated —
- Portable Workstation
- 3 DOF Handcontrollers
- Command & Display Panel (JSC)

PAYLOAD BAY ELEMENTS:
- Dexterous Manipulator & Avionics (MMAG)
- Controls Architecture (MMAG & LaRC)
- ISSA Task Elements and Robotic Tools (JSC)

MISSION OPS CENTER:
- POCC (JSC)
- Mission Monitoring
- Remote Control Software (JPL

Figure 1. DOSS Concept

Summary

The Dexterous Orbiter Servicing System is a dexterous robotic spaceflight system that is based on the manipulator designed as part of the Flight Telerobotics Servicer program for the Space Station Freedom and built during a "technology capture" effort that was commissioned when the FTS was cancelled from the Space Station Freedom program. The FTS technology capture effort yielded one flight manipulator and the 1g hydraulic simulator that had been designed as an integrated test tool and crew trainer. The DOSS concept was developed to satisfy needs of the telerobotics research community, the Space Shuttle, and the Space Station. As a flight testbed, DOSS would serve as a baseline reference for testing the performance of advanced telerobotics and intelligent robotics components. For Shuttle, the DOSS, configured as a movable dexterous tool, would be used to provide operational flexibility for payload operations and contingency operations. As a risk mitigation flight demonstration, the DOSS would serve the International Space Station to characterize the end to end system performance of the Special Purpose Dexterous Manipulator performing assembly and maintenance tasks with actual ISSA orbital replacement units. Currently, the most likely entrance of the DOSS into spaceflight is a risk mitigation flight experiment for the International Space Station.

System Architecture

The DOSS is a Shuttle based flight system and consists of three major components: An aft flight deck crew control workstation, a payload bay manipulation and work space element, and a ground control workstation (Figure 1). Specifics of the DOSS workspace components have evolved from a technology centered configuration (generic task panels) and now include the ISSA specific Orbital Replacement Units and associated interfacing tools. These ORU's require a rotary drive function within the robotic gripper to loosen and tighten retention bolts, and the tools are required to access "out of the way" bolt head locations that are not accessible with the baseline SPDM end effector. The robotic function is also required to interface with three types of "handles" on the ORU's that have been accrued from specific incremental design solutions during the development and evolution of the space station.

The aft flight deck work station consists of a laptop computer, two three degree of freedom handcontrollers, a Standard (Orbiter payload service) Switch Panel, data and video recorders, closed circuit television and monitors, and cabling. The operator will be afforded direct aft window viewing of the DOSS payload bay element.

The payload bay element consists of a MPESS payload carrier, the FTS DTF-1 dexterous manipulator mounted on a base on the carrier, four ISSA Orbital Replacement Units (a battery box, remote power control module, multiplexer/demultiplexer "6B" box, and a representative Mobile Servicing System components) also mounted on the MPESS, avionics, and cabling. The grapple fixture on the DOSS is for the contingency jettison of the entire experiment via the Shuttle Remote Manipulator System in the unlikely event that a failure renders the experiment inoperative and in a configuration that is hazardous to the Shuttle.

The ground control station consists of multiple displays, a predictive kinematic graphics simulation, hand controllers, and a keyboard. The high fidelity solid model graphical simulation will be used to preview the expected results of all commands sent from the ground control station to the onboard manipulator, before the "execute" command is sent to allow the manipulator to proceed. As the manipulator then moves, joint angle data will be dowlinked to drive a "wire frame" representation of the manipulator that will "catch up" with the solid model representation of the predicted movement.

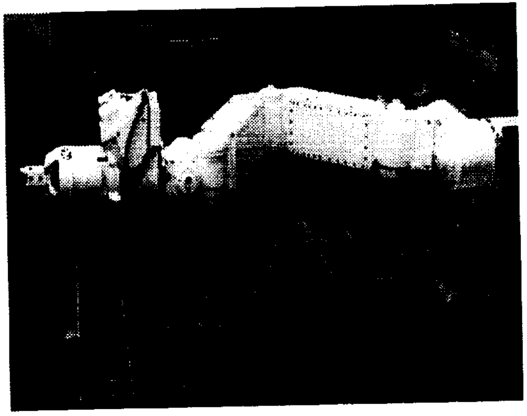


Figure 2. DOSS Manipulator on Air-Bearing Table.

DOSS Flight Experiment Objectives

- Characterize and assess the manipulator design and on-orbit task performance capabilities to improve mission success of future operational space telerobotic systems.
- Develop and evaluate an aft flight deck man/machine interface for on-orbit teleoperation with future capability to accept control from a ground-based telerobotic control station.
- Correlate fundamental engineering relationships of system performance in space with ground simulations and analysis predictions to increase fidelity of simulation models used for task assessments, mission planning, training, and recovery techniques.
- Demonstrate the functional utility of an on-orbit dexterous manipulator

C-4

- to reduce EVA operations by performing both Orbiter and Space Station tasks.
- to reduce the risks associated with Space Station first-use of telerobotics.

The Benefits

The DOSS provides valuable on-orbit manipulator demonstrations, experience, and data to the NASA telerobotics community, to the Space Station Program, and to the Space Shuttle Program. To the NASA telerobotics community DOSS is the culmination of many years of technology development in an on-orbit demonstration of our achievements and a platform for additional on-orbit demonstrations. To the Space Station Program DOSS is a vehicle to mitigate risk, gain on-orbit experience, and capture on-orbit performance data regarding dexterous manipulator technologies. The DOSS demonstrated manipulator technologies utilize similar configurations, tasks, and environments as those planned for Space Station and the SPDM. To the Space Shuttle Program DOSS is a potential tool to reduce reliance on EVA operations and reduce EVA timeliness, particularly on over-subscribed satellite servicing missions.

More specifically, DOSS:

- Delivers verification of and experience with Space Station robotic interfaces and maintenance tasks prior to SPDM deployment. Candidate interfaces and tasks include:
    - OTCM Interfaces
    - ORU Interfaces
    - ORU Changeout Operations
    - Alignment and Mating Tasks
    - Inspection/Verification that Tasks and Elements are Secure
    - Visual Surface Inspection Tasks
- Mitigates risk associated with Space Station's first on-orbit use of telerobotic technologies:
    - Impedance Control (Force/ Moment Accommodation)
    - IVA Control of Dexterous Manipulator
    - Flat Flex Cable
    - High Accuracy Manipulator Control
    - Collision Avoidance Techniques
    - Fault Tolerance and Redundancy Management
- Yields on-orbit verification data of manipulator engineering, design, modeling, and analysis prior to deployment of SPDM. Identified areas of interest are:
    - Manipulator and actuator dynamics and non-linearity
    - Manipulator accuracy and repeatability
    - Man-Machine interface (operator fatigue, lighting effects, camera views)
    - Manipulator control envelope
    - Manipulator single joint control
    - Singularity handling
    - Collision avoidance
    - Autonomous functions
    - Impedance control performance and contact stability
- Provides for near-term on-orbit demonstration of telerobotic ground control technologies:
    - Ground-based telerobotic control station
    - Time Delay Handling (e.g. predictive displays)
    - Scene Calibration Methods
    - Safety Assurance Mechanisms
    - Telemetry Interfaces
    - Data Displays
- Supports near-term use of telerobotics program technologies on-orbit for EVA time-line reductions.
- Demonstrates telerobotic technologies and capabilities to perform more elaborate servicing and maintenance tasks and provides an experience base for performing future Space Station task elements.
- Provides on-orbit data regarding performance of dexterous manipulator technologies under environmental extremes and longer duration space exposure.
- Results in re-usable flight hardware for

continued telerobotic technology demonstrations.

- Provides use of the Space Station funded FTS manipulator, cameras, end-of-arm tooling and Standard Data Processor. (see Figure 3).

The Organization

The DOSS is a partnership of three NASA Centers (JSC, LaRC, & JPL) and an expert contractor (Martin Marietta Astronautics Group - MMAG) teamed to produce a flight demonstration of dexterous robotics. The program planning, systems engineering, hardware development, and cooperative agreements for DOSS have already begun. Each participant has an agreed-upon role in providing the final product - a successful Orbiter flight experiment. In this manner, the DOSS team capitalizes upon the strengths of each participant to reduce overall costs, minimize duplication of effort, and produce a technically superior robotics flight experiment.

JSC will manage the program and be responsible for the formal Orbiter payload integration process. This process includes systems engineering, safety analysis and reporting, engineering analyses, and a Payload Integration Plan. JSC will also develop and deliver to MMAG the simplified aft flight deck workstation and the flight task panel with task elements. Engineering models for systems analysis and post-flight verification will be developed and maintained at JSC. Additionally, JSC will support engineering efforts at MMAG, ground control developments at JPL, and controls and crew training at LaRC.

MMAG will develop much of the flight systems and deliver the integrated payload bay elements. The payload bay element include the flight manipulator (currently operational at MMAG, see Figures 2 and 3), the flight avionics (partial designs complete), the aft flight deck command and display systems (partial designs complete), and the system software. MMAG will work with JSC to verify and certify these systems for flight. MMAG will take delivery of all payload elements, integrate and test them, and prepare them for shipment to KSC.

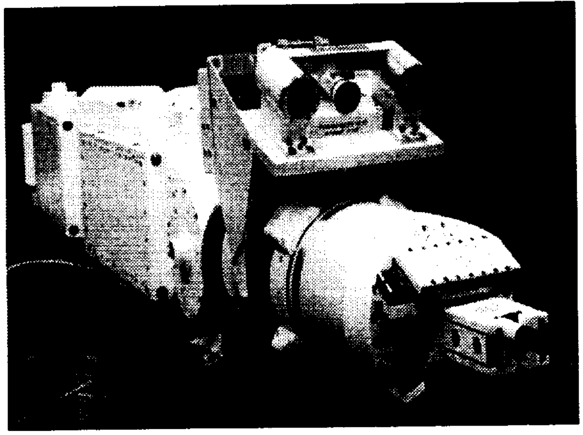


Figure 3. End-of-Arm Tooling with Wrist Camera & Lights

LaRC will use the Hydraulic Manipulator Test Bed (currently in use at LaRC) for MMAG software tests, task panel check-outs, and crew training. LaRC will monitor and direct MMAG development of control software utilizing the LaRC experience-base with manipulator controls and with the HMTB. In coordination with MMAG and JSC, LaRC will prepare the HMTB for crew training and carry out the crew training activities. LaRC will also play a key role in the development and maintenance of the engineering models used for analysis and post-flight verification of the manipulator systems.

As ground control technologies at JPL mature, JPL creates an integrated ground control system that will provide the necessary functionality of a remote POCC (Payload Operations Control Center). Anticipated key elements of the POCC include real-time video, graphic, and predictive displays, off-line task sequencing and verification, availability of autonomous actions, and high rate telemetry feedback & display.

The Space Station Servicing System

D. Hunter

THIS PAPER WAS NOT SUBMITTED.

ROBOTIC SYSTEM FOR THE SERVICING OF THE ORBITER THERMAL PROTECTION SYSTEM

Todd Graham, Richard Bennett, Kevin Dowling,
Davoud Manouchehri, Eric Cooper, Cregg Cowan
NASA Kennedy Space Center
DM-MED-12
Kennedy Space Center, FL 32899
Tel: 407 867-7890, Fax: 407 867-1645
todd-graham@ksc.nasa.gov

KEY WORDS AND PHRASES

mobile robot, ground processing of space shuttle orbiter, active compliance, fail-safe systems, visual inspections using machine vision, advanced information systems

INTRODUCTION

This paper describes the design and development of a mobile robotic system to process Orbiter Thermal Protection System (TPS) Tiles. This work was justified by a TPS automation study which identified tile rewaterproofing and visual inspection as excellent applications for Robotic automation.

BACKGROUND

Robotics and automation technologies have historically not played a role in the ground processing operations of spacecraft and space systems. In part, this has been due to skepticism regarding the viability of these technologies and a strong concern for safety of flight hardware and personnel. In 1990 ground processing activities related to the Orbiter Thermal Protection System (TPS) were investigated [NASA-TPS 90]. The study identified two tasks were automation was technically possible and economically justifiable. These were rewaterproofing and visual inspection of lower surface tiles.

Rewaterproofing

The Orbiter lower surfaces is covered with tiles which are made from highly porous silica fibers covered with a glazed coating. These tiles will absorb water. The absorbed water presents several problems one of which is that it can freeze on orbit and damage the tile. As the Orbiter may be exposed to rain, the tiles must be waterproofed. This is done with, Dimethylethoxysilane (DMES), which is manually injected into a small hole in each tile by a hand held tool. A rubber nozzle is held against the tile and the chemical is forced into the tile by a pressurized nitrogen purge.

Inspection

During launch, reentry and transport tiles can be damaged. This is evident as scratches, cracks, gouges, discoloring, and (or) erosion of surfaces. This damage can impact the flight safety of the vehicle. It is critically important that all tile damage be identified and repaired if necessary. Each tile is visually inspected to see if it has been damaged.

SYSTEM REQUIREMENTS

The primary goal of this effort was to automate rewaterproofing and inspection while minimizing changes to the current methods and process parameters. It was originally considered necessary to do these tasks at any of the three Orbiter Processing Facilities (OPF) or outdoors at the Dryden mate, de-mate facility. It was decided that either automated process should take no longer than five eight hour shifts to complete. Also, it was extremely important to have a design which meets the stringent NASA safety requirements. Finally, the interface to the system must allow effortless manipulation and analysis of an extremely large data set.

At the outset, is was clear that budget constraints made it impossible to deliver a system which had completed the rigorous NASA certification process. So the design team proposed that a certifiable prototype be delivered. This strategy required that the system be designed and fabricated so that all certification requirements could be met without actually completing the required testing and documentation. Once the system design has been validated, additional funding will be sought to fully certify the prototype system.

The system was decomposed into three major sub-systems, 1) sensors and tooling, 2) a positioning mechanism, and 3) an information system. Detailed specifications were written to define the required system.

INFORMATION SYSTEM

An overview of the information system is detailed in Figure 1. Five computer systems are linked together to form the information system. These are 1) existing NASA databases, 2) the WorkCell Controller (WCC), 3) the High Level Controller (HLC), 4) the vision system computer, and 5) the rewaterproof system computer.

The WCC takes data from existing NASA databases and creates the tables which contain the data required by the robot to complete a job. The Oracle Relational Database Management System runs on both the WCC and the HLC. Data transfer between the two systems is accomplished via a temporary Ethernet connection using SQL. The WCC will interface to the Master Dimension Database (MDD) and the Tile Information and Processing System (TIPS). The MDD contains information on the geometry and location of each tile on the Orbiter. This data is used to calculate where to send the robot in order to complete a task. TIPS is a database which contains information about the Orbiter which is dynamic. The WCC utilizes a multitasking, distributed architecture. It is networked using TCP/IP and multiple workstations can be supported.

MOBILE ROBOT

Many options were examined before a mobile robotic system was chosen. This included classes of devices that allowed inspection from afar, large fixed but movable manipulators and even suction-cupped walkers. As a result of these preliminary studies the system chosen was that of a mobile base integrated with a manipulator system.

Mechanical System

The size constraints of the vehicle coupled with the close quarter navigation needs for operating in the OPF required a locomotion system of high maneuverability. A wheeled system utilizing Mecanum wheels was selected. This device utilizes novel roller wheels to obtain three-degree-of-freedom (DOF) motion in the plane. The drive trains for locomotion are within the diameter of the wheel hub. A locking hub allows the operator to disengage the wheels from the drive train completely. This enables the machine to be pushed or towed out of the way in an emergency. The base is formed by a very rigid welded steel frame. The design was deflection driven to provide a very stiff base from which to operate the manipulator. Figure 2 shows a general outline of the sub-systems of the mobile robot. The base also supports two enclosures for electronics and rewaterproofing equipment as well as an on-board nitrogen tank and a battery cage .

Manipulation

When the base reaches a particular work area stifflegs are deployed. The manipulator then deploys itself from it's stowed configuration. The manipulator provides a number of motions to reach the tiles. As shown in Figure 2 the first vertical motion is termed the Major-Z. Linear rails connect the two Major-Z actuators to give a vertically raised rigid platform that can move the rest of the mechanism along the length of the robot. A second vertical motion (Minor-Z extend) is then used to lift the later sections of the manipulator. The two vertical motions are used because a single telescoping device could not provide the combination of stroke length, short unextended height, payload and accuracy needed. Atop this motion is a 360 degree rotating motion (Minor-Z rotate). From this rotate motion a boom nearly a meter in length extends to a stow-deploy link. This link only swings the wrist and toolplate into position for the work. The need for this motion stems from the height requirements and the need to package the robot within the constraints imposed by the facilities. The wrist is a modified Rosheim wrist that provides a hemispherical non-singular workspace. It is capable of moving and accurately positioning the end-effector (25 kg). Precise positioning of the robot relative to the Orbiter is needed to achieve accuracy's of 1mm across the lower surface of the Orbiter. An approach that utilizes two systems delivers the required accuracy. A rotating eye-safe laser scanner reads bar code targets that are precisely located in the facility. Triangulation from three or more of the many targets can give us robot position with a few centimeters. This will position us precisely enough to find a specific tile. The tile positions are known with respect to the shuttle and we can register the tile position with the vision system.

Computing Systems

Three of the on-board computers are VMEbus based real-time systems: a robot controller which controls the base and manipulator motions and monitors the overall health and status of the robot; a vision system which performs the registration and inspection tasks; and a rewaterproofing system which controls the rewaterproofing injection system. The two computer systems which directly control actuator motion (robot controller and waterproofing system) employ "safety circuits" between the computer servo outputs and the motor amplifiers. The fourth on-board system is the High Level Controller (HLC). The HLC is responsible for planning the course of action to complete a given task. In the case of an error or failure in any system, primary safing is performed via the safety circuits, and the HLC performs recovery actions. The HLC also maintains a graphical operator interface.

Electrical Systems

The electronic design is driven by two major constraints: It must 1) run untethered for up to 10 hours, and 2) meet the NEC Class 1 Division II group D requirements for operating in a hazardous atmosphere. Fifteen kilowatt-hours of energy are required to meet the first requirement. Standard gelled lead acid batteries were chosen since they offer good power density. To meet the NEC requirements, all of the electronic enclosures are purged and pressurized, including the battery pack. Additionally, excess heat will be removed from the main electronics enclosure with heat pipes .

REWATERPROOFING SYSTEM

The rewaterproofing system was designed to automate the current manual rewaterproofing process. The system was designed to be fail safe to ensure that tiles were not damaged and that the proper amount of fluid was injected in each tile's rewaterproofing hole(s). It utilizes force control with redundant sensing to ensure that proper contact force is maintained between the rewaterproofing nozzle and the tile surface during the injection process. The nozzle is surrounded by a containment system seal and a slight negative pressure to capture any DMES from a failed injection. The containment system helps to minimizes unnecessary DMES from being vented to the local environment. Process completion is verified through redundant sensing of injection force and DMES injection pressure.

VISION SYSTEM

The vision system has two primary functions. One is to accurately determine the relative position and orientation of the robot tooling with respect to Orbiter tiles. The other is to perform post-flight visual inspections. The vision system uses a two step process to accurately position itself with respect to a tile. First, it uses its laser light projectors to determine the perpendicular distance from the robot tool plate to the tile surface and the orientation of the optical axis with respect to the tile surface. This information is used by the HLC to move the camera to the proper position and orientation so the remaining 3 degrees of freedom can be calculated. These remaining degrees of freedom are calculated with image matching techniques that utilize the current and baseline tile images. The vision system performs visual inspections by comparing pre- and post-flight time images to identify areas in a tile images whose visual appearance has changed. It does this by first aligning the pre and post flight images very accurately. The differences between these images are calculated. These differences are then processed and the differences in the tile's visual appearance are reported to operations personnel. Currently the vision system is capable of identifying missing tile coating and missing pillow type gap fillers.

CONCLUSION

A prototype mobile robotic system for space shuttle servicing has been configured, designed and is currently undergoing system integration and testing. This robot system, when implemented, will mark the beginning of a new era in the ground processing of critical space flight hardware at NASA's Kennedy Space Center.

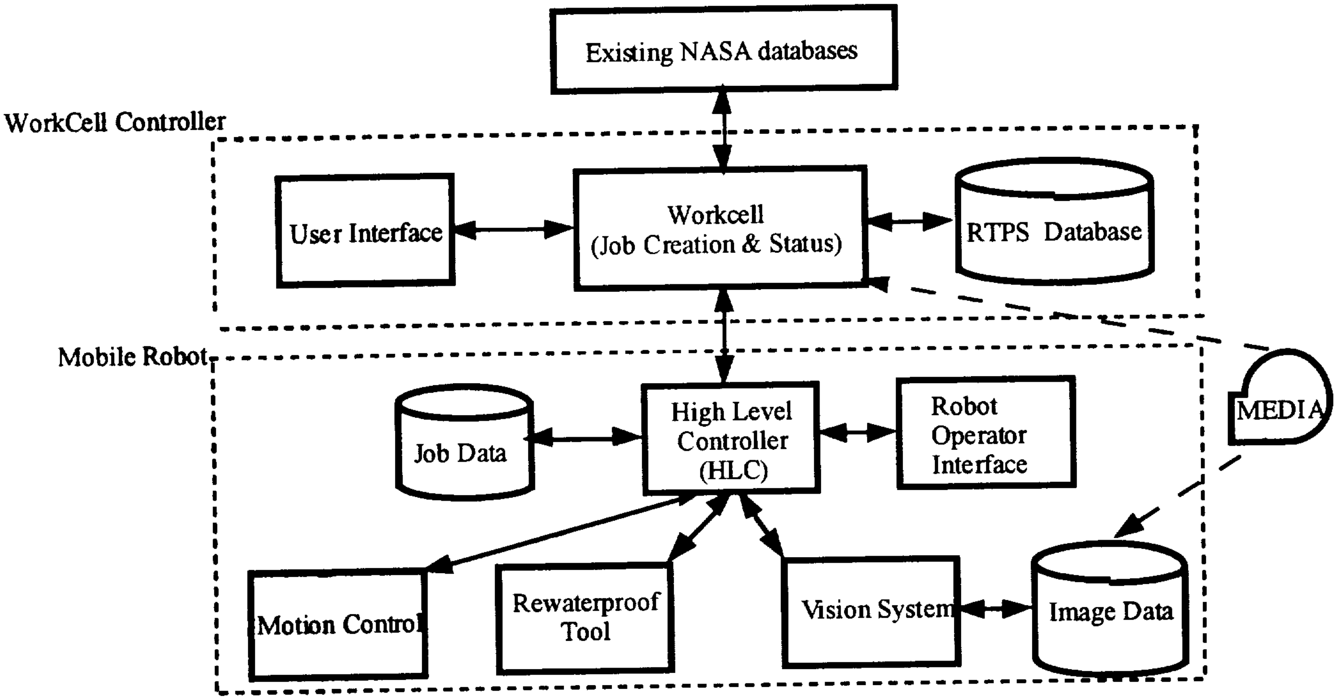


Figure 1 Information System Architecture

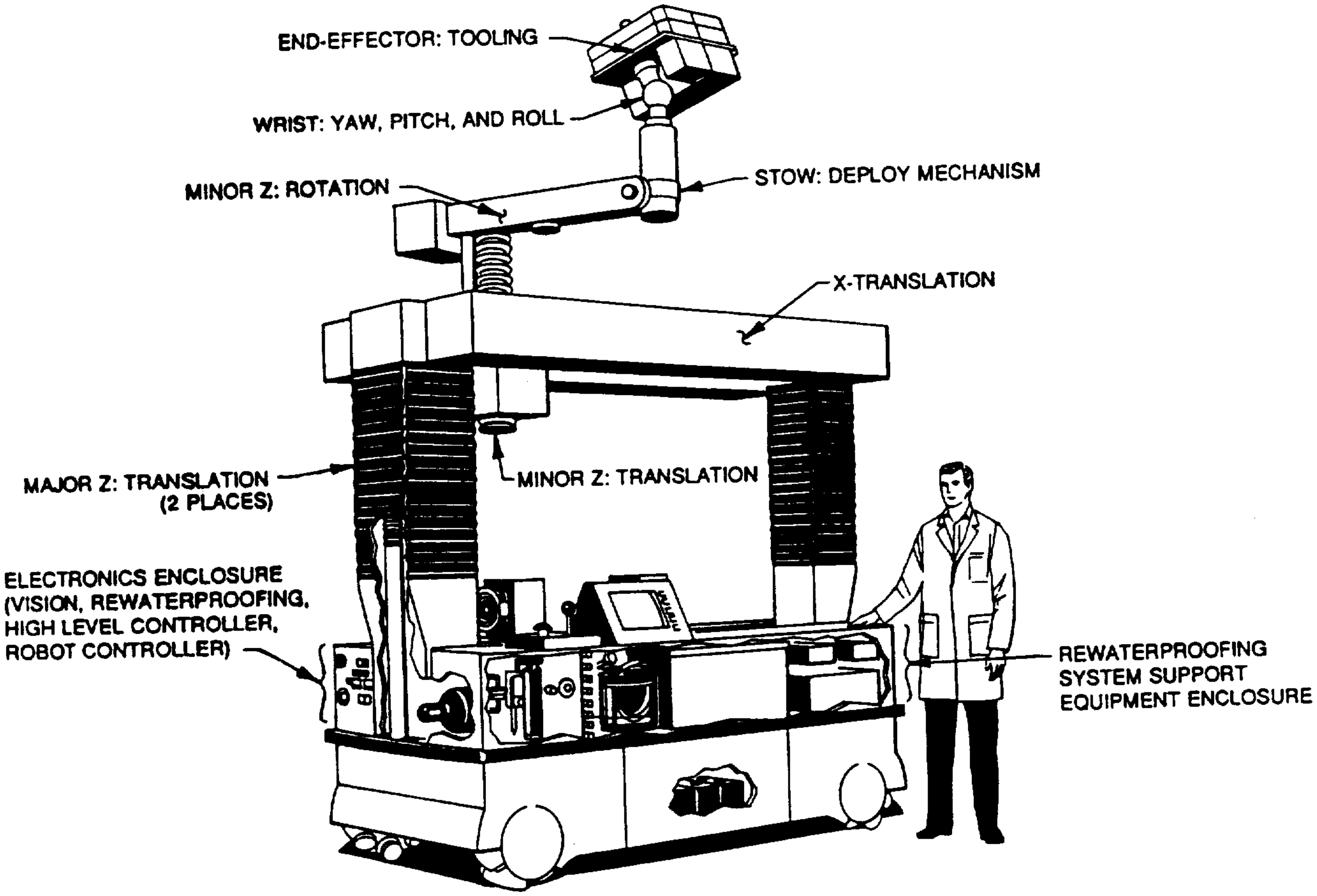


Figure 2 Mobile Robot System

SESSION TT

Teleoperations/Telepresence

Robotics and Telepresence for Moon Missions

Christian Sallaberger[1]
Canadian Space Agency
6767 Route de l'Aeroport
Saint-Hubert, PQ
J3Y 8Y9, Canada
Tel: (514)926-4800, Fax: (514)926-4878

KEY WORDS AND PHRASES

Lunar robotics, mobility, moon missions, telepresence.

MOON PROGRAMME

An integrated moon program has often been proposed as a logical next step for today's space efforts [1,2,3]. In the context of preparing for the possibility of launching a moon program, the European Space Agency is currently conducting an internal study effort which is focusing on the assessment of key technologies. Current thinking has this moon programme organized into four phases.

Phase I of these phases will deal with lunar resource exploration. The goals of this phase of the programme would be to produce a complete chemical inventory of the Moon, including oxygen, water, other volatiles, carbon, silicon, and other resources. A high resolution topographical mapping of the surface of the moon will also conducted. This phase will be accomplished through lunar polar orbiting satellites, possibly equipped with tethered instruments, and a small lander craft. This small fixed lander(s) shall be equipped with a robotic arm to conduct some in situ analysis.

Phase II of the moon programme will establish a permanent robotic presence on the moon via a number of landers and surface rovers. These rovers could continue the chemical analysis, conduct a geophysical survey, and deploy and service various instruments. Some instrumentation would also be located on the fixed landers. Control of these rovers, and the robotic elements of the landers, will generally be handled through remote control from the earth. Telepresence will play a vital role.

Phase III will extend the second phase and concentrate on the use and exploitation of local lunar resources. Automated oxygen production pilot plants, robotic construction investigations, and life support and biological experimentation could all be elements of this phase. In addition to this preliminary astronomical observation is foreseen. A robotic rover might deploy a Very Low Frequency (VLF) Array, probably on the farside of the moon.

Phase IV will be the establishment of a first human outpost. Some preliminary work such as the building of the outpost and the installation of scientific equipment will be done by unmanned systems before a human crew is sent to the moon. Once there, the astronauts will be able to conduct experiments and geological investigations, as well operate the astronomical telescopes and imple-

---

[1] formerly with the European Space Agency

ment the oxygen production plant. To assist the human crew with these tasks, several robotic assets are foreseen.

ROBOTIC MISSIONS

Any near to mid-term European moon programme will undoubtedly be restricted to unmanned missions. One cannot expect the manned Phase IV of the moon programme to begin before 15 or 20 years from now. For this reason the area of lunar robotics and telepresence is considered to be critical.

Missions for lunar surface robotics can be grouped into the following five general profiles:

Simple In Situ Analysis Missions

These missions involve such tasks as operation of imaging cameras, spectrometry, temperature probing, and regolith sample analysis. These missions can generally be accomplished from a fixed lunar lander. A robotic arm attached to the lander could accomplish the tasks of placing sensor heads into the ground, and acquiring small surface samples for analysis by equipment on board the lander. This robotic arm would be controlled remotely from the ground via a telepresence interface to execute its tasks. In a similar fashion the camera pointing and focusing could be accomplished via telepresence.

Instrument Deployment Missions

Scientific Sensors and Stations will need to be deployed at various locations on the moon. These could range from simple thermal probes, to dipoles and seismic stations, to complex telescopes. While small probes could be deployed at a considerable distance from a fixed lander (10s of metres) by harpoon ejection devices and tether instrument deployment crawlers, larger instrument packages will require sophisticated rovers to deploy them at distances up to several hundreds of kilometres from the landing site. Simple deployment functions could occur relatively autonomously, with perhaps supervisory control from the earth. The control of more advanced deployment sequences, such as those involving complex scientific station deployment via a multifunction rover, will call for a more sophisticated control scheme of telepresence by earth-based human operators.

Geological Investigation Missions

These missions will involve the use of mobile rovers to map up terrain over long distances, and also includes the acquisition of samples of interest and the possible return of them to a fixed analysis station, or to return capsule destined for ground laboratories. Due to the investigative nature of this class of missions, human judgement will certainly be constantly required. A good virtual reality interface for the ground based operators is very desirable.

Engineering Support Missions

These missions can be accomplished by a monitoring and servicing vehicle, which will execute such tasks as visual inspection and servicing of installations, selection of suitable landing sites for future missions based on safety criteria, operation of beacon to guide incoming landers or rovers, cargo transportation, communication back-up, etc. Such a monitoring and servicing vehicle will be need both automated capabilities and the ability to be remotely controlled from the ground.

Construction Missions

The final group of robotic missions are those that entail the setup and construction of equipment on the lunar surface. This could be the assembly of communication equipment such as a large, possibly inflatable, dish for ground communication, or an antenna tower for surface communication with rovers. The assembly of the critical elements of a manned lunar outpost before the arrival of the human crew is another task to be accomplished in such missions. Various robotic elements will be required in these construction missions, and various control options will be required. If future manned missions are imminent, capability for future control by crew on the lunar surface should also be considered as a design requirement for these robotic systems.

MOBILITY ISSUES

Most lunar missions will have requirements to move various items from one location on the Moon to another. These items will range from simple experiment packages which have to be deployed at a distance of a few metres from an initial fixed lander, to large volumes of cargo that will be transported from one side of the Moon to the other during advanced base operations.

A critical component of the earlier unmanned segments of a Moon exploration and utilisation programme will be mobile lunar rovers. An analysis and evaluation of possible mobility methods for these rovers has been conducted as a comparative tradeoff between wheels, tracks, and legs as mobility mechanisms [4].

Studies have shown that conical wheels are better suited to climb over obstacles than regular ones, and thus are most desirable for lunar surface vehicles. Wire mesh wheels cause less dust levitation, and therefore are desirable for vehicles carrying instrumentation that is very dust sensitive. Unfortunately these wire mesh wheels also have less grip with the surface. With regard to number of wheels on the rover, six seems to be the optimal compromise which maximises performance criteria, such as manoeuvrability and climbing ability, and minimises complexity of the entire system.

Tracks on the other hand have less surface slip than wheels, and a much higher performance on loose regolith. The disadvantage of tracks is that they have the risk of clogging with lunar dust, as well as having inherent mass and complexity penalties associated with their designs. For these reasons it is not recommended that lunar rovers, which have to operate in the dusty, atmosphereless moon environment, and also should be as reliable and light-weight as possible, be equipped with tracks as their propulsion mechanism.

Legged locomotion is currently a very immature technology, and is not considered to be developed to the level where its use on lunar systems is realistic. However, in theory, legged locomotion could offer good terrain adaptability with high performance in rough terrain and a minimum of locomotion power consumption. Such a system would require active stabilization with sophisticated attitude sensors, and also would require high computing effort for trajectory planning and control. Skis could improve performance on sandy terrain by adding some weight distribution. In general legged locomotion could become the method of choice for lunar surface transportation of the future, but is unadvisable for missions being planned today.

Displacement from one point on the moon to another via mechanical hoppers was also examined, and pogo and anthropomorphic designs were considered. While these concepts are theoretically interesting, the control problems inherent in keeping such systems upright are significant. For this reason such methods are not recommended. Furthermore, if extension to crew systems is attempted, the tolerance of the human vestibular system to the repeated accelerations could prove unacceptable.

Chemical or rocket hoppers were examined, but were found to be only interesting in the context of large displacement for heavy cargo in a mature Moon base(s) scenario. Engine gimballing and throttling will be required. These systems depend on similar technologies as lunar landers, and possibly could be evolved from the technologies developed for a future lander.

Tethered crawlers are interesting as they could offload power and control to a fixed lander while they investigate/deploy instruments close by. Very light-weight crawlers could be built that could deliver a sensor head into the regolith a few metres away from a fixed lander. Tethered probes are also potentially interesting for scenarios where the interior of permanently shadowed crater is to be explored, as the power could be transmitted from a solar array located in the sun on the rim of the crater.

Ejected harpoons could also be used to deploy sensors from a lander. The energy may be delivered by a mechanical, electrical or chemical system. Tethered hooks could be ejected in similar ways, and could assist rovers to climb steep slopes, or escape from loose regolith.

CONTROL ISSUES

Robotic lunar rovers will be a key component of any European Moon exploration and exploitation scenario. These unmanned rovers will certainly encounter unexpected situations, including obstacles and rough terrain. The rover control must be divided between onboard computers, ground computers, and ground based human operators. This division must maximise rover performance, while minimising costs and risks.

Onboard computers have the advantage that they have no communication time delays to the rover, and thus can react to unexpected situations instantly, but have the disadvantage that they have mass and power restrictions, and are physically remotely located, making design errors difficult to rectify.

Ground based computers do not suffer from mass and power restrictions, and thus can carry out much more complex calculations, but have communication time delay to the lunar rover. The round trip time delay is about three seconds.

Control can also be handled by a human operator on the ground. This allows for a maximum of adaptability to unexpected situations, as well as the superior human information extraction capability from visual imagery. Unfortunately the communication time delay is also a handicap for the ground based human operator. Predictive displays could partially overcome this.

The task at hand involves finding the best distribution of the control functionality between the three locations, and assessing relevant technologies.

Four concepts of the distribution of autonomy for the rover have been developed [5], and are being used as a basis for further analysis. They are summarised here:

Concept I: Everything is controlled with the human in the loop. All control is handled remotely by a ground-based human operator, with the sole exception of low-level hardware control which will remain close to the controlled equipment on board the rover.

Concept II: Hazard detection is done autonomously. The detection and the putting of the rover in a safe state is done autonomously. The process of re-planning or hazard avoidance is done by the human operator. A hazard is defined not only as an obstacle, but also shadows, steep gradients, etc. The hazards applicable for a particular rover are dependent on the type of the rover.

Concept III: Trajectory planning is automated. The trajectory planner has as an interface the human generated path segments. Trajectory planning here is defined as the specification of how the path is to be followed in time, as well the conversion from task space coordinates to rover actuator space coordinates (axle speed for wheeled rovers, joint space for legged rovers).

Concept IV: Path planning is automated (i.e. the interface from the human is the specification of the goal location where the rover should go, and the path planning and all lower levels are done autonomously).

The above four concepts do not necessarily identify the place where the autonomous functionality has to be implemented. There remain two possibilities (on board the rover, and in a ground computer), which depend partially on the mission envisaged. While the onboard computer can react instantaneously to sensory input, the ground based computer can be much larger and carry out much more complex calculations.

The optimal control strategy is thus one that distributes control between the onboard computer(s), the ground-based computer(s), and the human operator who can execute either direct or supervisory control.

Virtual reality offers exceptional capabilities to enhance the remote rover control by ground based humans, but is not yet a fully mature technology area. In a virtual reality system, the human operator has complete sensory inputs which give him the feeling that he is (or is in) the remote robotic rover. The operator gives his control inputs in a natural way. For example, if he wants to look to the left, he moves his head towards the left, which causes the cameras on the rover to point to the left, and subsequently for the correct image to be projected on the head mounted display worn by the operator. Such systems allow for a very high or total sense of immersion for the operator. Initial analysis has identified 300 kbit/s as the approximate bandwidth required for ground based control via a virtual reality type interface. This assumes stereo vision with advanced compression ratios of 10, and relatively low resolution video with 3 to 5 frames per second.

The round trip communication time to the Moon is limited by the speed of light. The minimum time is about 3 seconds. This makes realtime control of lunar rovers from the ground awkward and slow. One possible area that might form a partial solution to this is predictive display technology. The computer generated displays could predict the view from the rover three seconds ahead, based on an internal map, and the current motion of the rover. This technology area is still in the early research phase both in Europe and outside.

CONCLUSIONS

Robotic missions which form part of a moon program would typically involve such tasks as geological surveying, instrument deployment, and sample acquisition and analysis. The issues of mobility and control will be critical ones. The mobility technology used by the robotic system will depend on the task requirements. Wheeled locomotion is generally the preferred option for lunar rovers. Fixed robotic landers could use ejected harpoons or tethered crawlers to deploy sensor heads in the area surrounding the lander. The optimal strategy for any lunar robotic asset will involve distributed control, utilizing both human ground-based operators, and artificial intelligence located in various terrestrial and lunar computers.

REFERENCES

[1] *Mission to the Moon*, ESA Publication SP-1150, June 1992.

[2] *Feasibility Study of a Lunar Exploration Programme based on Unmanned Orbiter / Lander Spacecraft*, Christian Bank, Estec Working Paper 1676, August 1992.

[3] *Lunar Outpost*, Advanced Programs Office, NASA, JSC-23613, 1989.

[4] *Lunar Surface Displacement Methods and Techniques*, C. Sallaberger, ESA/Estec internal document # 94007-EST0178, January 1994.

[5] *Control of Unmanned Lunar Rovers*, C. Sallaberger, ESA/Estec internal document # 94007EST0202, January 1994.

Truss Structure Tele-Manipulation Experiment using ETS-VII

N95-23726

Kohtaro Matsumoto, Seishiroh Kibe, Isao Yamaguchi, Takashi Kida,
Sachiko Wakabayashi
National Aerospace Laboratory
7-44-1 JINDAIJI-HIGASHI-MACHI CHOFU, TOKYO 182, JAPAN
PHONE +81-0422-47-5911
FAX +81-0422-49-8813
E-mail : matumoto @asuka.aerospace-lab. go.jp

Hiroshi Ueno, Hitoshi Sato, Shigeru Aoki, Tetsuji Yoshida
Shimizu Corporation
SEVANS South 1-2-3 Shibaura, Minato-ku, Tokyo 105-07, JAPAN
PHONE +81-3-5441-0085
FAX +81-3-5441-0346
E-mail : hueno @ spo. shimz. co.j p

Key Words: Space Robot, Truss assembly, Space Construction, Tele-operation

Abstract

In this report, a robot experiment concept of space truss tele-manipulation by National Aerospace Laboratory (NAL) will be described in its flight model development. The experiment will be carried out on the Engineering Test Satellite No. 7 (ETS-VII) using its robot arm. The satellite is scheduled to be launched in 1997 by National Space Development Agency of Japan (NASDA). The truss flight model is composed of deployable truss system and assemble truss joint. Those truss components will be manipulated by the ETS-VII robot arm using its small Grapple Fixture type-N (GPF-N), and the experimental task operation will be executed from the ground control station .

1. Introduction

Future orbital space systems are going to be larger and more intricate in their structure. Such future space structure will be obliged to consider assemble mechanisms in space, instead existing ground-assembled systems. Space truss system will play a major part of such future on-orbit assemble systems with its high transportation performance in smaller packing volume.

For the space truss construction, two kinds of task ---link mechanism structure deployment and strut joint connection--- are considered to be an initial research issue at NAL. Recent space systems are made of rather simple deployable components for one dimensional deployment, but the future large deployment structures will take more complicated mechanism for more intricate configuration with two or three dimensional deployment.

Construction tasks for these systems will require dexterity of human or advanced autonomous systems. NAL believes the ETS-VII (Fig. 1) mission could be the first step for space truss construction.

2. Truss Experiment

NAL experiment preliminary design for ETS-VII was completed in 1993 and experiment scenario and engineering model development has begun in this April. Below is major outcomes in the preliminary design phase.

2.1 Experiment Components

On the preliminary design phase, experimental components were examined in its performance required for robot arm and space qualification. The components are to be deployable truss system and assemble truss joint independently on the task board. ETS-7 robot arm will execute truss handling motion using its grapple fingers at the end of the arm. The arm has cameras to take visual information of robot tasks robot.

2.2 Truss System

Truss system is composed from several systems as follows;
- Deployable truss system
- Assemble truss joint system
- Launch lock system for the truss
- Task board
- Target markers for camera
- Telemetry devices (Thermisters, etc.)

Deployable truss system has rigid triangle truss and one set of struts connected each other by one degree of freedom (dof) hinge. Task board top surface has every truss system and is designed to be inclined in 20 degrees to make the arm's work smooth while robot is accessing and handling the truss. (Fig. 2, 3)

2.3 Truss Experiment Plan

Scheduled NAL experiment plan includes robot motions as follows;
- fine motion to handle small work
- follow pre-determined track
- grasp and releasing motion of the truss

For tele-manipulation from ground station, NAL is planning to introduce control technology as follows;
- time delay compensation
- graphical information processing on the ground station
- autonomous control architecture with hierarchical structure of robot task components
- operation supporting by computer models

3. Design and Development Status

3.1 Truss Operation

To confirm the function and feasibility of the truss experiment, NAL prepared the BBM (Bread Board Model) of the deployable truss and the assemble truss for its ground testbed. The robot was refined from an industrial robot so that it directly controls the angular velocity (25 ms). The robot performance is far better than ETS-VII robot arm, and it will be necessary to adjust parameters close to ETS-VII arm's in the future.

The arm is expected to use impedance control to absorb the position errors on trajectory in space. For the truss deployment test on the ground testbed, the impedance control for the three axis of both translation and rotation was applied, because the required operational force for the arm is low and the deployment 3-D trajectory is complicated.

3.2 Assemble Truss Joint

As the most of proposed space joints need twisting motion by an astronaut's hands, they may not be suitable for one hand arm task such as ETS-VII arm. For one-hand operation, Star*Bay mechanism has been introduced to NAL truss joint.

In order to use truss joint in space, it is obvious the joints have to maintain operational force lower than maximum arm force. Lever and wedge mechanism are introduced to NAL truss joint to make its force lower. The joint operation motions are; A) to insert and fix the joint into node, and B) to latch the mechanism at the beginning and end of A). A) is achieved by applying the sliding force of the robot arm, while B) is by twisting the arm.

3.3 Grasping Fixture-N (GPF-N)

GPF-N (Fig. 4) is specially designed to grasp small NAL truss system (40 mm dia. pipe). The size is approximately one thirds of standard grasping tool for ETS-VII arm (150 mm dia.).

3.4 Robot Experiment Panel

The truss system is mounted on the robot equipment panel of the ETS-VII robot mission (Fig. 5, 6). The truss location and configuration was fixed by simulation study avoiding collisions considering arm operation clearance (25 mm-45 mm) and the arm joint angle range. The range satisfies 5 degrees margin on every axis, except in the case of emergency.

3.5 Ground Operation System

The ground operation systems for ETS-Vll, including NAL mission, will be built in NASDA's Tsukuba Space Center. NAL is planning to use NASDA's station for critical part for communication and command operation. The ground station will have functions as follows,

(1) hierarchy control teleoperation
(2) image processing and measuring
(3) orbital simulation image display
(4) tele-manipulation support information display
(5) operation and collision simulation
(6) joystick interface
(7) control and data interface to the arm

4. Future work

4.1 Development Schedule

By the fall of 1994, STM (Structural Thermal Model) and tele-operation model will be delivered to NASDA. The truss PFM (Proto-Flight Model) delivery will be in the fall of 1995. (Fig. 7)

For the limited experiment time schedule in space, NAL, NASDA and other agencies are working how to share the time, as the ETS-VII mission life is designed to be around 1.5 years.

4.2 Technical Issues

It is obvious NAL's small truss system requires higher performance in positioning and trackability however, the arm stability and capability are designed for more rough and tough space tasks using its power. Trade off study adjusting performance of the truss and of the arm is now going on.

Supporting systems on the satellite and on the ground are also being studied to relax the severe operational conditions.

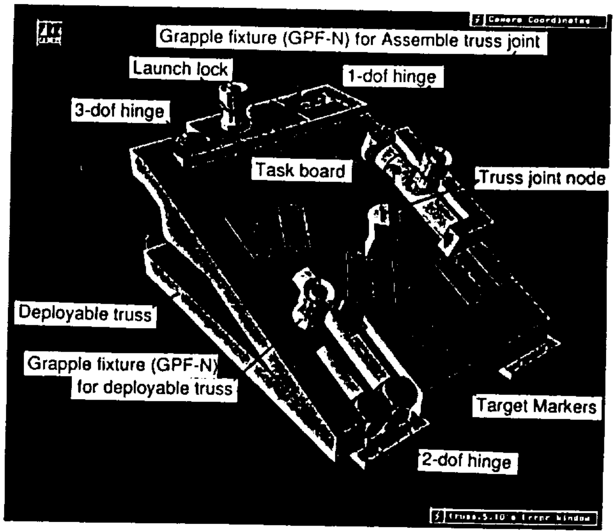


Fig.2 Truss configuration for Launching

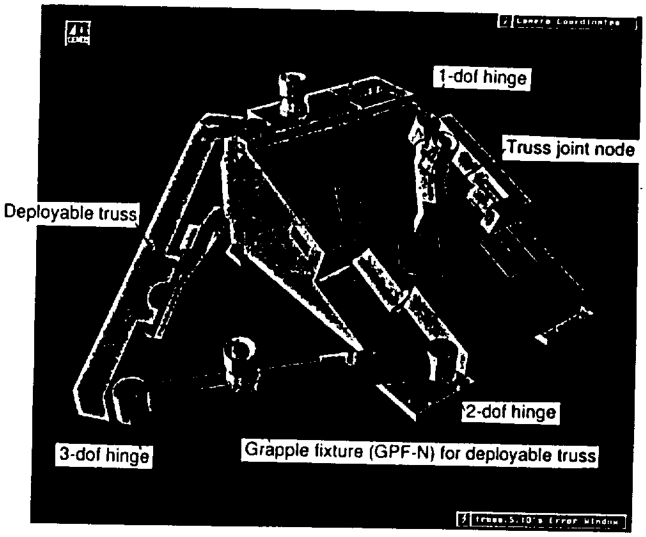


Fig.3 Deployed configuration

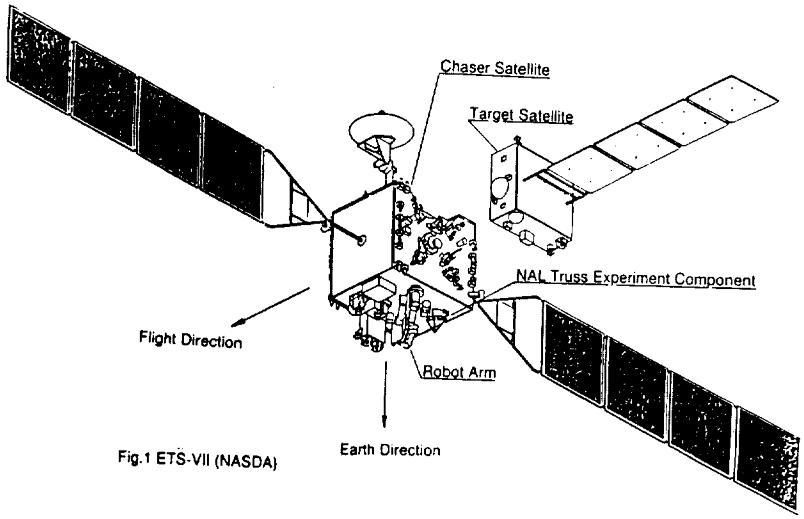


Fig.1 ETS-VII (NASDA)

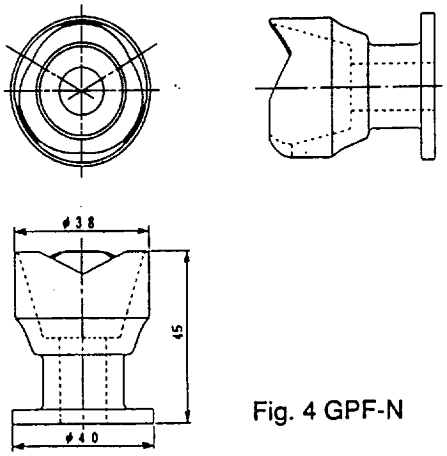


Fig. 4 GPF-N

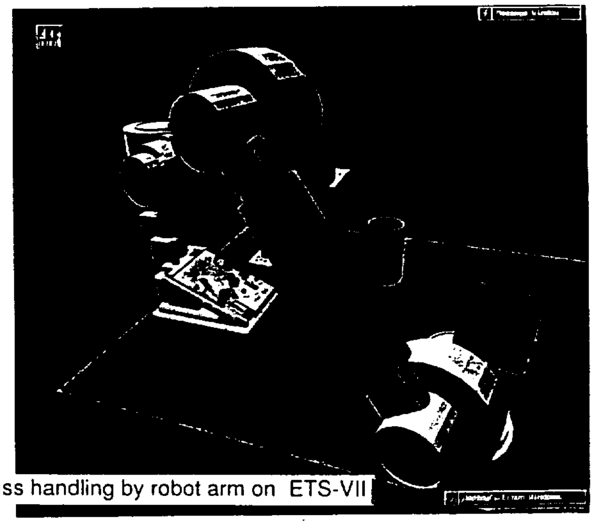


Fig. 5 Truss handling by robot arm on ETS-VII

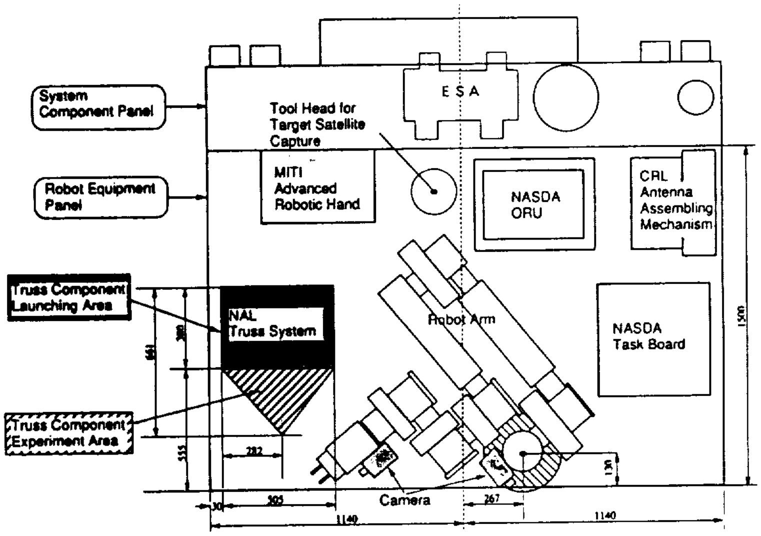


Fig. 6 NAL truss system on the ETS-VII robot equipment panel

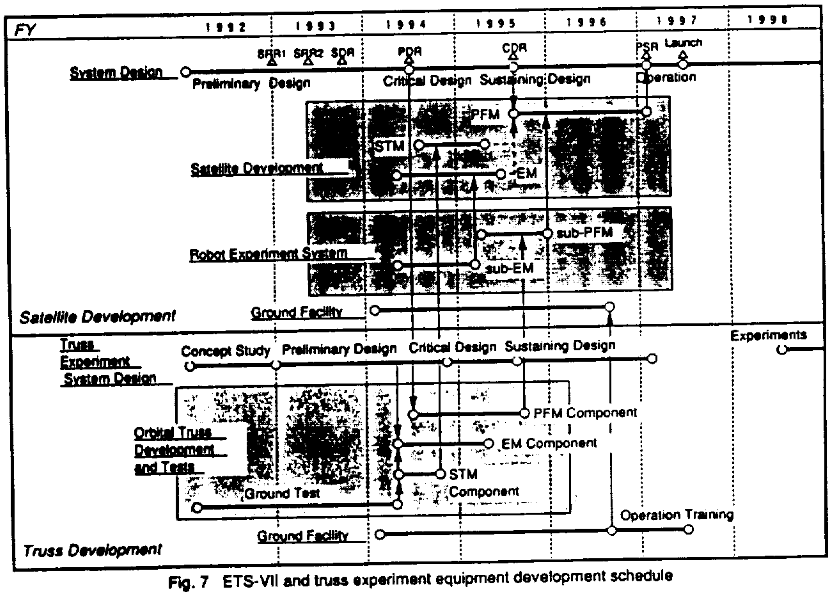


Fig. 7 ETS-VII and truss experiment equipment development schedule

Antenna-Assembling Mechanism Test on ETS-VII

N95- 23727

Yoshiaki Suzuki, Shinichi Kimura, Tetsuo Takahashi,
Communications Research Laboratory
4-2-1, Nukui-Kitamachi, Koganei-shi,
Tokyo 184, Japan
Tel 81 423 27 7512 Fax 81 423 24 8966
E-Mail Address ryo@crl.go.jp
Kazuo Nakamura, and Hajime Morikawa
NEC Corporation
4035, Ikebe-cho, Midori-ku,
Yokohama 226, Japan

KEY WORDS AND PHRASES

Engineering test satellite, large space antenna, antenna assembling, teleoperation, telerobotics

ABSTRACT

The Communications Research Laboratory plans to test an antenna-assembling mechanism on the Engineering Test Satellite VII. The test is one of the application missions for the space robotics experiments that will be conducted mainly by the National Space Development Agency of Japan (NASDA). The purpose of the test is to verify the ability of the antenna assembling mechanism to function in space and to experiment on the teleoperation of a space robot to develop antenna-assembling technology. In this paper, we present the test experiment plans and the outline of the onboard assembling mechanism.

INTRODUCTION

Assembling antennas by means of a robot is one method to build large-scale dish-type space antennas for high frequency applications. Assembling-type antennas have possible advantages over deployable-type antennas to achieve highly accurate reflector-surface construction and is appropriate for high-frequency applications. [1] The Communications Research Laboratory (CRL) planned the antenna-assembling experiment on the Japanese Experimental Module (JEM) of Space Station and started developing assembling-type antennas in 1986. At first, we developed the assembling mechanism which couples the center panel with the divided peripheral panels. The mechanism is a key component for assembling antennas easily in space. A smart mechanism makes antenna construction possible using a robot arm instead of Extra Vehicular Activity (EVA). Two antenna scale models with different types of mechanisms are developed [2] and tested first on the ground using a robot arm. Before the experiment on JEM, we planned a precursor experiment to test the mechanism on the Engineering Test Satellite-VII (ETS-VII) [3], which will be launched in 1997. The purposes of the test on the ETS-VII are to verify the ability of the assembling mechanism to function in space and to experiment on the teleoperations of a space robot to develop antenna assembling technology.

ASSEMBLING-TYPE ANTENNA

Figure 1 shows the configuration of the assembling-type antenna. The main reflector is divided into 8 peripheral panels and a center panel. A sub-reflector is attached to the center panel with four stays. The assembling mechanism is used to couple the center panel and the divided peripheral panel, as shown in the figure.

In the JEM experiment, construction of a 2- to 5-meter diameter antenna is planned as the first step. Initially, the center part attached behind the center panel, which contains RF compartment and pointing equipment, will be assembled on the exposed facility of the JEM. Next, divided panels will be attached to the center panel, one by one, by using the Japanese Remote Manipulator System (JRMS).

EXPERIMENTAL SYSTEM FOR ETS-VII

CRL plans to test the antenna-assembling mechanism on the Engineering Test Satellite VII (ETS-VII). In this experiment, only the assembling mechanism is to be tested because it is the key component of the assembling-type antenna. Testing the assembly performance under various conditions (described later) as well as its durability in space is an important objective of the experiment.

Another objective is testing the teleoperation technology under the effects of communication delay and the limited communication capacity caused by the long distance from the operator on earth to the assembly site in space.

Figure 2 shows the experimental system block diagram including the onboard system and the earth control system. CRL will develop hatched equipment: an onboard antenna-assembling mechanism and a ground auxiliary teleoperation system for the experiment.

Assembling Mechanism

The antenna-assembling mechanism is composed of a fixed part (FP) which contains the mechanism for the center panel and a coupling part (CP) which contains the mechanism for the divided panel as shown in Fig. 3. The FP is attached to the satellite main body structure. The coupling mechanism used is a rotary hook-type latch actuated by a spring force (Fig. 4).

To make the assembly procedure easy and secure, we introduced both a mechanical guide system and a visual guide system. The mechanical guide system consists of a guide cone and a cone receptacle which mechanically compensates for the positioning error in the assembling process. The visual guide system uses a three-dimensional target mark (Fig. 5) attached to the FP and a hand camera system attached near the robot hand. The image of the target is transmitted to the ground control system and is used to determine the relative position and attitude of the CP to the FP. This information is used to teleoperate the onboard robot arm. The compliance mechanism in Fig. 3 is introduced to absorb the possible reaction force induced by the mechanical contact of the FP and the CP.

Teleoperation System

The teleoperation equipment consists of a workstation and an image processor, and functions as the robot arm control, image processor and robot operation simulator. Teleoperation commands generated in the equipment are transmitted to the satellite via NASDA's satellite control facility.

Figure 6 shows the block diagram of CRL's teleoperation equipment. The basic system consists of a teleoperation computer, an image processor, a video processor, and a monitor. The teleoperation computer is used for arm control calculations and data communication. The video processor is used for processing the target-mark image. The image processor is used for overlaying the camera image on the CRT.

The teleoperation auxiliary computer is used only for the predictive bilateral control experiment described later.

The basic system software consists of a teleoperation manager, a simulation module, a visual simulation module, and an interface module. Each module works as an independent parallel process. The teleoperation manager communicates with other modules and exchanges parameters and data. Almost all operations such as the menu selection, parameter change, and command transmission, can be done by using a mouse on a graphical user interface (GUI).

The teleoperation manager functions are
*robot teleoperation,
*data management, and
*controlling other modules.

The simulation module functions are
*robot movement simulation
*3-D wire frame simulation-image display of the robot and the assembling mechanism, and
*pre-operation check.

The visual simulation module functions are
*image processor control, and
*video processor control.

The data which will be acquired from the experiment are
*torque and force data of the arm,
*position of the arm (angle of the joints),
*video image of the CCD camera,
*target position calculated from the video image, and
*operation duration time.

TEST EXPERIMENT PLAN

Several kinds of assembling and disassembling experiments are planned:

Basic Assembly and Disassembly Experiment

Repeating the assembly and disassembly operation while changing the following parameters:
* Operation mode of the arm (teaching/manual mode)
* Operational speed of the arm
* Insertion force of the coupling
* Control mode of the arm (position control/force control mode)

Allowable Positioning Error

The FP and the CP can be assembled even with a certain positioning error using the mechanical guide cone. In this experiment, the allowable positioning error is measured.

Assembling and Disassembling with Intentional Disturbance

Sine-wave/random positioning error will be added to the robot command to simulate the assembly using a long-armed manipulator with excess vibration.

Fully-Automatic Assembling Experiment

The relative position of the FP and the CP is calculated using visual feedback of the target mark images. The position data enables automatic assembly of the FP and the CP.

Predictive Bilateral Control

A virtual model of the operation target is constructed in real-time. The virtual image and the reaction force calculated from the model is supplied to the operator who controls the master arm while watching the virtual image with the superimposed real image delayed by the distance.

CONCLUSION

The plans of the assembling mechanism test on ETS-VII and the outline of the onboard mechanism is presented in this paper. The experiment will demonstrate the possibility of constructing large-scale dish-type antennas using a space robot controlled by a ground operator. The actual antenna construction in space with this type of assembling mechanism is planned for the space station using JEM's remote manipulator system.

The onboard mechanism for the ETS-VII is now under the critical-design stage and a BBM is under construction. Before the ETS-VII experiment, a ground simulation test using the BBM and the NASDA's test bed system will be done.

ACKNOWLEDGEMENTS

We would like to thank the members of ETS-VII group in NASDA for their great effort and help in including our mission on the ETS-VII Project.

REFERENCES

[1] T. Iida, K. Okamoto, Y. Ohkami, S. Kibe, H. Koshiishi, M. Naka, and H. Yamamoto, : "A Space Station Experiment on Large Antenna Assembly and Measurement," Acta Astronautica Vol.12. No. 5, pp.335-344, 1985.

[2] Y. Suzuki, T. Takahashi, M. Tanaka, T. Iida, S. Yokota, and K. Ohshima, : "Scale Model Development of Assembling Type Space Antenna," 42nd Congress of the International Astronautical Federation, IAF-91-314, October 5-11, 1991, Montreal, Canada.

[3] M. Oda, Y. Wakabayashi, S. Ichikawa, R. Imai, and T. Anzai, : "ETS-VII, The World First Telerobotic Satellite (Mission and Its Design Concept)," Artificial Intelligence, Robotics and Automation, in space, Toulouse, September 92, pp307-318.

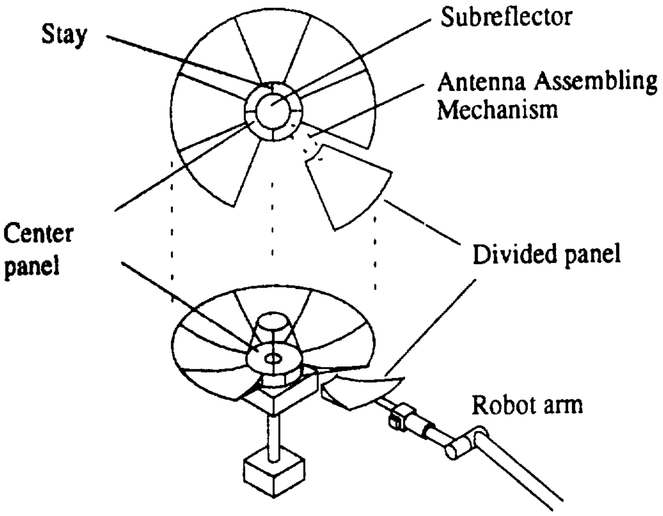


Fig. 1 Assembling-type antenna and assembling mechanism.

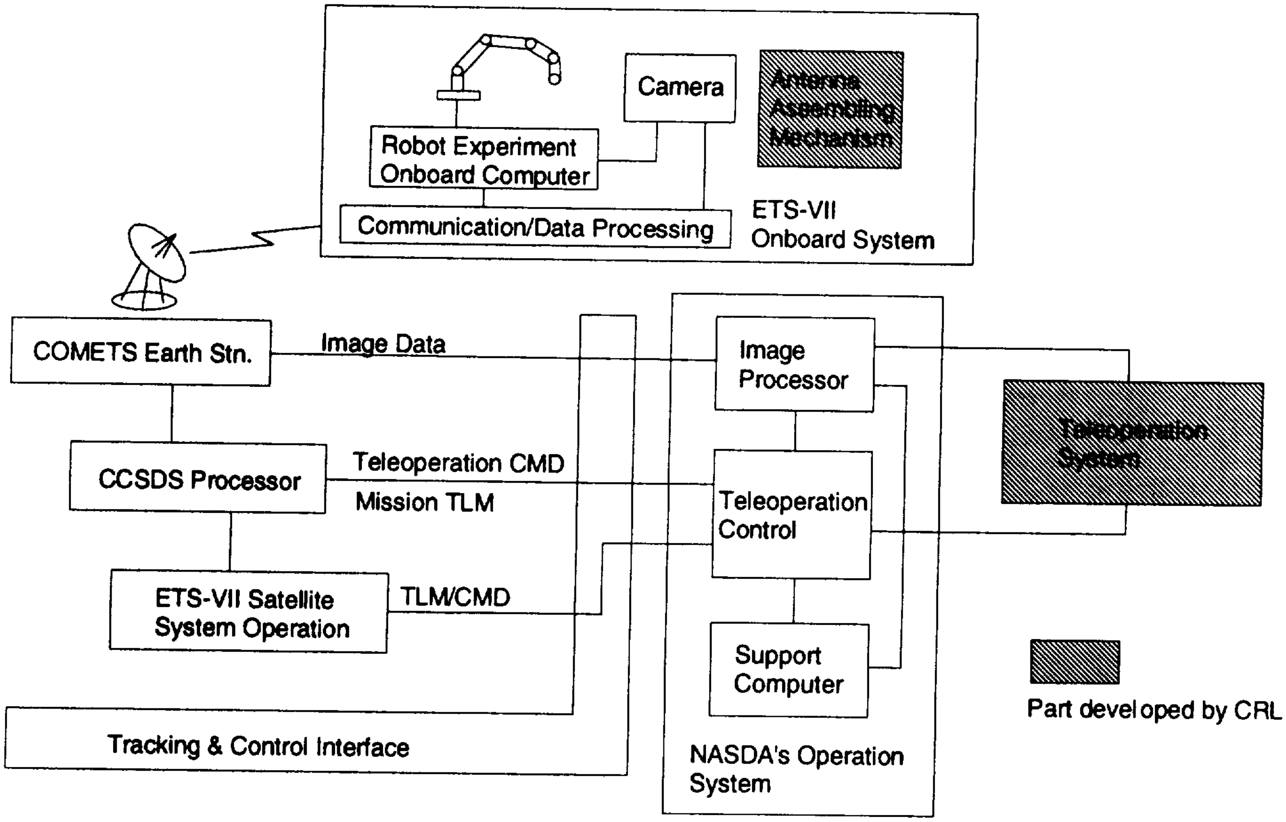


Fig. 2 ETS-VII robotics experiment system block diagram.

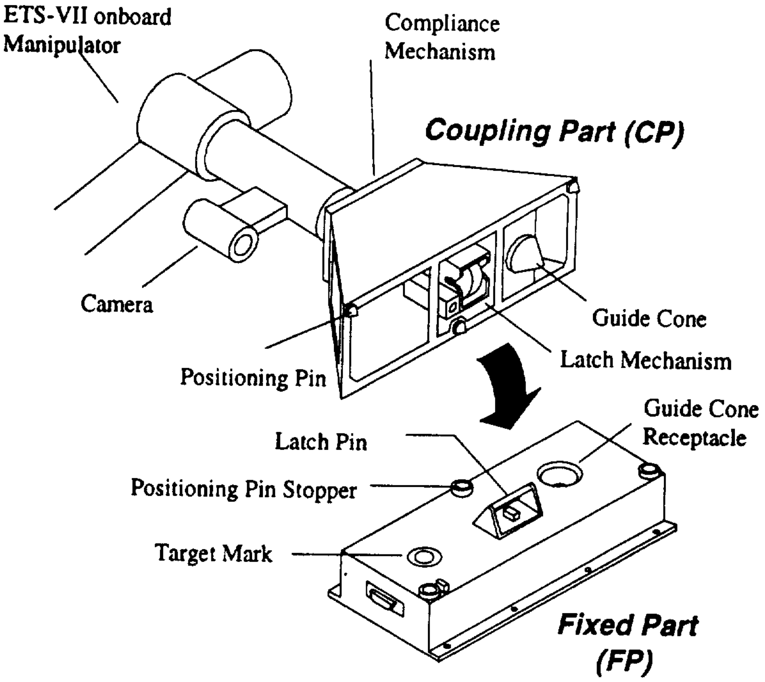


Fig. 3 Onboard antenna-assembling mechanism

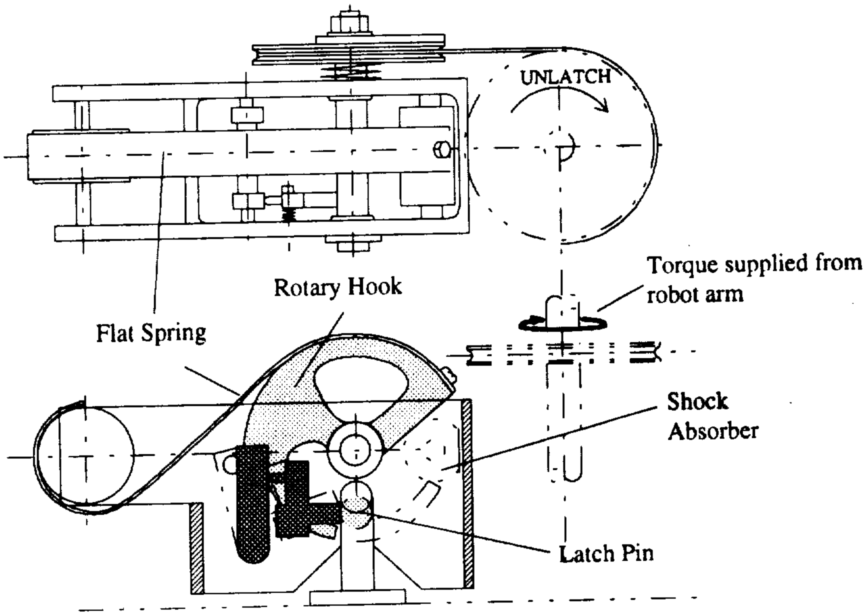


Fig. 4 Structure of the latch mechanism.

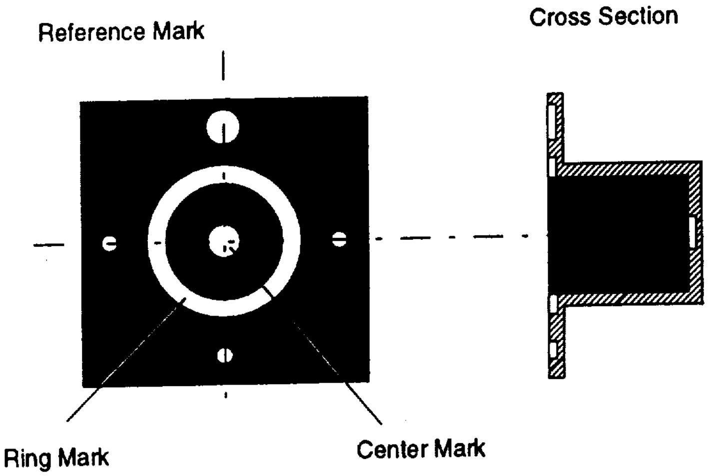


Fig. 5 Target mark for the visual guide

Intelligent Monitoring System Applied to Super Long Distance Telerobotic Tasks

Yujin Wakita*, Shigeoki Hirai, Kazuo Machida

Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba-shi, Ibaraki 305, JAPAN

*tel:+81298585969, fax:+81298585989, wakita@etl.go.jp

N95-23728

KEYWORDS AND PHRASES

Communication Time Delay, Intelligent Monitoring, Knowledge Base, Small Capacity of Communication, Telerobotic System.

ABSTRACT

Time delay and small capacity of communication are the primary constraint in super long distance telerobotic systems such as astronautical robotic tasks. Intelligent telerobotics is thought to break this constraint. We aim to realize this super long distance telerobotic system with object handling knowledge base and intelligent monitoring. We will discuss about physical and technical factor for this purpose.

INTRODUCTION

Telerobots such as space telerobotic systems use both autonomous and direct human control (manual control) in execution of their tasks. Supervisory control is a well known concept applied to these hybrid systems. From the viewpoint of flexibility and human friendliness in telerobotic task execution, we have proposed a more cooperative way to effectively utilize both autonomous functions of the robot and direct maneuvering by the human operator and developed MEISTER system[1].

In the telerobotic task execution visual information, such as TV monitors, is most important for cooperation between robots on a remote site and a human operator on a local site. However, in order that TV monitor supports a human operator effectively, the monitor should display scenes relevant to task situation. In the conventional systems, a human operator must control camera direction and viewing angle (zooming) manually along with robot task control. It increases burden of the operator severely. Therefore we have proposed intelligent control system of monitoring camera for telerobotic task execution[2].

First we will describe the principle of the intelligent monitoring system briefly, second discuss special issues of telerobotics executed over super long distance, and then show possible extension of the current intelligent monitoring functions for the issues.

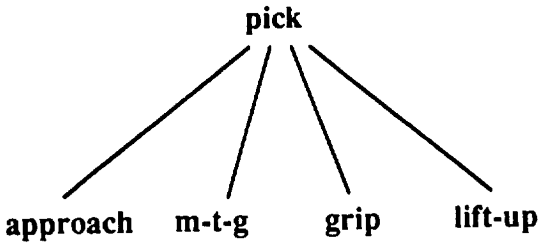
INTELLIGENT MONITORING

The MEISTER system has a collection of task oriented object models as the knowledge base[3]. Each model contains environmental data which work as a world model. Handling knowledge, both generic and specific, is described by methods attached to object classes.

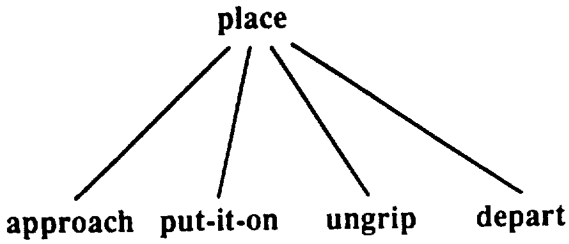
The human operator achieves cooperation by watching the robot motion through TV monitors. Whether the human operator can help robot effectively or not depends on whether or not the TV monitors display appropriate scenes of task executions. In the original MEISTER system, the human operator controls viewing-point (camera direction) and viewing-angle (zooming) of the camera manually. We found that this controlling operation is the busiest part of the operator's task. Considering the problem, we have introduced intelligent monitoring in telerobotic task execution[2]. We call 'intelligent' to mean that a robot autonomously reports to the operator selected information relevant to a given task.

The basic concept for the intelligent monitoring is based on the observation that we control our view according to what to see, when to see, how to see. These strategies seem to have deep relation with structure of manipulation tasks.

In the knowledge base of MEISTER, a pick operation is expanded into lower level motion commands such as 'approach,' 'm-t-g(move-to-grip),' 'grip,' and 'lift-up' motions depicted in Fig. 1. Differences in the meaning of these motions should correspond to different control strategies for monitoring action. This implies understanding the contents of the task and how the task is proceeding. We apply this concept to the control of viewing direction and the viewing angle of the monitoring cameras so that the human operator can receive appropriate information to cooperate with the robot. The details of the strategies of camerawork for each motion are as follows.



(a) Expansion of "pick" operation.



(b) Expansion of "place" operation.

Fig. 1. Expansion of pick-place operation.

approach
In this motion the robot hand approaches an object. This is a kind of global motion. The hand moves straight to the destination with a certain speed. If an obstacle exists on the path, the operator should stop the robot or control the robot to avoid it. So in approach motion the monitoring camera should catch the whole area of the motion or follow the robot hand with viewing angle as wide as possible.

m-t-g(move-to-grip)
In this motion the robot hand moves to the grasping position of an object. It is a kind of guarded motion. Since the robot hand is already close to the object, the operator wants to look at the robot fingers and the object closely so that he or she can check their relative position. Therefore the camera should zoom in on the hand and the object.

grip
In this motion the robot hand does not change its position but its fingers close to hold the object. The aim of camera control is almost similar as the m-t-g motion. Further closing up helps the operator to confirm that the fingers hold the object successfully.

lift-up
The hand goes up into the free space to prepare next approach to another target. The camera should zoom out smoothly expecting this motion.

put-it-on
The hand sets the object in hand to the destination place. The camera should zoom in to cover the object in hand and the other one to which the former one will be assembled.

ungrip
The hand releases the object in hand. The camera is centered to fingers so that the operator can confirm that the object is successfully releases from the hand.

depart
This motion is similar to lift-up motion. The camera should zoom out smoothly covering current hand position and the point in the free space the hand goes to.

PROBLEMS TO BE ATTACKED

Time delay and small capacity of communication are the hardest constraint in super long distance telerobotic systems. Direct power or position feed back loop between local and remote sites is not realistic because the time delay will be a few ten seconds or a minute. We cannot construct an efficient servo loop between local and remote sites. The problems are summarized into two points.

Commanding Level

An operator needs to command to robots with some high level robot language. If the commands which are sent from a local site to a remote site are much abstract, amount of communication will be decreased.

On the other hand too high level commands are not sufficient to let an operator and robots execute tasks cooperatively and such a telerobotic system is not effective. Therefore it is also an important theme to determine command level corresponding to the degree of time delay of communication.

Information Selection

The operator should achieve not only commanding the robot but also watching the task environment with monitoring camera and various sensors in order to cooperate with the robot to execute tasks. So bi-directional communication is necessary between a local site of an operator and a remote site of robots.

However not only time delay but also capacity of comunication are under constraint in super long distance telerobotic systems. It is not expected that all information can flow incontinently from the remote site to the local site. The remote system itself needs to select the information important to cooperate for the operator and send it to local site.

MONITORING FUNCTIONS

Basic strategies of intelligent monitoring described in the previous chapter will not be sufficient for the communication channel constraints problem. We propose following extensions for it.

snapshot function
Selecting and sending only important scenes when all the images can not be sent. Selection of viewing angles and viewing ranges in a sequence of task execution should be also included.

simulation function
Showing graphically simulated task procedure to give expected images of task status between the snapshots. These expected images help the operator to prepare response when the next new scene is displayed.

confirmation function
Confirming task status on each step using comparison of expected and real image on remote site. Though this is not directly monitoring, it can be seemed as an extension of monitoring of task procedure by robot itself.

recording of whole video
Storing whole scenes on remote site and send it to local site without any omission after the execution for analyzing errors later. This is a kind of telemetry.

EXPERIMENT PLAN

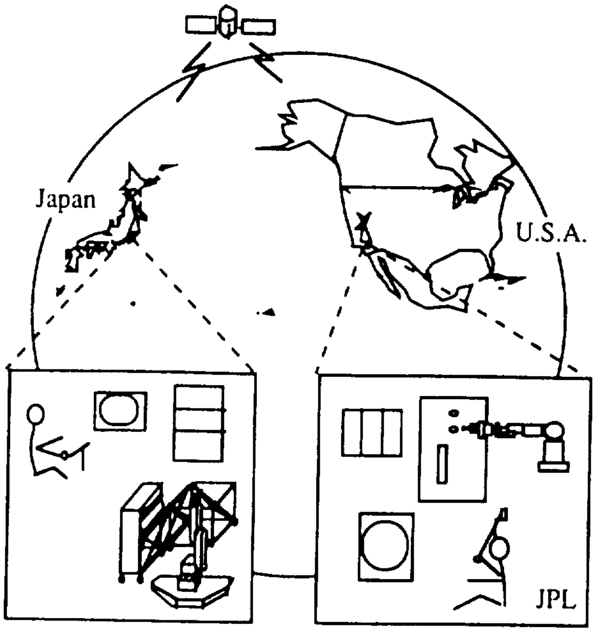


Fig. 2. Over pacific telerobotic operation.

Research topics described in this paper are for the collaborative research of ETL and JPL. In the collaborative research we plan to operate mutual telerobot testbeds (Fig. 2).

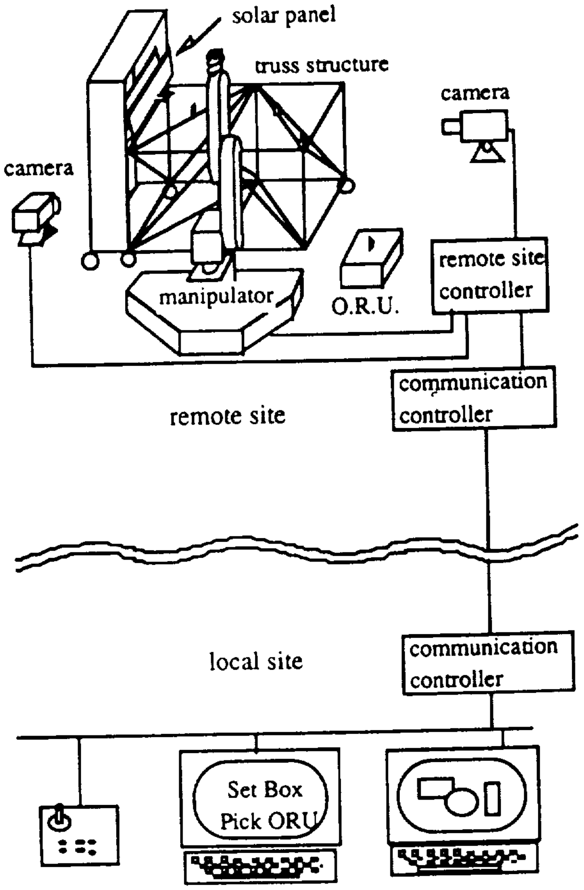


Fig. 3. Testbed plan at ETL.

The outline structure of the experimental telerobot testbed under construction at ETL is illustrated in Fig. 3.

Remote site subsystem includes a manipulator to execute task and monitoring camera(s) to monitor it. An image processing board is used to grab video images for monitoring, compress it to send to the local site.

Local site subsystem includes an interface for object handling knowledge-base and graphics on UNIX workstations. Direct control device such as joy-stick or master-manipulator will be also included for emergent intervention.

For connection line between local and remote sites, we plan to try several ways such as inter-network, ordinary telephone-line with ISDN and/or conventional modem connection. These are to study about influence of quality of communication to telerobotic task execution.

CONCLUDING REMARKS

We discussed problems of super-long distance telerobotics, and plan to extend and apply the intelligent monitoring system. KHI (Kawasaki Heavy Industry Co.) collaborates with us to construct the testbed for this experiment. Detail of the construction of the testbed is presented in another paper.

REFERENCES

[1] Hirai, S., Sato, T. and Matsui, T., 1992. Intelligent and Cooperative Control of Telerobot Tasks, *IEEE Control Systems*, 12(1): 51-56.

[2] Wakita, Y., Hirai, S. and Kino, T., 1992. Automatic Camera-work Control for Intelligent Monitoring of Telerobotic Tasks. In *Proceedings of the IEEE Intelligent Robots and Systems (IROS'92)*, 1130-1135.Raleigh, North Carolina, U.S.A.

[3] Hirai, S. and Sato, T., 1988. Object Model for Telerobot. In *Proceedings of the IEEE Intelligent Robots and Systems (IROS'88)*, 701-706. Tokyo, Japan.

Manipulating Flexible Parts Using a Teleoperated System with Time Delay: An Experiment

N95- 23729

T.Kotoku, K.Takamune, K.Komoriya and K.Tanie
Mechanical Engineering Laboratory, AIST-MITI.
1-2 Namiki, Tsukuba, Ibaraki 305 Japan
Tel: +81-298-58-7284, Fax: +81-298-58-7201
E-mail: toku@mel.go.jp

N.Matsuhira, M.Asakura and H.Bamba
R & D Center, Toshiba Corporation
4-1 Ukishima, Kawasaki, Kanagawa 210 Japan
Tel: +81-44-288-8026, Fax: +81-44-288-8210
E-mail: 000091050512@tg-mail.toshiba.co.jp

KEY WORDS AND PHRASES

Engineering test satellite VII (ETS-VII), Flexible parts handling, Predictive display, Teleoperation, Time delay problem.

ABSTRACT

This paper reports experiments involving the handling of flexible parts (e.g. wires) when using a teleoperated system with time delay. The task is principally a peg-in-hole task involving the wrapping of a wire around two posts on the task-board. It is difficult to estimate the effects of the flexible parts, therefore, on-line teleoperation is indispensable for this class of unpredictable task.

We first propose a teleoperation system based on the predictive image display, then describe an experimental teleoperation testbed with a four-second transmission time delay. Finally, we report on wire handling operations that were performed to evaluate the performance of this system. Those experiments will contribute to future advanced experiments for the MITI ETS-VII mission.

INTRODUCTION

Remote manipulation in outer space from the ground is one of the most important technologies for assisting outer space activities such as the construction and maintenance of space stations, and the operation of space laboratories. The long distances between the ground command station and outer space robots incur an inevitable time delay of communications between these two systems; there are many research activities being conducted on this time delay problem.

Several ideas have been proposed: local intelligence with sensory feedback [1], a predictive image display system which superimposes a phantom robot with no delay on the remote camera image [2], a teleoperation system using force-reflecting simulator [3,4] and a tele-programming system which issues program segments to the remote site [5]. Space robot experiments have also been carried out on a space-lab mission [6].

In this paper we consider the tasks involved in handling flexible parts by a teleoperated system with communication delays, and focus on a wire handling task as an example of a general unpredictable task for teleoperation.

This task is complicated for two reasons: 1) the dependence of the generated path on the changes of the shape of the flexible component, and 2) the difficulty of estimating the forces generated by the deformation of the flexible part. It is difficult to estimate the effects of the wire, and pre-programmed methods are not suitable for this class of task; an on-line teleoperation system is indispensable.

We first propose a teleoperation system based on a predictive computer graphics display, then describe an experimental teleoperation testbed with a four-second communication time delay. Finally, we report on wire handling operations that were performed to evaluate the effectiveness of the system.

TELE-OPERATION SYSTEM BASED ON PREDICTIVE CG DISPLAY

In this section, we propose a remote manipulation system based on the predictive image display technique. Figure 1 shows the block structure of this system. The system consists of the master operating station subsystem and the

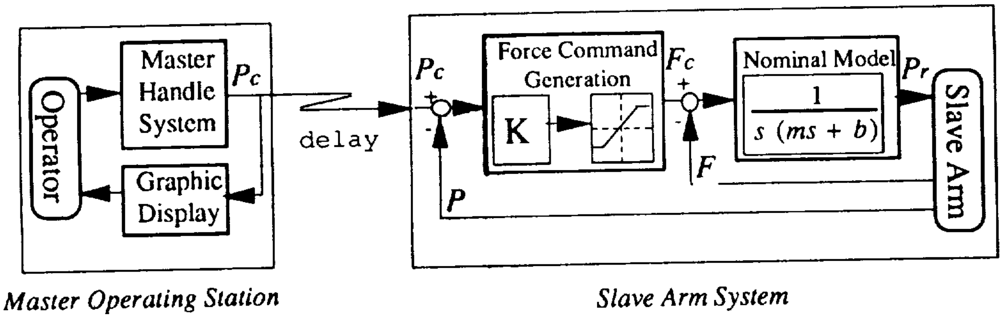


Figure 1. The proposed master-slave tele-operation system

slave arm subsystem, and these two are connected by a low-bandwidth communication line with a large time delay.

If we assume a several-second-delay, we are not able to use the conventional master-slave system which consists of a large control loop, so we adopt the predictive graphic image display technique.

The master subsystem is composed of a master handle, its controller, and a graphic simulator. This master subsystem simulates a virtual arm and displays it as a three-dimensional image. In this simulator, we ignore interactions with the environment and the hardware limitation of the slave system, hence the operator can control the virtual arm on the graphic display freely through the master handle. The series of configurations (position / orientation) of the virtual arm during operation are transmitted to the slave arm controller as a command configuration P_C.

The slave arm subsystem is composed of a slave arm and its double-loop controller which prevents the slave arm from excess loads. In the outer position loop, the operational force command F_C for the inner loop is given by

$$F_c = \begin{cases} -F_{\text{limit}} & (\text{in case } K(P_c - P) < -F_{\text{limit}}) \\ F_{\text{limit}} & (\text{in case } K(P_c - P) > F_{\text{limit}}) \\ K(P_c - P) & (\text{others}) \end{cases} \quad (1)$$

where F_{limit} is the limit force/torque to prevent the slave arm from applying excess loads, K is the gain parameter of the outer position loop, and P_C and P are the commanded and the sensed configurations of the slave arm, respectively.

We assume a position-controlled slave arm. In the inner force control loop, the reference position P_r for the slave arm is calculated by using a nominal model of the slave arm. Its transfer function is as follows,

$$P_r(s) = \frac{1}{s(ms+b)}(F_c(s) - F(s)) \quad (2)$$

where F is the sensed force at the tip of the slave arm, and m and b are the inertia and the damping parameters respectively of the nominal model. These parameters are designed to keep the bandwidth of the output reference position P_r within the bandwidth of the slave arm.

The operator manipulates the virtual arm on the graphic display using the master handle, sometimes watching the monitor of the slave arm system to check the motion of the slave arm for any failure of the wire-wrapping task. In the event of any such failure, the operator returns the virtual arm to its previous state and retries the wire wrapping.

EXPERIMENTAL TELEOPERATION TESTBED

To confirm the function of the proposed system, we constructed a teleoperation testbed with a four-second time delay. Figure 2 shows an overview of this experimental setup. The sequence of the command positions from the master subsystem is stored once in a ring buffer program which simulates a four-second communication time delay. The response of the slave arm is thus delayed by four seconds.

An IRIS workstation (Crimson / Reality-Engine) is used for the three dimensional computer graphic display, and a newly designed hybrid master system (Fig. 3) is used for the master handle. This handle has three degrees of orientational freedom, and the orientation of the virtual arm follows the orientation of this master

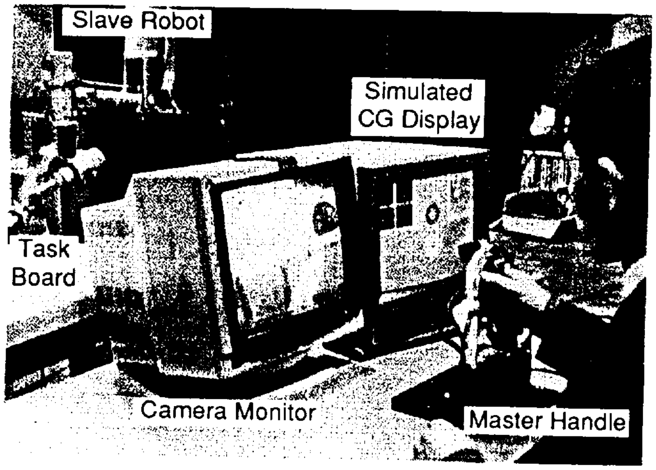


Figure 2. Overview of the experimental setup

handle. A six-axis force/torque sensor is installed at the base part of this handle, and the translational velocity of the virtual arm is proportional to the force which is applied by the operator.

Two monitor displays are used for the master operating station, one to display the computer graphics which simulate the virtual arm, and the other to display the delayed camera image of the slave arm system. On the graphic display, the front view of the task board with two holes and two poles, and the manipulated peg, are displayed as 3D solid models.

To check the real motion of the slave arm, the information of the real peg is also displayed super-imposed on this window as a 3D wire-frame model. On the top-right corner of the graphic display, a small window displays a side view of the slave arm system. On the top-left corner of the graphic display, another small window displays the force information of the slave arm.

A direct-drive arm with six degrees of freedom is used for the slave arm. At the tip of this arm, a six-axis force / torque sensor is installed to detect the forces generated by the interaction with the environment. A slave arm control algorithm described before is implemented on a parallel processing system of transputers. The force limit F_{limit} was set to 5N to protect the slave arm from excess loads during operation.

WIRE HANDLING TASK

As an example task of manipulating flexible parts, we tested a wire handling operation. We used a simple task-board with two holes, two poles and one manipulated peg with a thin copper cable. The clearance between the peg and the hole is 0.035mm.

The task is principally a peg-in-hole task involving wrapping a wire around two posts on the task-board. The task consists of three stages; first, extracting the peg from the hole, second, wrapping the wire around the two poles, and third, inserting the peg into the initial hole.

The results of the experiments are shown in Fig. 4 as a sequence of wire handling operations. Despite of the large communication time delay in this teleoperation system, we confirmed the success of the wire handling operation.

To maintain consistency between the virtual arm space and the real slave arm space, we calibrated the system prior to the experiments.

CONCLUSION

In this paper, we investigated a wire handling task as an example of an unpredictable task. We proposed a teleoperation system with the predictive image display and the double-loop slave controller, constructed a master-slave teleoperation testbed, and performed the wire handling task with a four-second communication delay.

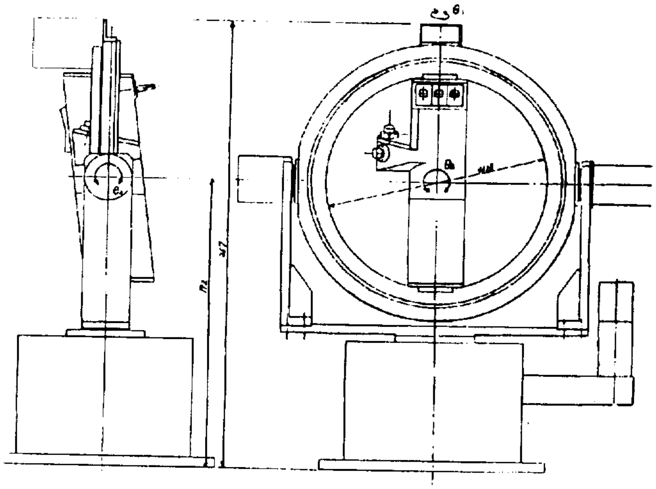


Figure 3. Structure of the hybrid master handle

MITI is planning to participate in the space robotic experiment on the ETS-VII [7], and an advanced robotic hand (ARH) with multiple degrees of freedom and sensors has been developed for this mission [8]. This experiment will contribute to future advanced experiments for the MITI ETS-VII mission.

This research was performed as a joint research project between the Mechanical Engineering Laboratory and the Research and Development Center of Toshiba Corporation.

Acknowledgments

The authors would like to thank Dr. K. Machida, ETL, for his valuable suggestions and discussions on this joint research project.

REFERENCES

[1] Sheridan, T., 1992. *Telerobotics, automation, and human supervisory control*, MIT Press.

[2] Bejczy, A. K.; and Kim,W. S., 1990. Predictive displays and shared compliance control for time-delayed telemanipulation. In *Proc. of the IEEE Int. Workshop on Intelligent Robots and Systems (IROS'90)*, 407-412.

[3] Kotoku, T.; Tanie, K.; and Fujikawa, A., 1990. Force-reflecting bilateral master-slave teleoperation system in virtual environment. In *Proc. of the Inter. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'90)*, 295-298.

[4] Kotoku, T., 1992. A predictive display with force feedback and its application to remote manipulation system with transmission time delay. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'92)*, 239-246.

[5] Funda, J.; Lindsay, T. S.; and Paul, R. P., 1992. Teleprogramming: towards delay-invariant remote manipulation. *Presence* 1(1): 29-44.

[6] Brunner, B.; Hirzinger, G.; Landzettel K.; and Heindl, J., 1993. Multisensory shared autonomy and telesensor-programming -- key issues in the space robot technology experiment ROTEX. In *Proc. of the IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS'93)*, 2213-2139.

[7]Oda, M.; Wakabayashi, Y.; Ichikawa, S.; Imai R.; and Anzai, T., 1992. ETS-VII, the world first telerobotic satellite (mission and its design concept), In *Proc. of the Inter. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'92)*, 307-318.

[8] Machida K.; and Ono, S., 1992. Space experiment plan of advanced robotic hand for ETS-VII, In *Proc. of the Inter. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'92)*, 219-223.

Figure 4. Task sequence of a wire handling

Interactive Autonomy and Robotic Skills

N95- 23730

A. Kellner, B. Mädiger
Deutsche Aerospace AG
Raumfahrt-Infrastruktur
Huenefeldstr. 1-5
28199 Bremen, Germany
Tel 421 539 4987 Fax 421 539 5726
e-mail: bernd.maediger@erno.de

KEY WORDS AND PHRASES

Autonomy, Neural Nets, Robotic Skills, Space Robotics

INTRODUCTION

Current concepts of robot-supported operations for Space Laboratories (payload servicing, inspection, repair and ORU exchange) are mainly based on the concept of "interactive autonomy" which implies autonomous behaviour of the robot according to predefined timelines, predefined sequences of elementary robot operations and within predefined world models supplying geometrical and other information for parameter instantiation on the one hand, and the ability to override and change the predefined course of activities by human intervention on the other hand.

Although in principle a very powerful and useful concept, in practice the confinement of the robot to the abstract world models and predefined activities appears to reduce the robot's stability within real-world uncertainties and its applicability to non-predefined parts of the world, calling for frequent corrective interaction by the operator, which in itself may be tedious and time-consuming.

In this paper methods are presented to improve this situation by incorporating "robotic skills" into the concept of interactive autonomy.

CONTROL FUNCTIONS AND INFORMATION BASES FOR INTERACTIVE AUTONOMY

The control and information architecture associated with the concept of interactive autonomy can be conceived as a three-layered structure, where the top-layer (the system layer) reads in the timeline of robot, payload and subsystem tasks driving the whole system, checks the tasks for consistency and delegates them to the different recipients (robot, payloads, subsystems), the middle layer (subsystem layer) breaks down the tasks into robot- and payload-specific action sequences, instantiates their parameters and delegates them to the bottom layer (equipment layer) where the final control execution is performed.

Associated with each control layer is a database of <u>predefined</u> operational knowledge (timelines, action sequences, control strategies, as well as failure handling methods) and a database containing <u>predefined</u> environment representations (e.g. geometrical world-model for the robot) updated according to <u>predefined</u> transitions after action execution.

To support interaction with the real world, <u>predefined</u> expected sensor values (e.g. forces and torques) may be supplied with the predefined actions.

Moreover, associated with each control layer there is an MMI which allows operator interaction on the respective layer at any time during the autonomous execution of the timelines, thus providing for <u>interactive</u> autonomy.

NEED FOR OPERATIONAL ENHANCEMENTS

First analyses and practical experience with prototypes realizing the a.m. control and information architecture show both the power of this concept of interactive autonomy and its shortcomings.

The power of the concept is particularly apparent on system level in the case of payload servicing operations. By a suitable MMI, the coordinated, interactive robot-payload operations can easily be monitored, and whenever a change in robot-payload interaction is necessary, this can easily be achieved by changing the task sequences accordingly.

However, on subsystem-level problems can occur when there is a mismatch between predefined world-model and real-world data, e.g. due to erroneous input or update, deformation in the environment, or miscalibration of the robot, or when objects need to be handled which have not been foreseen in the world-model or which are not amenable to modelling, e.g. hoses and cables.

Operator intervention on subsystem-level in this case implies selection of robot action sequences and action parameter tuning, which can be extremely tedious and time-consuming.

Of course, operator intervention on equipment level, i.e. by telemanipulation (joystick control) seems more appro-priate in these cases.

However, if the control is performed from the ground, the command-feedback round-trip time of several seconds again leads to tedious and time-consuming operations, not to speak of the problems inherent per se in fine-manipulation using video feedback.

The same applies to problems which may occur on equipment-level during control execution, such as jamming in insert/extract operations.

Obviously, some type of sensor-based control algorithms would be required to eliminate these problems.

However, in general these cannot only be of the type providing closed-loop sense-act cycles (e.g. for force/torque-based compliant motion) but need to provide strategies based on general knowledge, e.g. how to grasp objects which are not amenable to modelling in a world-model, such as hoses or cables. This leads to the concept of "robotic skills" as an additional, essential ingredient of the concept of interactive autonomy.

ROBOTIC SKILLS

As examples, in the following two skills are presented: the "grasping skill" and the "insert/extract-skill".

In the first case, the robot is provided with the ability to grasp an a priori unknown object indicated by placing the cursor on its 3D-video image generated by a pair of gripper cameras - certainly an enhancement of the a.m. concept of interactive autonomy, which would otherwise require action sequence selection and parametrization "by hand", or telemanipulation as explained above.

In the second case, the skill provides for a general jamming-free insertion/extraction capability.

Grasping Skill

This skill comprises an image preprocessing function which segments out the object indicated by the cursor, and a "sensomotory mapping" which incorporates generic knowledge for mapping object images onto robot commands such that the gripper can grasp the objects. In the following, only these sensomotory mappings are discussed further:

Since they represent generalized "grasping knowledge" which is not easily amenable to explicit (algorithmic) coding, the approach taken was to encode them in Neural Nets trained on a set of samples and to investigate the generalization capability of these mappings.

In the first, straightforward analysis a 3-layered backpropagation net was trained on a large number of objects, each in various orientations, together with the corresponding correct grasping poses of the robot, thus providing mappings from object shape and orientation to robot commands. Essentially these commands are joint angle increments which improve the gripper pose relative to the "graspable" area of the object. After each increment execution, the sensomotory mapping is performed again, thus providing a "servoing" on the object's shape. However, training times appear to be quite prohibitive and, in particular, the generalization capabilities to non-trained shapes is not satisfactory.

In a second approach the image of the indicated object is scanned for grasping areas by means of a filter realized by a 3-layered backpropagation net which has learned the human (!) assessment of a large number of object-partitions which can be grasped and partitions which cannot be grasped by the robot. This method produces excellent results in acceptable computation times.

Surprisingly, a third method also proved very promising: in this case both architecture and synaptic weights of a Neural Net were designed "by hand" such that as soon as an area fitting between the gripper fingers is detected by the first layer of neurons as the robot slowly rotates (by default) the gripper cameras over the object, the shape of the area generates robot commands such that the area's line of gravity is aligned with the symmetry line between the gripper fingers. Grasping is performed when the width of the aligned are is identified by the net as large enough for the robot's gripper. However, this method only applies for objects with not too complex structures of the grasp surfaces.

Of these three approaches, the first was analyzed by simulation only. In the latter two cases both simulation and subsequent testing on a 6 DOF commercial robot with gripper cameras were performed.

Insert/extract-Skill

In this case the "sensomotory mapping" is given by the mapping of force/torque-histories typical for imminent jamming (measured by suitable sensors in the robot's wrist) onto appropriate corrective robot commands to avoid the jamming situation in insert or extract operations.

Input signals are the 6 components of the force/torque signals and the current position of the robot. In order to incorporate the temporal evolution of the input signals, backpropagation nets with tapped delays are used. The difficulty lies in the training procedure: the only possibility is to record a large number of examples of a human operator performing jamming-free inserts/extracts or remedies in case jamming is imminent, and to train the net on this human behaviour.

First tests already showed promising results. However, further investigation is necessary to provide a truly general insert/extract-skill module.

CONCLUSIONS

The current concept of interactive autonomy for robot operations in Space Laboratories can be enhanced by robotic skills. Since these imply complex sensomotory mappings not easily amenable to explicit coding, training these mappings by Neural Nets seems to be an appropriate approach.

First tests with such Neural-Net-based skills for grasping and insert/extract operations provided promising results and appear to undergird the feasibility of the method of neural control.

N95- 23731

Visually Guided Grasping to study teleprogrammation within the BAROCO testbed

M. Devy * V. Garric * M.Delpech ** C. Proy **

* L.A.A.S. - C.N.R.S.
7 Avenue du Colonel Roche
31077 Toulouse Cedex
FRANCE

Email: michel, garric@laas.fr

** CNES
18, avenue Edouard Belin
31055 Toulouse Cedex
FRANCE

Email: delpech, proy@philae.cnes.fr

KEY WORDS

Model-Based Vision, Localization , Grasping

INTRODUCTION

This paper describes vision functionalities required in future orbital laboratories; in such systems, robots will be needed in order to execute the on-board scientific experiments or servicing and maintenance tasks under the remote control of ground operators. For this sake, ESA has proposed a robotic configuration called EMATS; a testbed has been developped by ESTEC in order to evaluate the potentialities of EMATS-like robot to execute scientific tasks in automatic mode.
For the same context, CNES develops the BAROCO testbed [1] to investigate remote control and teleprogrammation, in which high level primitives like "Pick Object A" are provided as basic primitives.
In nominal situations, the system has an a priori knowledge about the position of all objects. These positions are not very accurate, but this knowledge is sufficient in order to predict the position of the object which must be grasped, with respect to the manipulator frame. Vision is required in order to insure a correct grasping and to guarantee a good accuracy for the following operations.
In this paper, we describe our results about a visually guided grasping of static objects. It seems to be a very classical problem, and a lot of results are available [3]. But, in many cases, it lacks a realistic evaluation of the accuracy, because such an evaluation requires tedious experiments. We propose in this paper several results about calibration of the experimental testbed, recognition algorithms required to locate a 3D polyhedral object, and the grasping itself.

SYSTEM CALIBRATION

The figure 1 shows the LAAS experimental testbed: a 6 d.o.f. classical manipulator, with a camera mounted near the gripper. Before any experiment, a lot of knowledge must be learnt: we do not focus on these steps, but, the final results, and especially, the accuracy of the grasping, depends heavily on the calibration quality. In this work, we only use a classical "Look and

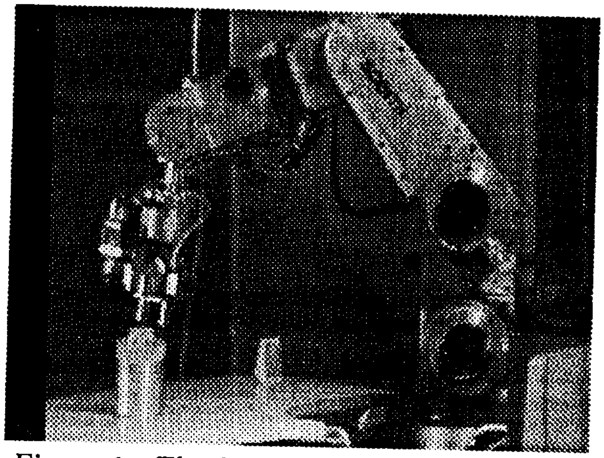


Figure 1: *The LAAS experimental testbed*

Move" strategy in order to guide the manipulator towards the object.

On figure 2, the five different frames used during the Pick and Place task, are represented: the more important is R_{rob}, static frame linked to the robot, in which the position of the effector frame R_{eff} is known by the transform T_{re}. Two transforms must be estimated off line: T_{eg} and T_{ec}. The transform T_{co} must be estimated by the object localization from the image, corrected from distortions. In nominal situation, we have a rough estimate for the transform T_{ro}, from the a priori knowledge of the environment model.

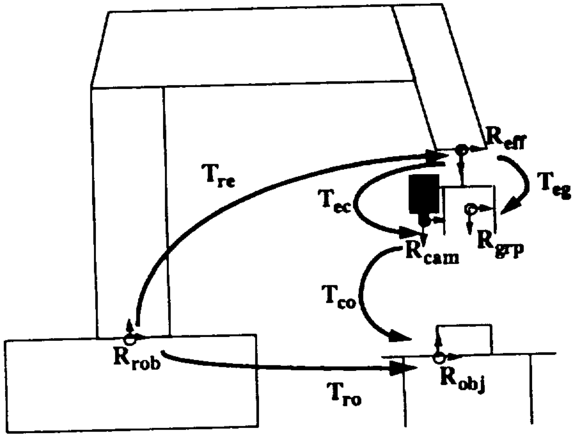


Figure 2: *Reference frames*

These gripper and hand-eye calibrations have been performed by the Tsai method [5], using a specific object (a dihedral part, fitted with visual patterns). We have evaluated the stability and the accuracy of the hand-eye calibration, for several positions of the camera around the object; we compare the estimations of the object position with respect to the robot frame R_{rob}; this position is computed by the transform product: $T_{re} * T_{ec} * T_{co}$.

Then, the stability of this product means good estimations for T_{re} measured by internal sensors, T_{ec} estimated by the hand-eye calibration and T_{co}. We can use localization functions, which take as inputs, point matchings [4] : mean deviations of less than 1 mm for the translation, 0.06 degrees for the orientation.

Once the manipulator is calibrated, we must initialize an approximative environment model, such that the initial positions of the work areas and of the objects around the robot, are known with a maximum deviation of 5 cm in translation, and 15 degrees in orientation. At last, the object models are described by a R.E.V. graph. For each direction around the object, we index the visible 2D primitives, and we point to the discriminant clues which could provide good hypothesis, without time consuming: especially discriminant perceptual groupings, like a polygonal chain or a set of parallel segments.

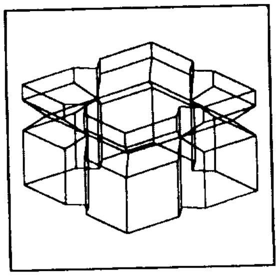


Figure 3: *Grasp interface*

The figure 3 presents the wireframe model of the grasp interface (3*3*2 cm cubic part) which will fit all equipments that the manipulator will have to pick.

OBJECT RECOGNITION

A general model-based method performs identification and localization of a 3D polyhedral object only from one image. The recognition algorithm is based on the R.E.V. models and the aspect graphs of the objects; it relies first on a generation of hypotheses, then on a verification of each pertinent hypothesis. Experiments have shown that this method required very good results for the segmentation , and that complexity could be very important (cluttered environments, occlusions, noisy images, ...). Nevertheless, 3D object recognition from a single image can provide fair results if it exists on the object model, some discriminant clues, from which a rigth hypothesis can be generated without any complexity.

Generally, for the generation, hypotheses are searched in a compatibility graph, in which each node corresponds to a so-called elementary hypothesis i.e., a matching between a scene feature and a model feature

(segments, regions, elliptic contours) , and each arc stands for the compatibility between two matchings; for each consistent hypothesis, the object position is computed. For the verification and refinement, we look for new matchings between scene features and predicted positions of model features. The generation of the elementary hypotheses relies on length criterion for single segments, or from different parameters for perceptual groupings (parallel or convergent segments). In order to determine if two elementary hypotheses are compatible, we use two kinds of constraint: *topological constraints* (connexity using the REV graph, and visibility, using the aspect graph), and *numerical constraints* (invariant measures according to affine tranformation).
Once the compatibility graph is built, the search for recognition hypotheses is performed by the maximal cliques algorithm. This method can be very expensive in computing time, due to their significant combinatorial complexity, especially if the compatibility graph is very large (too many elementary matchings, too weak compatibility criteria).
For each pertinent hypothesis, a first localization based on the segment matchings, is computed by [2]. Then, we can predict the object position in the image and infer (scene segments, model edges) matchings. If such matchings are not found, the confidence rate on this hypothesis must be reduced; otherwise, it can be increased, and a more accurate localization can be computed using Kalman filtering algorithm.

VISUALLY GUIDED GRASPING

Effectively, in the nominal case, when the system must execute a high level primitive "Pick object CYLINDER", the approximative position of CYLINDER can be found in the environment model. If this position was perfectly known, and with a perfect robot, we could directly command a movement towards the final grasp position from which the gripper could be closed.
In order to reach the actual grasp position, a vision procedure is required to correct the T_{ro} estimate during the approach, and to dynamically correct the error due to the geometrical model of the manipulator. The last movement towards the grasp position will be undertaken, only when the T_{ro} estimate will be refined and when the length of this last movement will be weak enough to insure that the grasp position will be reached with an error lower than the required tolerance (at this time, half a millimiter).

Figure 4: *First image*

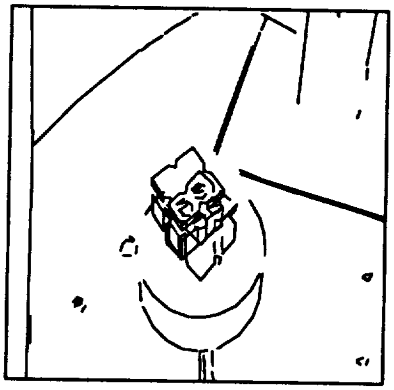


Figure 5: *Scene features*

So, through the first estimate of the object, T_{ro_0}, through the aspect graph which says what is the better view point to deal with the recognition of the grasp interface on CYLINDER, a planification module can off line select an optimal effector position T_{re_1}, from where an image is acquired and segmented (figures 4 and 5). From this image 1, the recognition of the cubic grasp interface, could be very simple, since the environment model gives directly the hypothesis on the object position according to the robot frame; using the different

transforms shown on figure 6 (the dashed box represents the estimated object position, according to the a priori knowledge), we can directly predict the object position T_{pred_0} with respect to the camera:
$T_{pred_0} = T_{ec}^{-1} * T_{re_1}^{-1} * T_{ro_0}$.
This prediction can replace the one given by the hypothesis generation procedure of a recognition system; it could be validated in the verification step. We show on figure 5 a possible predicted position of the object model.

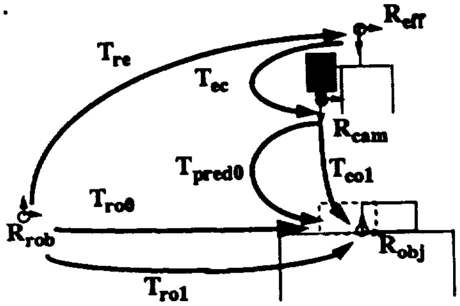


Figure 6: *Model prediction*

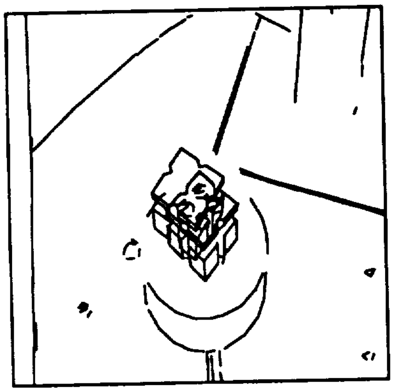


Figure 7: *First localization*

The final localization T_{co_1} is presented on figure 7. From this localization with respect to the camera frame, we can compute a better estimate T_{ro_1} of the object position with respect to the robot frame:
$T_{ro_1} = T_{ro_0} * T_{pred_0}^{-1} * T_{co_1}$
For the last iteration, the figure 8 shows the projection of the visible model edges for the prediction and for the final localization; the final localization seems perfect (model edges confounded with the scene segments). We have at this time some difficulties to estimate the error on the final grasp operation. The only result is visual; it seems we have about 1 mm error, when the effector reaches the grasp position.

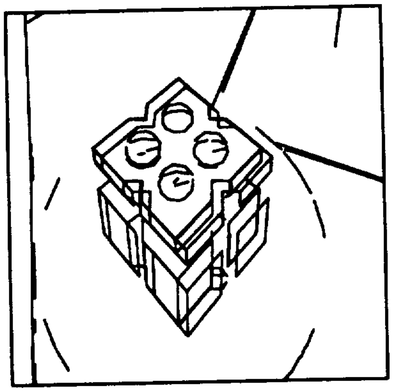


Figure 8: *Last iteration*

CONCLUSION

We have described in this paper, a perception application related to visually guided Pick and Place task which will be required in teleprogrammation mode to undertake scientific experiments in future in-orbit laboratories. Other research works will be done in order to improve the perceptual algorithms, especially to take in account more complex objects.

References

[1] M. Delpech, J.Y. Prado, P. Giron, and J. Tailhades. Baroco, a testbed to study teleoperation within the columbus environment. In *Third ISAIRAS, Toulouse (France)*, September 1992.

[2] M. Dhome, M. Richetin, J.T. Lapresté, and G. Rives. Determination of the Attitude of 3D Objects from a Single Perspective View. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, December 1989.

[3] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-based space robotics-rotex and its telerobotic features. *IEEE Transactions on Robotics and Automation*, October 1993.

[4] P. Puget and T. Skordas. An Optimal Solution for Mobile Camera Calibration. In *1st ECCV (Nice)*, April 1990.

[5] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, June 1989.

Planetary Stations and Abyssal Benthic Laboratories: an Overview of Parallel Approaches for Long-Term Investigation in Extreme Environments

N95- 23732

S. Di Pippo (*), W. Prendin (**), F. Gasparoni (*
(*) Italian Space Agency; (**) Tecnomare Spa,
San Marco 3584,
30124 Venezia, Italy,
Tel.+39-41-796711, Fax +39-41-796800

KEY WORDS AND PHRASES

Robotics, Space, Planetary Stations, Benthic Stations, Deep Water.

ABSTRACT

In spite of the apparent great differences between deep ocean and space environment, significant similarities can be recognized when considering the possible solutions and technologies enabling the development of remote automatic stations supporting the execution of scientific activities.

In this sense it is believed that mutual benefits shall derive from the exchange of experiences and results between people and organizations involved in research and engineering activities for hostile environments, such as space, deep sea and polar areas.

A significant example of possible technology transfer and common systemistic approach is given in this paper, which describes in some details how the solutions and the enabling technologies identified for an Abyssal Benthic Laboratory can be applied for the case of a lunar or planetary station.

INTRODUCTION

As recently highlighted by the European Space Agency (ESA) Lunar Study Steering Group, the utilization of the Moon offers a wide range of possibilities, for a better understanding of the Moon itself, of the Earth-Moon system, of the history of the solar system, as well as an improved potential return for astronomy and later on for life science activities and research into artificial ecosystems. On this regard three possible categories of scientific activities for future lunar missions can be envisaged:

- Science of the Moon, covering determination of physical, chemical, and geological characteristics of lunar surface and internal structure;
- Science on the Moon, dealing with questions relating human activities in space and development of artificial ecosystems;
- Science from the Moon, including specific areas in astronomy that can be better studied from the Moon than from satellites or Earth.

These activities call for the availability of dedicated stations, capable of operating autonomously for long periods and carrying out a wide number of scientific tasks. A similar approach is being studied for the study of underwater abyssal environment.

THE CASE OF THE ABYSSAL BENTHIC LABORATORY

As for the lunar environment, knowledge of deep sea bottom and related processes (physical, chemical, biological and geological) is still quite limited, but at the same time the demand for the execution of research activities at depths below 4000 to 6000 meters is growing and wider ranges of scientific needs are being identified. What is lacking nowadays is the possibility to go deeper in the ocean and conduct long term and large scale multidisciplinary activities, not limited to sensing and observation, but extended to sampling and, above all, ensuring real "experimenting" capabilities.

On behalf of European Union (Directorate General XII), Tecnomare assessed the feasibility of a configuration for a benthic underwater system, called ABEL (Abyssal BEnthic Laboratory), capable of operating both under controlled and autonomous modes for a period of several months to over one year at abyssal depths up to 6000 m.

A network of co-operating stations, open to different configuration arrangement, has been identified in order to satisfy the widest range of scientific expectations, and at the same time to address the technological challenge to increase the feasibility of scientific investigations, even when request is not yet well clarified. The overall system (shown in Figure 1) consists of three main elements:

- a **Main Fixed Station** devoted to the execution of the most complex scientific activities, characterized by a high level of interaction between internal functions, like sampling, observation and sensing, and performance of experiments asking for actuations and manipulation as well as tele-operated activities.
- one or more **Satellite Stations**, acting as nodes of a measuring network (e.g. for seismic, geodynamic, hydrographycal measurements), or as remote stations, placed in proximity of a site or phenomenon worth a continuous monitoring activity.
- a **Mobile Station** extending ABEL capabilities with the possibility to carry out surveys over the investigation area and interventions on the fixed stations such as visual inspection, instruments positioning and maintenance, data/sample transfer, reprogramming of activities.

Communication between stations is based on hydroacoustic links (shown as dashed arrows in Figure 1).

ABEL architecture also includes a dedicated Deployment and Recovery Module, as well as sea-surface and land-based facilities. Such an installation constitutes the sea-floor equivalent of a meteorological or geophysical laboratory.

Three different operating modes have been envisaged, each referring to a different level of interaction among ABEL system components and surface facilities:

- **autonomous mode**, characterised by the absence of any interaction with surface facilities after system installation. Mission autonomy is not completely determined a priori; the capability of modifying mission profiles according to observations and events has to be included.
- **interactive mode**, in which the ABEL system interacts with surface facilities, such as a vessel or a moored buoy, by means of a low capacity, time delayed link, based on hydroacoustic transmission; in this way a limited capability of data transfer and further instruction transmission is ensured.
- **controlled mode**, characterised by a direct and real-time remote interfacing of scientific personnel with the ABEL system. A high capacity, fiber optic link, communication is provided by the Deployment and Recovery Module. This mode make it possible to perform the most complex tasks requiring direct operator control and data/image transmission.

APPLICATION OF KEY ROBOTIC TECHNOLOGIES TO UNDERSEA AND SPACE EXPLORATION

Among the various analogies existing between a deep ocean benthic laboratory and a planetary base, the need to tele-operate a scientific laboratory from a remote control station is the aspect involving the use of very similar robotic technologies such as supervisory control, tele-operation, man-machine interface (MMI) and telepresence, computer vision, etc.

This paragraph deals with a short description of these key technologies, the approaches and the results achieved in the marine sector and highlights possible technology transfers to space.

The basic control approach which has been adopted in the more advanced telemanipulation systems installed on free swimming Remotely Operated Vehicles (ROV's) to carry out underwater complex tasks in substitution of divers, is *Supervisory Control.*

As known, Supervisory control (Figure 2) represents a methodology aimed at properly combining human and computer actions for the efficient control of complex systems. According to this methodology, the different interactions between operator and computer are suitably combined in such a way to substantially facilitate the human Operator in carrying out the system control: in this sense the system is conceived to assist, not to substitute for the operator. More specifically, in the supervisory control scheme, the Operator is requested to carry out high level tasks such as planning, system instruction, monitoring during system operation and intervention when necessary (e.g. on account of unexpected situations or for varying pre-defined task parameters). The supervisory computer(s), instead, takes care of the interpretation and decoding of the Operator high level commands in elementary tasks and of their execution by using the sensors and actuators of the controlled system. The supervisory control paradigm is particularly suited for the control of advanced telerobotic systems. In particular it easily allows to increase more and more the system autonomy in dependence, for example, of the development of Artificial Intelligence (AI) technology and of the experience gained in actually carrying out tasks.

Key technologies and capabilities constituting the prerequisites of the supervisory control approach are:

1) Motion-force primitives, i.e. elementary tasks that the system is able to carry out both in fully automatic way or in tele-operation. An example could be the motion from point A to point B while avoiding obstacles. In a bottom-up approach for automation they represent the elementary building bricks of a large spread of tasks. The approach of motion-force primitives may be considered of general application; for this reason the developments carried out for underwater environment can be easily modified and finalised for space robots such as rovers.

2) Advanced MMI and telepresence. These technologies are fundamental in remote-controlled operations. In fact even if supervisory control greatly simplifies the Operator's tasks, he remains a fundamental element in the control chain. The human factors include any telepresence techniques, associated with methods for computer representation of the working scenario. As known, the ultimate target of telepresence is to make the operator feeling to be within the working scene as he was looking and manipulating with his own senses (eyes, hands etc.). This may be approached in different ways: one is to proceed by testing step by step new techniques. For example, considering the extremely poor scene perception obtained from underwater TV cameras, an idea is to complement TV images with 3D graphic representation of the working scenario (made possible after scene reconstruction). This solution enhances considerably the effectiveness of operator interface especially when TV images are mixed with graphics in such a way to artificially increase the TV cameras field of view. Particular synergies exist between space and underwater areas, to make the advanced MMI technologies developed for one sector almost directly applicable to the other.

3) Computer vision systems. This technology deals with vision methods for measuring the geometry of the working environment. Computer vision is one of the key elements in measuring the size and shape of the working environment with a view to computer workspace modelling. To this purpose Tecnomare developed the TV-Trackmeter (Fig. 3) a stereo computer vision system capable to measure points of the scene taken by stereo cameras, while tracking them in case of relative motion between the vehicle and the scene. Typical measurement accuracy is 4 mm at 2 m range; repetition rate is around 12 Hz. To reconstruct the scene geometry, a very high number of points are measured; then the measurements are "fitted" to the geometrical shape of the scene, assumed known, by using optimal algorithms. In this way key geometrical parameters (e.g. radius and axis for a cylindrical shape) are estimated with considerable accuracy. Computer stereo vision is a typical robotic technology having a large spread of different applications and it is almost directly applicable in different sectors such as underwater and space.

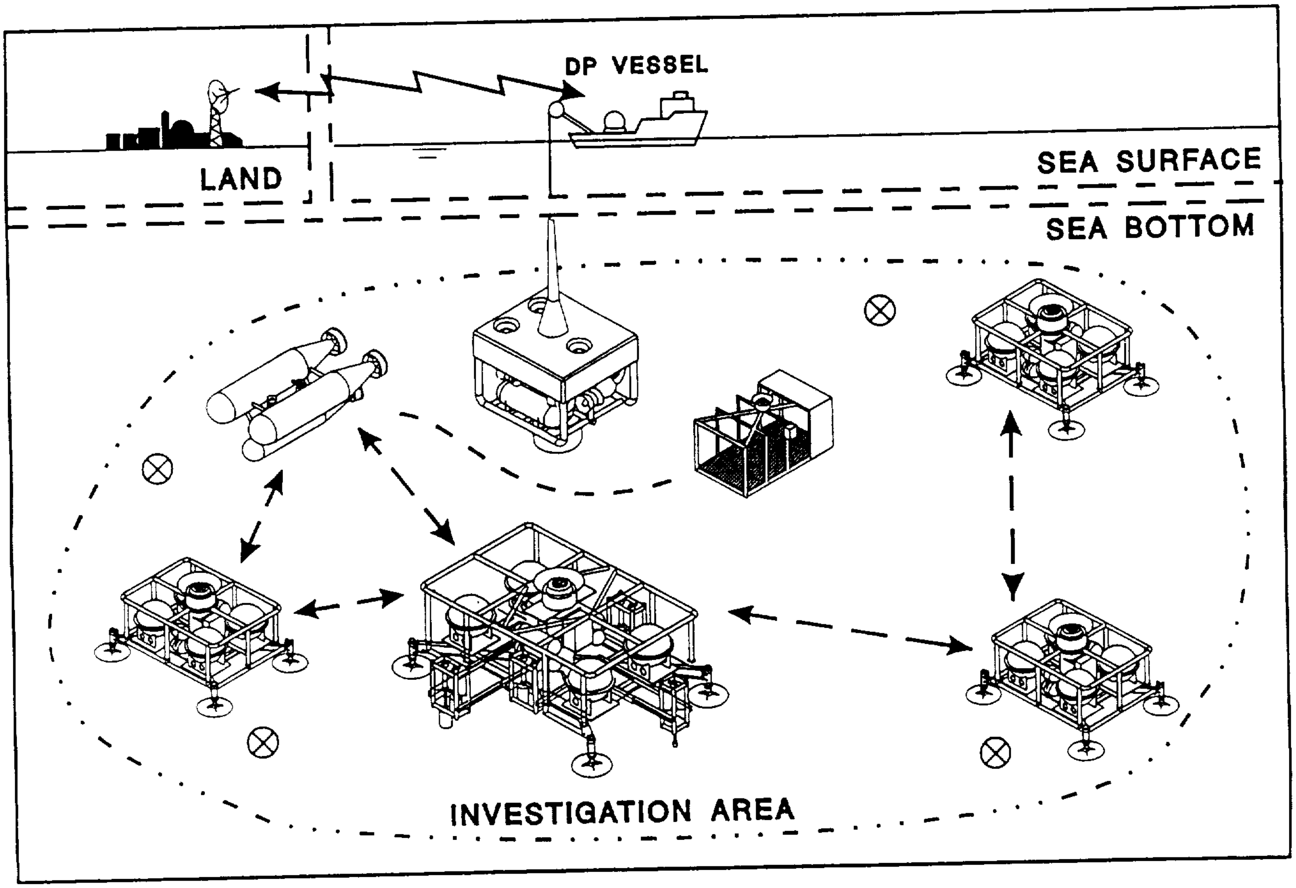


Figure 1. Abyssal Benthic Laboratory

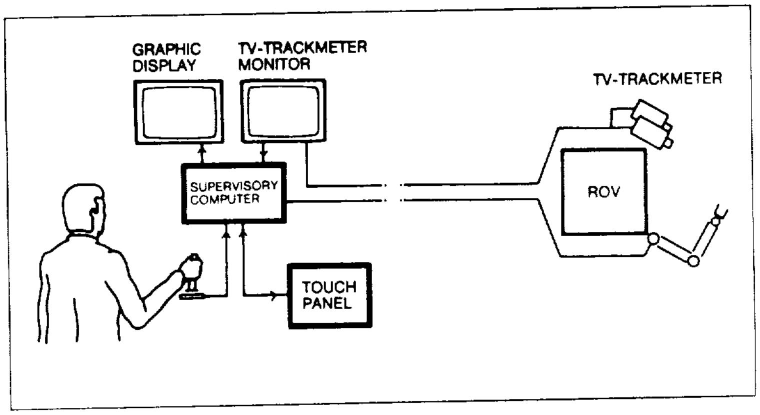


Figure 2. Supervisory Controlled Telemanipulation

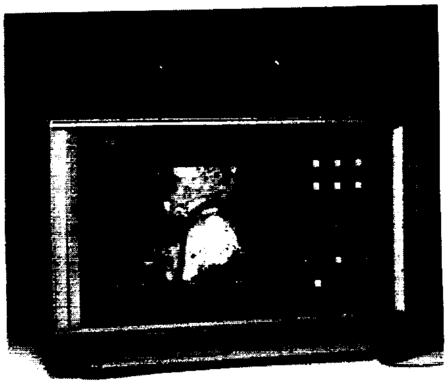


Figure 3. TV-Trackmeter

Design of Teleoperation System with a Force-Reflecting Real-time Simulator

Mitsunori Hirata, Yuichi Sato, Fumio Nagashima, and Tsugito Maruyama
Fujitsu Laboratories Limited
1015 Kamikodanaka, Nakahara-ku, Kawasaki
211, Japan
Tel: +81-44-754-2659 Fax: +81-44-754-2582
E-Mail: hirata@lucy.stars.flab.fujitsu.co.jp

N95- 23733

KEY WORDS AND PHRASES

Force-reflection, peg-in-hole, predictive display, teleoperation, virtual collision.

ABSTRACT

We developed a force-reflecting teleoperation system that uses a real-time graphic simulator. This system eliminates the effects of communication time delays in remote robot manipulation. The simulator provides the operator with predictive display and feedback of computed contact forces through a six-degree of freedom (6-DOF) master arm on a real-time basis. With this system, peg-in-hole tasks involving round-trip communication time delays of up to a few seconds were performed at three support levels: a real image alone, a predictive display with a real image, and a real-time graphic simulator with computed-contact-force reflection and a predictive display. The experimental results indicate the best teleoperation efficiency was achieved by using the force-reflecting simulator with two images. The shortest work time, lowest sensor maximum, and a 100% success rate were obtained. These results demonstrate the effectiveness of simulated-force-reflecting teleoperation efficiency.

INTRODUCTION

In order to establish on-orbit manipulation and rendezvous-docking technologies, a satellite mounted with a robot manipulator will be launched in 1997[1]. The experiments involve a challenging attempt of master-slave teleoperation from the ground, which raises the crucial problem of communication time delay between the onboard system and the ground system. It is expected that a delay of 2 to 4 second will exit in each way, and this delay deteriorates teleoperation efficiency.

In the past, the following methods have been proposed to overcome this delay. The most primitive is the move-and-wait strategy, which is time-consuming and increases the fatigue of the operator. The predictive display provides a delay-free clear picture through a predictive simulator on a real-time basis. Even with its support, however, the operator tends to make large operational commands to the robot due to lack of contact force feedback. This situation can cause damage to the equipment or generate vibrations that affect the satellite's attitude. The compliance control of the slave arm is able to accommodate the force that is generated by excessive operational command input, but cause of the limited capacity of computer resources mounted on the satellite, damage or negative affects still occur. The bilateral master-slave manipulation loop is known to be unstable in teleoperations involving time delays above 1 second [5].

In this paper we propose computed-force-reflecting teleoperation using a real-time simulation and show its effectiveness through a typical teleoperation task of peg-in-hole. Originally, a similar idea was proposed in [4], [5], [6], [7], and [8], but the proposals did not include precise evaluations of the idea. We developed a teleoperation system including a display of degraded real images with a time delay, a real-time graphic simulator that provides contact force information, and a predictive display [3]. This enabled us to compare different types of teleoperation in practical basis. In this paper, we also propose a new concept of "virtual collisions in a virtual world". Based on this concept, the constraint force is generated from virtual objects that do not exist in reality. This force guides the slave arm along a safe path and prevents it from colliding with obstacles. With this system, high-precision peg-in-hole tasks involving round-trip communication time delays of up to a few seconds were performed at three support levels: a real image alone, a predictive display with the real image, and a real-time graphic simulator with computed-contact-force reflection and a predictive display. We show the experimental results which demonstrate the effectiveness of simulated-force-reflecting teleoperation.

COLLISIONS IN A VIRTUAL WORLD

In this section, we describe the concept of our teleoperation system. Real collisions and virtual collisions are implemented in the force-reflecting real-time simulator.

Real Collisions in a Virtual World

Real collisions in a virtual world involve collisions of similar objects built in a virtual world as well as the real world. For example, when collisions are generated between the slave arm and the equipment in a virtual world, they would also produce collisions in the real world. We define these collisions as "real collisions in a virtual world". Systems that can feed back the collision forces simulated by these models have been reported [5], [6], [7], [8].

Because direct collisions between objects would produce collisions in the real world as well, they may cause equipment damage or generate vibrations that affect the satellite's attitude. To prevent damage and vibration and also to guide the slave arm to a safe position, we propose the new concept : Virtual collisions in a virtual world.

Virtual Collisions in a Virtual World

"Virtual collisions in a virtual world" are collisions between virtual objects that do not exist in reality. Collisions between virtual objects would not produce collisions in the real world because they do not exist in the real world. If this concept is applied to the master-slave operation, constraint force is generated from the virtual objects. It works to lead the slave arm to a safe position and to avoid direct contact with objects. We can create virtual objects of any kind in a virtual world, therefore, we can define a variety of constraint environments that do not exit in reality.

SOFTWARE IMPLEMENTATION

This section reviews the concept of real collisions and then discusses the generation of a constraint force from virtual collisions.

Implementation of Real Collisions in a Virtual World

A collision generated between the slave arm and an object on the satellite is noted as a real collision . It would also produce a collision in a virtual world, because similar objects are built in the virtual world. The force is calculated as a simple spring-loaded model as in the following equation (1).

$$F_b = K \Delta r \qquad (1)$$

where **Fb** is the force of collision as viewed from the slave arm base coordinate system Σ **b**. **K** is a stiffness. Δ **r** denotes the distance between the surface and the current position of the tip of modeled slave arm.

Implementation of Virtual Collisions

To generate a constraint force needed to move the slave arm to a target position (see **Figure 1**), a virtual collision is considered. No collision is induced in the real world. Our 5-step method to generate a constraint force is as follows:

[Step 1] Calculate the collision force **Fb** at the virtual collision point as viewed from the slave arm base coordinate system.

"Collision point virtual frames" are coordinate systems that can be set to any point of the object collision. They are set at the each collision point when a virtual collision occurs between the slave arm and the virtual object (see **Figure 1**). There are twice as many as virtual frames as there are collision points.

[Step 2] Convert the value of **Fb** to **Fv**, the force of a collision point virtual frame.

$$\mathbf{F_v} = {}^{v}\mathbf{R_b}\,\mathbf{F_b} \qquad (2)$$

where ${}^{v}\mathbf{R_b}$ is a transformation matrix from the slave arm coordinate system to the collision point virtual frame.

[Step 3] Set the force sensor coordinate system, Σ **c**, on any position on the slave arm. This coordinate position corresponds to a position on the handle of the master arm.

[Step 4] Apply a virtual collision force, **Fv**, to the collision point virtual frame relative to the virtual collision point on the slave arm. Calculate the force, **fc**, and the moment, **nc**, by using the real-time simulator function of inverse dynamics calculation processing [3].

[Step 5] Generate the constraint force on the master arm side to facilitate operation.

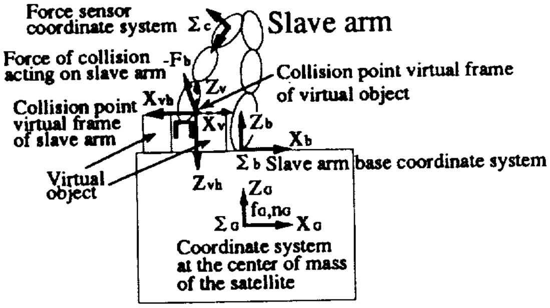


Figure 1 . Virtual collision

Application for a Peg-in-Hole Task

<u>**Virtual Collision Model for a Peg-in-Hole Task**</u> Virtual collision forms vary according to the task nature. In a peg-in-hole task application, for example, proper positioning during movement to the vicinity of the hole should permit inserting the peg into the hole through constrained movement only in the z-axis direction. Safe, efficient, and reliable positioning is ensured by the use of a virtual collision model as shown in **Figure 2**. In this model, a virtual wire is set at the tip of the peg, a variable virtual column is set at the center of the hole opening, and a virtual plane is set around the hole.

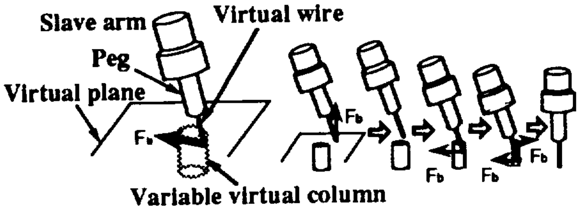


Figure 2. Virtual collision model

The model is characterized by :

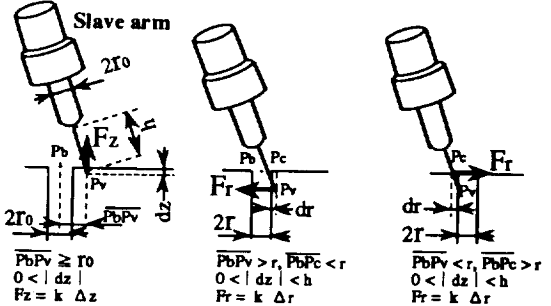
(1) A real-time collision calculation resulting from the use of a virtual wire and a variable virtual column.

(2) Responsiveness to changes in the attitude of the tip of the slave arm derived from the use of a virtual wire.

(3) Reduced radius of the variable virtual column upon insertion of the virtual wire. As a results, a constraint force is generated to erect the virtual wire. A second order function is used as a radius function.

Constraint Force Calculation Processing The following sections describe the procedure for calculating the constraint force to be fed back to the operator through the master arm.

[Step 1] Calculate the constraint force **Fb** as viewed from the slave arm base coordinate system Σ **b**. This virtual collision case is shown in **Figure 3**.



(a) Collision between the virtual wire and the virtual plane

(b) Collision between the tip of the virtual wire and the variable virtual column

(c) Collision between the virtual wire and the variable virtual column

Figure 3. Virtual wire collisions

[Step 2] Calculate the force, **fc**, and the moment, **nc**, acting upon the force sensor coordinate system as mentioned in the above section. Feed back this force and moment to the master arm on a real-time basis. This force enables the operator to reach the center of the hole opening quickly and safely. An operator can sense the constraint force through the master arm. A damping term is attached to the force **Fb** to stabilize the transition to a non-collision state [7].

$$\mathbf{F_b} = \mathbf{K}\Delta \mathbf{r} + \mathbf{D}\Delta\dot{\mathbf{r}} \tag{3}$$

where **K** is a stiffness, **D** is a damping coefficient, and Δ **r** denotes the distance between the surface of the virtual object and the current position of the virtual wire.

EXPERIMENT

Peg-in-Hole Tasks with Three Teleoperation System

Peg-in-hole tasks involving round-trip communication time delays were performed at three support levels: a real image alone, a predictive display with the real image, and a real-time graphic simulator with computed-contact-force reflection and a predictive display. The round-trip communication time delays used were 0, 4, and 8 seconds. The peg was 30 mm in diameter. The clearance between the peg and the hole was 0.9 mm. The depth of hole was 20 mm. The slave arm controller had local compliant control to avoid damaging the equipment.

Teleoperation System Configuration

Our system configuration is shown in **Figure 4**. The operator traces a path through the 6-DOF master arm while viewing the slave arm on a real image or predictive display. Simulated collision force is fed back to the operator through the master arm. The real slave arm follows the created path with time delays. **Figure 5** shows a man-machine interface system. The monitor display, force-reflecting simulator display, and master arm are arranged from left to right.

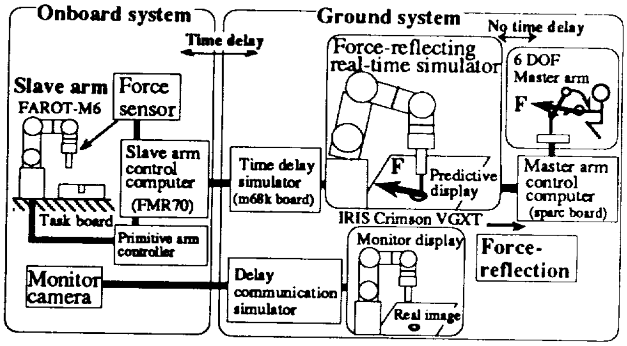


Figure 4. System configuration

Figure 5. Man-machine interface system

RESULTS

The experimental results are indicated below. **Figure 6** is a diagram of task time when using a real image. Longer time delays prolong the completion times. For an 8-second time delay, an operator supported by a real image alone could not perform "move-and-wait" type teleoperation.

We compared the support levels of teleoperation with round-trip time delay of 4 seconds. **Figure 7** is a diagram of the total time of each task. The task time for the simulated-force-reflecting teleoperation is the shortest. The constraint force reduces the time needed to move to the vicinity of the hole. The time of peg insertion was not affected by the teleoperation support levels, because peg insertion was carried out through constrained movement in the z-axis direction. **Figure 8** is a diagram of force sensor maximums. The sensor maximum for the simulated-force-reflecting teleoperation is also the lowest. **Figure 9** shows the sample records of force sensor measurements. For both the real image and the predictive display, large amplitude and vibration were measured, while for the simulated-force-reflecting teleoperation, the measurements varied much less. **Table 1** summarizes the effects of support levels and lists the success rates. We regarded the result as a successful execution when the peg was inserted into the hole. The success rate was calculated from several trials. The success rate of using the force-reflecting simulator is 100%. Overall, the simulated-force-reflecting teleoperation returned the best performance. The force-reflecting simulator provided us essentially identical performance even for 8-second time delays.

CONCLUSION

This study demonstrates the effectiveness of simulated-force-reflecting teleoperation. The experimental results with peg-in-hole tasks indicates the best teleoperation efficiency was provided by the force-reflecting simulator. The results also demonstrate the effectiveness of teleoperation based on the concept of virtual collisions in a virtual world. Feedback of a constraint force from virtual objects results in safer, more efficient, and more reliable task execution. We plan to apply these new teleoperation concepts to such tasks as paddle expansion and screw tightening.

REFERENCES

[1] M.Oda, Y.Wakabayashi, S.Ichikawa, R.Imai, and T.Anzai, 1992. ETS-VII, The World First Telerobotic Satellite (Mission And Its Design Concept),*Artificial Intelligence, Robotics And Automation, In Space*,307-318.

[2] Whitney,D.E., 1985. Historical Perspective and State of the Art in Robot Force Control, *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 262-268.

[3] F.Nagashima,Y.Nakamura,1992. Efficient Computation Scheme for the Kinetic and Inverse Dynamics of a Satellite-Based Manipulator, *Proceeding of the 1992 IEEE International Conference on Robotics and Automation*, vol.1, 905-912.

[4] Richard Paul, Thomas Lindsay, and Craig Sayers, 1992. Time Delay Insensitive Teleoperation, *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 247-254.

[5] W.S.Kim, B.Hannaford, and A.K.Bejczy, 1992. Force-Reflection and Shared Compliant Control in Operating Telemanipulators with Time Delay, *IEEE Trans. on Robotics and Automation*, vol.8, no.2, 176-185.

[6] T.Kotoku, K.Tanie, and A.Fujikawa, 1990. Force-Reflecting Bilateral Master-Slave Teleoperation System in Virtual Environment, *Proceedings of the i-SAIRAS '90*, 295-298.

[7] T.Kotoku, 1992. A Predictive Display with Force Feedback and its Application to Remote Manipulation System with Transmission Time Delay, *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.1, 239-246.

[8] S.Tachi, H.Arai, T.Maeda,1990. Tele-existence master slave system for remote manipulation (II), *Proceedings of the 29th IEEE Conference on Decision and Control*, Vol.1, 85-90.

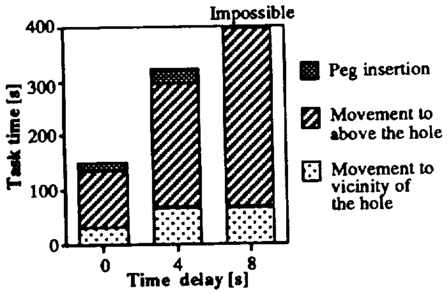


Figure 6. Task time with real image support

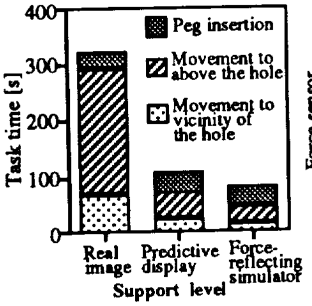


Figure 7. Task time
(Time delay : 4 [s])

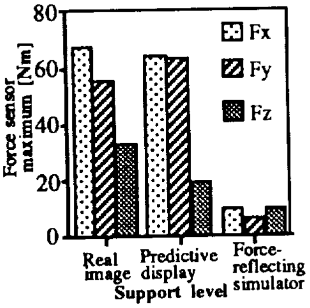
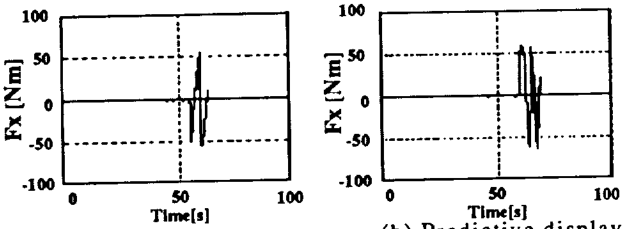


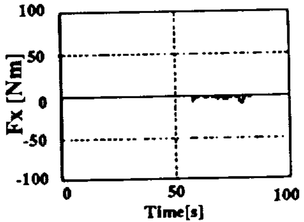
Figure 8.
Force sensor maximum
(Time delay : 4 [s])



(a) Real image

(b) Predictive display

(Time delay : 4 [s])



(c) Force-reflecting simulator

Figure 9. Sample records of force sensor measurements

Table 1 Comparison of the teleoperation support levels

Support level	Time delay [s][1]	Task time [s]				force sensor maximum |Fx|[N] |Fy|[N] |Fz|[N]	Success rate [%]
		Movement to above the hole	Movement to vicinity of the hole	Peg insertion	Total		
Real image[2]	4	68	225	29	322	67 55 33	38
Predictive display[3]	4	25	45	35	105	64 63 19	80
Force-reflecting simulator[3]	4	18	30	33	81	9.9 6.7 9.6	100
Force-reflecting simulator	8	14	30	32	76	6.8 3.6 10.0	100

1 Round-trip
2 Resolution:352 by 240,gray levels:16, frame rate:10 frames/s
3 Frame rate: 8 frames/s IRIS CrimsonVGXT (85MIPS,16MFLOPS,180kPolygon/s)

Telerobotics Test Bed for Space Structure Assembly

N95-23734

M. Kitami and K. Ogimoto
Space Systems Engineering Div.
Kawasaki Heavy Industries Ltd.
1 Kawasaki-cho, Kakamigahara city, Gifu Pref. 504, Japan
Tel:+81-583-82-8170, Fax:+81-583-82-1499
F. Yasumoto, T. Katsuragawa, T. Itoko and Y. Kurosaki
System Technology Development Center
Kawasaki Heavy Industries Ltd.
1-1 Kawasaki-cho, Akashi city 673, Japan
S. Hirai and K. Machida
Electrotechnical Laboratory,
Ministry of International Trade and Industry
1-1-4 Umezono, Tsukuba city, Ibaraki 305, Japan

KEYWORDS AND PHRASES

Telerobotics, Space robotics, Solar cells, Intelligent monitoring, Test bed

INTRODUCTION

A cooperative research on super long distance space telerobotics is now in progress both in Japan and USA (1).

In this program, several key features will be tested, which can be applicable to the control of space robots as well as to terrestrial robots. local (control) and remote(work) sites will be shared between Electrotechnical Lab.(ETL) of MITI in Japan and Jet Propulsion Lab.(JPL) in USA. The details of a test bed for this international program are discussed in this report.

TASK ANALYSIS

Task Decomposition

A space structure, which is supposed to be a part of a large solar power station, will be assembled with the telerobotics.

The assembly work has been decomposed into several tasks, ie,

(1) Deployment of the truss structure.
(2) Installation of an ORU(Orbital Replaceable Unit).
(3) Deployment of a solar cell panel.
(4) Installation of a wire harness.

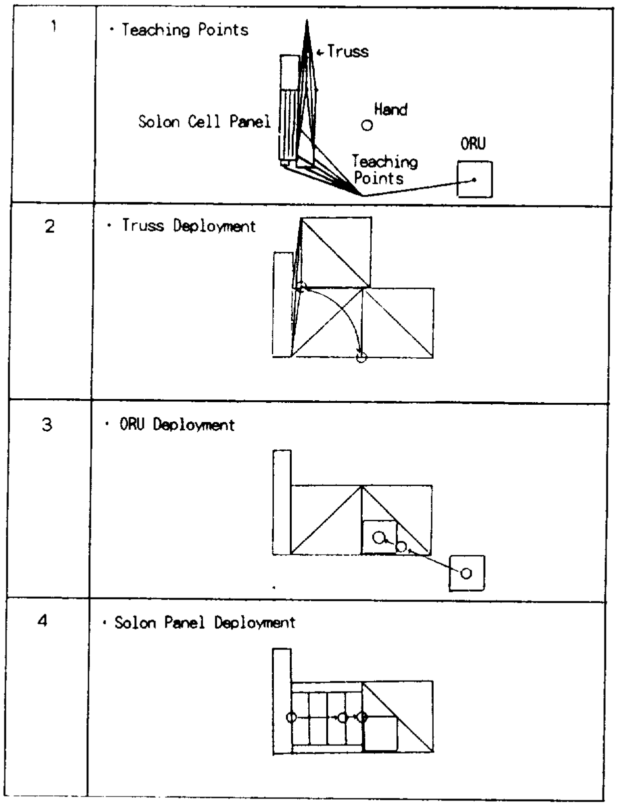


Fig. 1 Task Sequence

Each task is split into small events such as shown is Fig. 1

The time required for each event has been also evaluated.

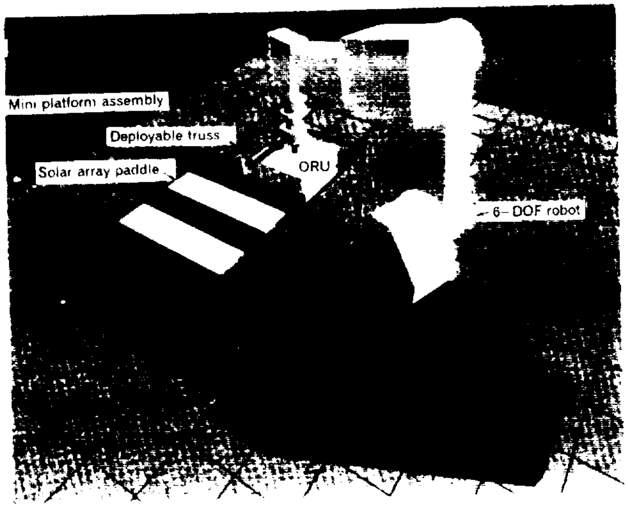


Fig. 2 Concept of the Remote Site in Japan

Off-Line Task Validation by the Graphics Simulator

To perform the teleoperation under the constraints of time-delay and limited capacity in a communication line, all the task sequence will be verified using the off-line graphics simulator (Fig. 2), before the execution of the tasks. The simulator will be operated based on a world model of the remote site stored and maintained in the knowledge base.

TELEROBOTIC CONTROL STRATEGY FOR SPACE STRUCTURE ASSEMBLY

The control strategy in the system is embedded in three different blocks of programs, namely,

(1) An intelligent monitoring system to control the viewing scheme (2).
(2) A knowledge base as object oriented programs to perform required tasks.
(3) A cooperative control system to cope with the teleoperation of a robot.

The knowledge base is the key element of this system, and the object oriented programs contain data of the work site and define procedures necessary to perform tasks. It accepts task commands described as message to an object model from an operator, and generates motions for both the robot and cameras. Those generated motions can be displayed on the graphics simulator for the confimation of the task. Generated motion for the robot is sent to the cooperative control system on the remote site. It achieves the servo control of the robot. It also accepts a direct motion control command from the operator, generated by a master-manipulator, a joystick or other commanding devices. The basic flow of the software is shown Fig. 3

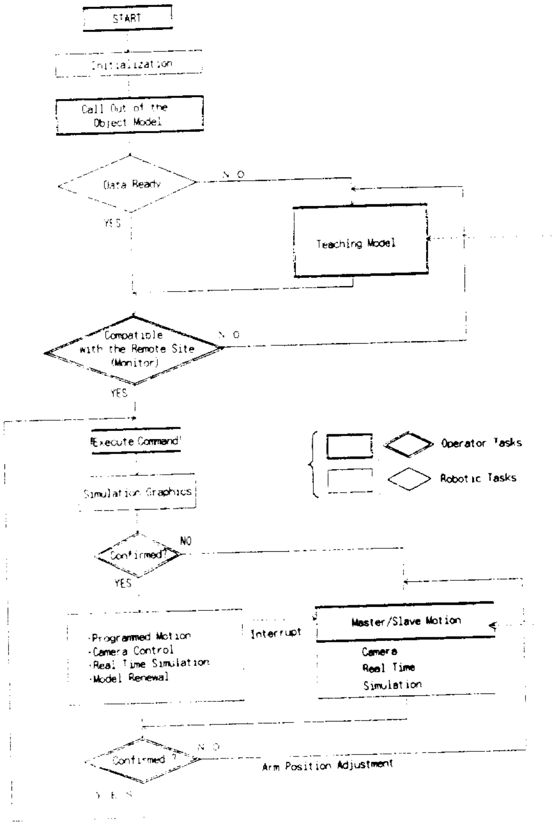


Fig. 3 Software Basic Flow

DESIGN OF TESTBED

System Architecture

A schematic diagram of the system is shown in Fig. 4. The system is split into two parts, one for a remote site which includes the manipulator and the various sensors, and the other for a local site which includes various computers and control software (Fig. 5). Most communication lines are connected through the Ethernet, and the time delay can be introduced in the image and command lines.

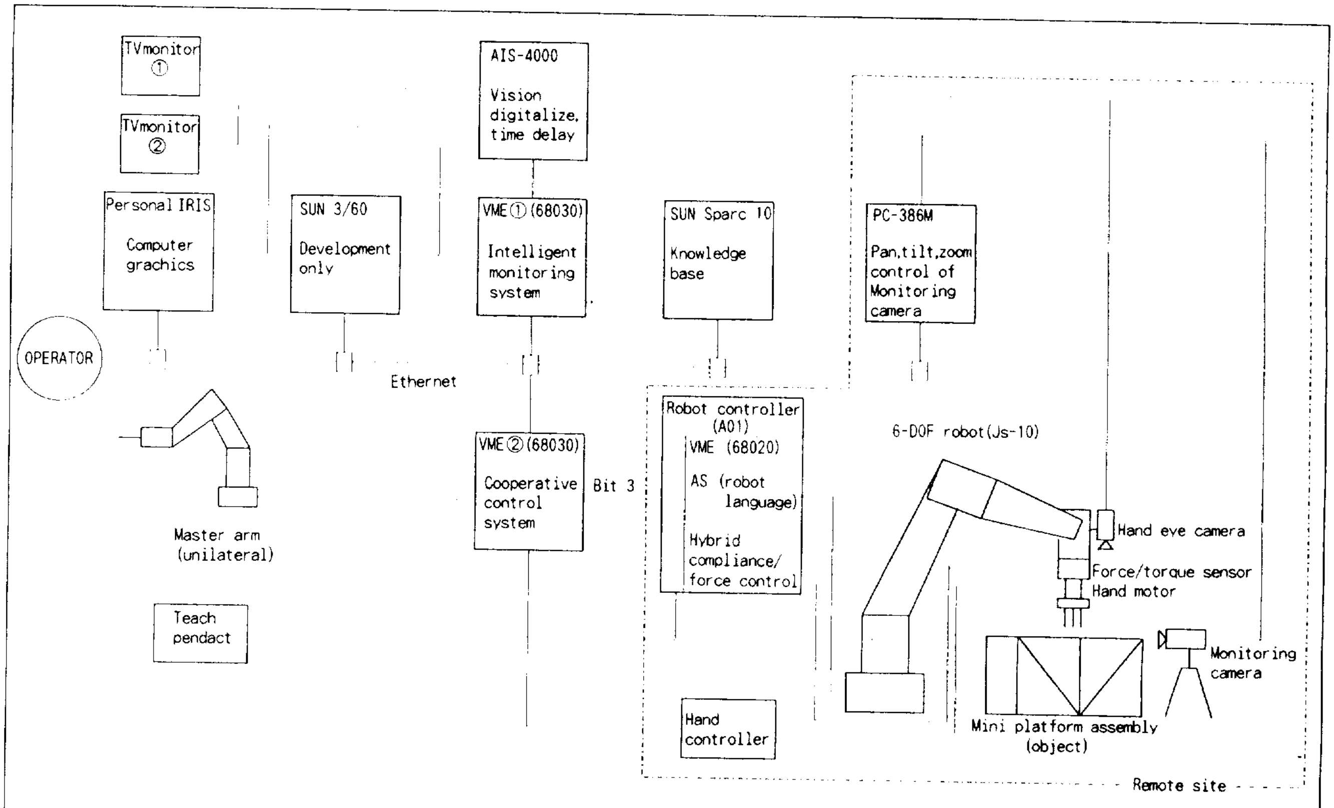


Fig. 4 Ling Distance Space Telerobotics Experiment System

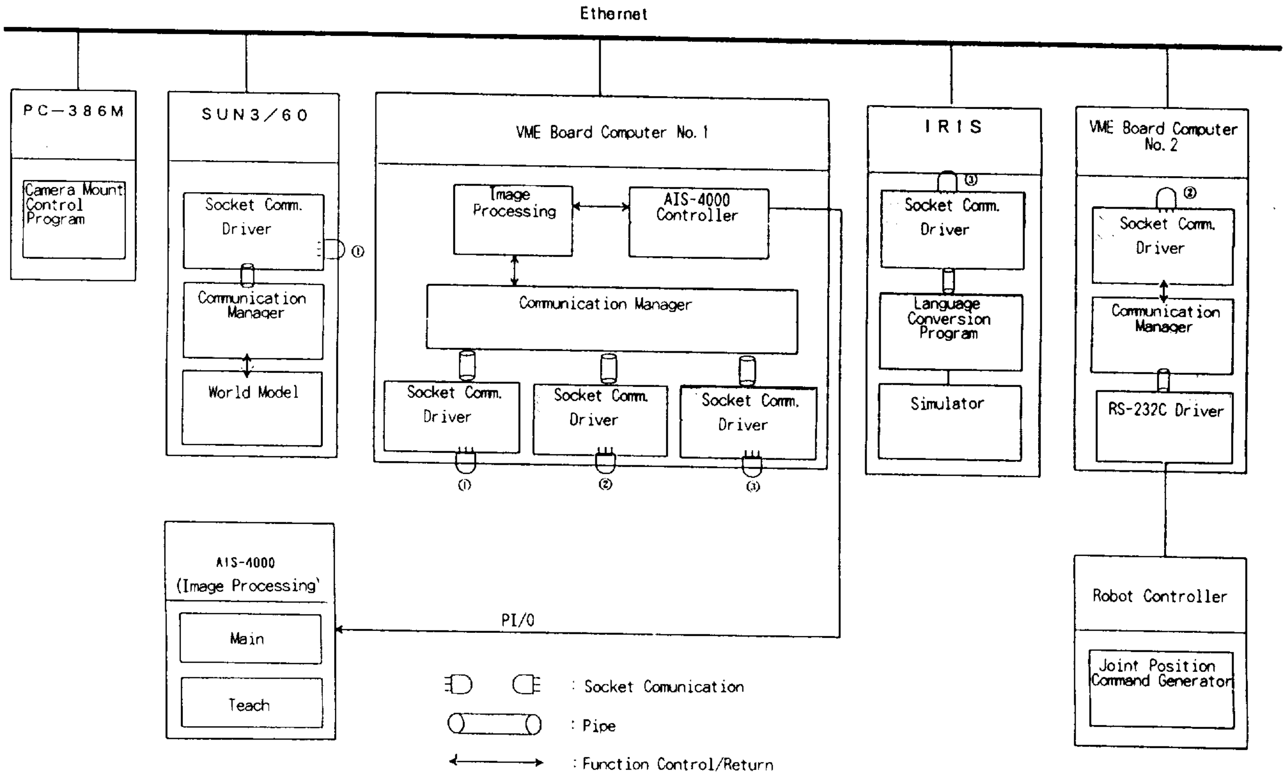


Fig. 5 System Architecture

Robot Friendly Truss Structure:

A two-cell truss structure has been designed and manufactured, with a simple and passive latch in each cell. This latch can be easily released by the remotely operated robot, and can be tailored for a robot assembled truss structure. This truss has provisions for the installation of an ORU and the deployment of a solar panel, and is supposed to be a part of the main structure for a solar power generation system.

Description of Hardware

The following hardware other than the truss structure has been manufactured and prepared;

(1) An industrial robot with a hand eye camera, a force-torque sensor and a three-finger hand.

(2) A robot controller with a hybrid compliance-force control capability.

(3) Two TV monitors with image processing capability.

(4) Two workstations with graphics capability.

The prepared testbed for the teleoperation experiment is shown in Fig.6.

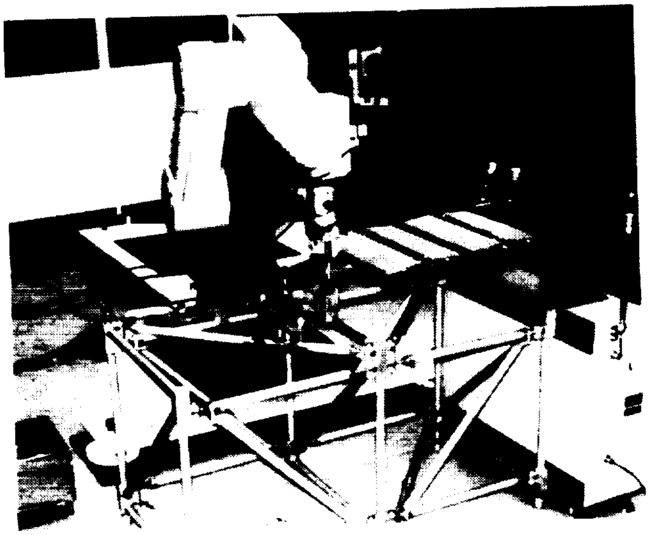


Fig. 6 Testbed for Teleoperation

FUTURE WORKS AND CONCLUSIONS

This telerobotics testbed will be completed by the end of this year, and various robotic tasks will be demonstrated. The first step will be the assembly of a space structure placed in Japan, which will be controlled from the local site in JPL, and the second set of the experiment (3) will be performed in the year after the next.

Once the performance of the system is verified, successive tasks will be planned to prepare for the future application of this technology in space, particularly for the deployment and assembly of a solar power generation system in space.

REFERENCES

(1) Machida.K; Hirai.S and Schenker.P 1993, Study on Long Distance Space Telerobotics: Concept of Interoperation Experiment between Japan and USA. Proceedings of the 37th Space Sciences and Technology Conference.213-214.Tokyo (In Japanese)

(2) Wakita, Y. Hirai.S and Machida, K 1994, Intelligent Monitoring System Applied to Super Long Distance Telerobotic Task. Proceedings of i-SAIRAS' 94 Los Angeles.

(3) Schenker, P.S and Kim, W.S 1994, Remote Robotic Operations and Graphics-Based Operator Interfaces, Proc. 5th Intl. Symp. on Robotics and Manufacturing, Maui.

N95- 23735

Telepresence Control of a Dual-Arm Dexterous Robot

Larry Li [1], Brian Cox [2], Susan Shelton [2], Myron Diftler [2]

[1] NASA Johnson Space Center
[2] Lockheed Engineering and Sciences Company
Houston, Texas 77058
713-483-9160(voice) 713-483-7580(fax)
li@aio.jsc.nasa.gov

KEY WORDS AND PHRASES

Anthropomorphic, dexterous robotics, human-factors, telepresence, virtual reality

ABSTRACT

Telepresence is an approach to teleoperation that provides egocentric, intuitive interactions between an operator and a remote environment. This approach takes advantage of the natural cognitive and sensory-motor skills of an on-orbit crew and effectively transfers them to a slave robot. A dual-arm dexterous robot operating under telepresence control has been developed and is being evaluated. Preliminary evaluation revealed several important observations that suggest the directions of future enhancement.

INTRODUCTION

The current approaches to robot teleoperation in the Space Shuttle as well as the International Space Station Alpha (ISSA) are based on "joystick" type hand controllers. The visual feedback is provided by multiple cameras, many of which are mounted on the robot arms. This approach to teleoperation is similar to the cock-pit design of fighter aircraft. For manipulator control, this approach can be counter-intuitive, and may overload the visual and manual capacities of the operator. The problem becomes even more amplified when the slave robot is not designed to reflect the degree of dexterity the human operator possesses. As a result, the operator's skill is not effectively transferred to the slave robot. A different approach to robot teleoperation is *telepresence*. In telepresence, the master control and feedback devices are designed to maximize the use of the operator's innate cognitive and sensory-motor skills [1][2].

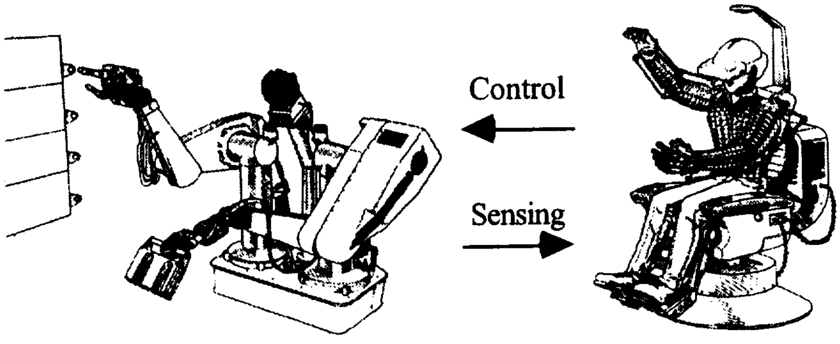
The following describes the Phase I activities of an evolving robotics testbed at the NASA Johnson Space Center (JSC). The testbed system, called the Dexterous Anthropomorphic Robotic Testbed (DART), has telepresence as its baseline operating mode. Ultimately, DART will be operating under *shared control*, where telepresence control of the robot will be augmented by intelligent automation. DART is controlled by the Full Immersion Telepresence Testbed (FITT), the interface to the human operator [2].

PHASE I OBJECTIVE

The DART Phase I objective is to develop, demonstrate, and optimize a baseline telepresence system. The steps to achieve this goal include: (1) developing a dexterous telerobotic system with telepresence control; (2) developing a flexible, modular, shared control architecture; and (3) conducting comparative evaluation of telepresence versus other types of controls. The following will primarily focus on the activities leading to the accomplishment of Step (1) and (2). Preliminary evaluations, in partial fulfillment of Step (3), will also be described.

SYSTEM OVERVIEW

Most telepresence applications require at least two functional components: *master* and *slave.* The master component is usually the operator's telepresence interface, and the slave is usually an emulator of the master. In our setup, FITT acts as the master that controls DART, the slave robot. The FITT and DART systems are shown conceptually in figure 1.



(a) (b)

Figure 1. Telepresence control of a dual-arm dexterous robot. (a) The Dexterous Anthropomorphic Robotic Testbed. (b) The Full-Immersion Telepresence Testbed (concept drawing).

Full-Immersion Telepresence Testbed

FITT provides intuitive control of DART. This testbed immerses the operator in the robot's environment and links human and robot motions and senses as transparently as possible to provide a natural feel to the operator. The FITT system includes interfaces for controlling DART's head camera unit, arms, hands, and base. A Virtual Research™ helmet displays DART's stereo camera images with a 100 degree field of view. The depth perception provided with stereo imaging is one of the testbed's most important immersion features. A Polhemus™ tracker located on the top of the helmet commands the orientation of DART's head camera unit. The same type of sensor is also attached to each of the operator's wrists, providing full 6 degree-of-freedom (DOF) control of the robotic arms. EXOS™ hand masters worn by the operator provide gripper and joint level finger control in dexterous teleoperation.

Dexterous Anthropomorphic Robotic Testbed

DART, shown in figure 2, includes several robotic devices, controllers, and supporting workstations. The robotic arms are PUMA 562's with an 8.8 pound payload capability. Each arm also has a force-torque sensor. On the right arm is a Stanford/JPL hand. Each finger has a urethane fingertip to provide a high static friction surface and can be hyperextended to provide a large manipulation envelope. On the left arm is a parallel jaw gripper. The head camera unit that provides video feedback to the teleoperator supports 3 DOF rotations and contains two color CCD cameras. The driver level software is executed on two Tadpole™ multiprocessor systems. Each multiprocessor system has four M88000 processors and runs a multiprocessor version of the UNIX operating system. The vision system is implemented on a DataCube™ pipeline image processor board.

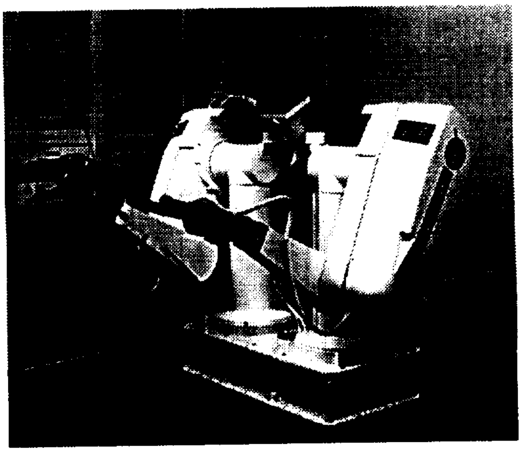


Figure 2. The Dexterous Anthropomorphic Robotic Testbed (DART).

The DART system is controlled via a modular, distributed control architecture as shown in figure 3. Each subsystem spans one or more processes. The subsystem processes are distributed across three separate computers, networked on an Ethernet backbone. The subsystems communicate and are synchronized by high-level communication software called the Tele-Robotics Interconnection Protocol (TelRIP) [4]. This architecture provides a flexible environment for development, maintenance, and future enhancements. FITT controls DART by linking to this Ethernet backbone and commanding the subsystems through TelRIP. The router process, denoted by **R**, is responsible for transmitting data to the appropriate subsystem processes.

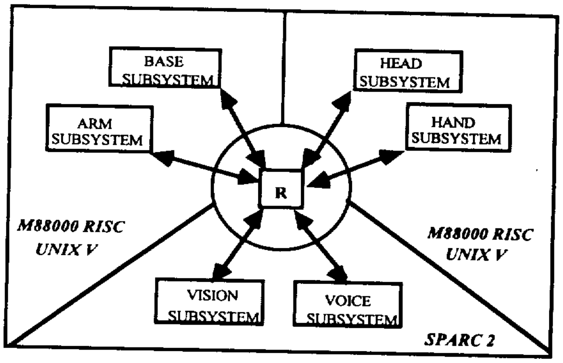


Figure 3. DART's distributed control architecture.

PRELIMINARY EVALUATIONS

Preliminary evaluations of the DART and FITT systems are currently being performed using operators of varying skill levels, ranging from several years of robotic experience to absolutely no engineering experience. This allows the intuitiveness of operation to be qualitatively evaluated. The tasks range from inspection to object handling to dexterous manipulation.

Inspection tasks are comprised mainly of bringing an object towards the head camera and viewing it from different angles. These tasks provide information about the required display resolution, stereo perception, as well as the effect of working with *egocentric* views of the workspace. The object handling tasks include picking up objects of various sizes and shapes (*e.g.*, balls, pipes, tools) and placing them at a different location, and handing objects back and forth between the dexterous hand and the gripper. Some of the dual-hand dexterous tasks performed are tying a knot with a rope, folding and unfolding a thermal blanket, and manipulating an electronic task panel which contains toggle and rocker switches, push buttons, sliders, and a dial. These tasks reflect some of the basic dexterity and skills required for on-orbit extra- and intra-vehicular activities (EVA/IVA).

OBSERVATIONS

One of the most significant observations from the preliminary evaluations is the short time it takes a new operator to become proficient with the system. For example, operators with no previous experience were able to transfer objects between the two hands and manipulate the controls on the panel within a 30 minute session. Operators with considerable experience in "cock-pit" type control have also found the training time greatly reduced due to the intuitiveness of the motion controls and the immersiveness of the visual feedback.

The weight of the exoskeleton hand masters causes muscle fatigue when the system is used for long duration. This limitation presents some difficulties when it is necessary to maintain a specific position for a long period of time. This observation suggests the need for a mechanism that will allow the operator to re-adjust his or her arm positions (*e.g.*, indexing).

While teleoperation of the dexterous hand offers much flexibility for grasping, it was found inadequate for manipulation. The difficulty lies in the inability of the operator to preshape the hand and execute manipulation primitives, such as turning and pinching, in a consistent manner.

The operator can experience mild motion sickness when using the system due to a slight delay between the motions of the operator's head and the DART camera system. This only occurs when the operator makes large, quick head movements. Motion sickness usually occurs whenever there is a significant mismatch between the robot's and the operator's rate of motion. Motion sickness can also be caused by unintended body and head movements. However, since the operator rarely has to make large head movements once focused on a task, this problem is not a major prohibiting factor.

Although the current system provides the necessary visual cues to perform many tasks, a few limitations of the visual feedback have been observed. The visual feedback the operator receives is coarse (320 X 240 pixels) and the distance between the head cameras is a little too narrow, so the depth perception of the operator is not optimal. These visual limitations can have serious impacts on the operator's performance. For example, since FITT currently does not offer force-reflection, the operator assesses the force imparted onto the environment by watching for the amount of physical compliance. The active compliance of the DART's fingers is very useful in this regard.

Another problematic area encountered is the transformation of human hand motions to DART's hand motions. Several transformation methods were explored [3]. These methods included joint-to-joint mapping, forward and inverse kinematics transformations, and a combination of joint and Cartesian control. The two major difficulties encountered when applying these techniques are the dissimilar kinematics of the human's and DART's hands, and the slight changes in the sensor positions when the gloves are taken off and put back on. Joint-to-joint mapping was chosen as the method of control due to the computational simplicity and the intuitiveness of the control.

The telepresence evaluations also revealed some interesting operator behaviors. For example, an initial *exercise* is desirable before each session to familiarize the operator with the system's behavior. The exercise typically involves having the operator command the robot's arms, hands and head in various different ways to explore the dexterity of the robot. Without the exercise, less experienced operators often have the tendency to move like a robot, not fully utilizing his or her natural coordination skill. After a few training sessions, the operator generally will learn to compensate for any kinematics dissimiliarities between the operator and the robot.

FUTURE WORK

The results of our initial evaluation have pointed out several areas for improvement. The exoskeleton gloves will be replaced by lightweight Cybergloves^{TM} to reduce fatigue. A position indexing mechanism will be implemented to allow the operator to reposition his or her arms while the robot remains still. Grasp and manipulation primitives modulated by operator's hand movements will be developed for tasks that require a high degree of accuracy and control. A second generation head camera unit will be fabricated to provide a tighter head tracking and to correct the narrow interpupilary distance. A high-resolution (640 X 480 pixels) head-mounted displays will be sought to improve operator's visual acuity.

A force-reflective dexterous arm master, developed by EXOS^{TM}, will be integrated with FITT to evaluate the effect of force-reflection. Additional evaluations will be conducted to quantify the performance of the DART/FITT system. New test subjects will be recruited to study the correlation between training-time versus performance, and the performance of "cock-pit" type control versus telepresence.

CONCLUSION

Telepresence is *not* a new idea. It is, however, an idea that is becoming a reality due to the recent advances in head-mounted display, dexterous glove controller, motion trackers, force-reflective masters, and other human compatible interactive devices. The DART and FITT combination represents an integration of these telepresence technologies for space robotics applications. Many lessons were learned in our preliminary evaluations. While several areas for improvement were identified, the benefit of telepresence in space robotics is clearly evident by the variety of complex tasks DART/FITT can perform under the control of an operator with minimal training.

REFERENCES

[1] Held, R. M.; Durlach, N. I., 1992. Forum Spotlight on: The Concept of Telepresence: Telepresence. *Presence*, 1:109-112.

[2] Li, L., 1993. Virtual Reality and Telepresence Applications in Space Robotics. *Virtual Reality Systems Magazine*, 1:50-56.

[3] Speeter, T. H., 1992. Transforming Human Hand Motion for Telemanipulation. *Presence*, 1:63-79.

[4] Wise, J. D.; Ciscon, L., 1992. *TelRIP Distributed Applications Environment Operator's Manual*. Technical Report 9103 George R. Brown School of Engineering, Rice University.

SESSION VI

Vision

Mark Tracking: Position/Orientation Measurement using 4-Circle Mark and its Tracking Experiments

Shinji Kanda, Keijyu Okabayashi, Tsugito Maruyama, and Takashi Uchiyama

Fujitsu Laboratories Limited
1015 Kamikodanaka, Nakahara-ku, Kawasaki-shi 211
Japan
Tel: +81-44-754-2659 Fax: +81-44-754-2582
E-Mail: kanda@flab.fujitsu.co.jp

KEY WORDS AND PHRASES

Mark, position and orientation measurement, robot, tracking, visual feedback control

ABSTRACT

Future space robots require position and orientation tracking with visual feedback control to track and capture floating objects and satellites. We developed a four-circle mark that is useful for this purpose. With this mark, four geometric center positions as feature points can be extracted from the mark by simple image processing. We also developed a position and orientation measurement method that uses the four feature points in our mark. The mark gave good enough image measurement accuracy to let space robots approach and contact objects. A visual feedback control system using this mark enabled a robot arm to track a target object accurately. The control system was able to tolerate a time delay of 2 seconds.

INTRODUCTION

The National Space Development Agency of Japan (NASDA) plans to conduct a series of space robot experiments on Engineering Test Satellite 7 (ETS-7)[1][2] scheduled to be launched in 1997. All experiments will study the feasibility of basic, rather than advanced, functions of the space robot. Technology for tracking objects using visual feedback control is essential to implementing tracking and capturing floating objects and satellites. The first step is to develop a technology that enables a robot arm to track a mark.

Our four-circle mark has four circles in a square. The position and orientation of the mark can be calculated by solving a perspective n-point problem from the four feature points extracted from the four circles by simple image processing. The robot arm tracked the mark using the results of mark position and orientation measurements.

PROBLEMS ADDRESSED BY THE MARK TRACKING EXPERIMENT

A visual feedback control system for a robot to track a mark must work in real-time. Onboard computers, however, have limited capacities, and are too slow to handle a large volume of image data. Processing and measuring the images of the mark being tracked require the following:

(a) A mark that allows feature points to be extracted by simple image processing.
(b) A measurement technique that does not require a high computer load to calculate position and orientation using extracted feature points.

A visual feedback control system also needs a control algorithm that involves less load on the onboard computer.

MEASUREMENT

Position and Orientation Measurement using the Four-Circle Mark

Generally, in a Perspective N-point problem (PnP) [3-5], if three or more feature points extracted from an image, the position and orientation of the mark can be determined from the positional relationships between these feature points and their corresponding points in the image. Our measurement determines the position and orientation of the mark from four feature points in the same plane with known positional relationships and their corresponding points in the image. This has the advantages of fewer feature points of interest, a unique solution, and lower calculation load.

Figure 1 shows the position and orientation measurement using the four feature points. The transformation matrix T, which represents translation and rotation, denotes the position and orientations of the mark [6]. Vectors α , β and γ , A, B, and C, and O and O' are related to each other by a matrix that is represented by the product of the transformation matrix T and the perspective transformation matrix P. The six linear equations derived from these relationships can be ordered as shown below.

$$MN = B \qquad (1)$$

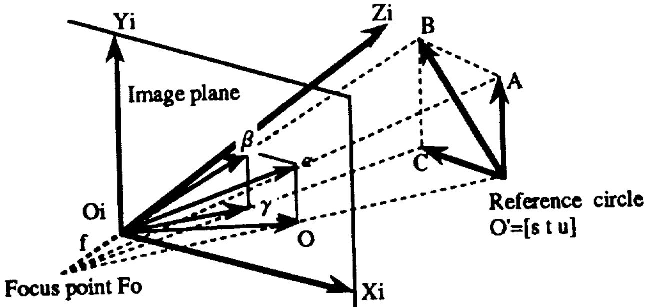
$$M=\begin{bmatrix} fa_1 & fa_2 & 0 & 0 & -\alpha_1 a_1 & -\alpha_1 a_2 \\ 0 & 0 & fa_1 & fa_2 & -\alpha_2 a_1 & -\alpha_2 a_2 \\ fb_1 & fb_2 & 0 & 0 & -\beta_1 b_1 & -\beta_1 b_2 \\ 0 & 0 & fb_1 & fb_2 & -\beta_2 b_1 & -\beta_2 b_2 \\ fc_1 & fc_2 & 0 & 0 & -\gamma_1 c_1 & -\gamma_1 c_2 \\ 0 & 0 & fc_1 & fc_2 & -\gamma_2 c_1 & -\gamma_2 c_2 \end{bmatrix} \qquad (2)$$

$$N=\left[\frac{n_{11}}{(u-f)}\ \frac{n_{21}}{(u-f)}\ \frac{n_{31}}{(u-f)}\ \frac{n_{12}}{(u-f)}\ \frac{n_{22}}{(u-f)}\ \frac{n_{32}}{(u-f)}\right]^T \qquad (3)$$

The N vector can be solved by applying the inverse matrix of M to both sides of Eq.(1). Since $nx=[n11, n21, n31]^T$ and $ny=[n12, n22, n32]^T$ have a norm of 1, the values of u, nx, and ny can be determined from the N vector. The value of nz can be determined from the outer product of nx and ny, and the values of s and t can be determined from the relationship between O and O'.

Mark Geometry

The geometry of the tracking mark should allow feature points to be extracted accurately by simple image processing. The tracking mark should not be affected by variations in illumination in space. To meet these requirements, we chose a mark with four black circles placed in a square in a white plane (Figure 2). The geometric center positions of the four circles are associated with four feature points. The position in the image that corresponds to each feature point can be determined by calculating the weighted mean from the vertical and horizontal projection deviations of the circle. This results in an imaging accuracy of one subpixel. One of the four circles is made larger than the others to define the correspondence between the feature points in the mark and the corresponding points in the image.



$$T=\begin{bmatrix} n_{11} & n_{12} & n_{13} & s \\ n_{21} & n_{22} & n_{23} & t \\ n_{31} & n_{32} & n_{33} & u \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad P=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/f & -1 \end{bmatrix}$$

$A=[a1\ a2\ 0]^T \quad B=[b1\ b2\ 0]^T \quad C=[c1\ c2\ 0]^T \quad O=[o1\ o2\ 0]^T$

$\alpha=[\alpha 1\ \alpha 2\ 0]^T \quad \beta=[\beta 1\ \beta 2\ 0]^T \quad \gamma=[\gamma 1\ \gamma 2\ 0]^T$

Fig. 1 Position/orientation measurement using 4-circle mark

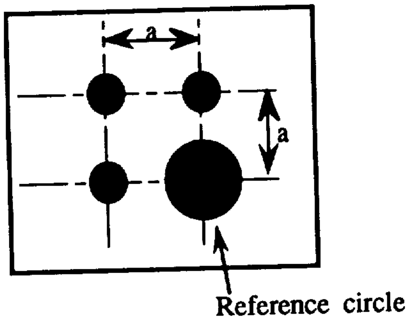


Fig. 2 Four-circle mark

TRACKING ALGORITHM

The robot arm is controlled in the mark tracking experiment by the basic control system shown in Figure 3. This proportional plus derivative (PD) control system calculates the deviation in the position and orientation of the tip of the robot arm from those of the mark by image processing and measurement. This deviation is multiplied by a proportional gain and a differential gain to generate a travel velocity command for the tip of the robot arm. This travel velocity command is transformed into the base coordinate system of the robot, then translated again into

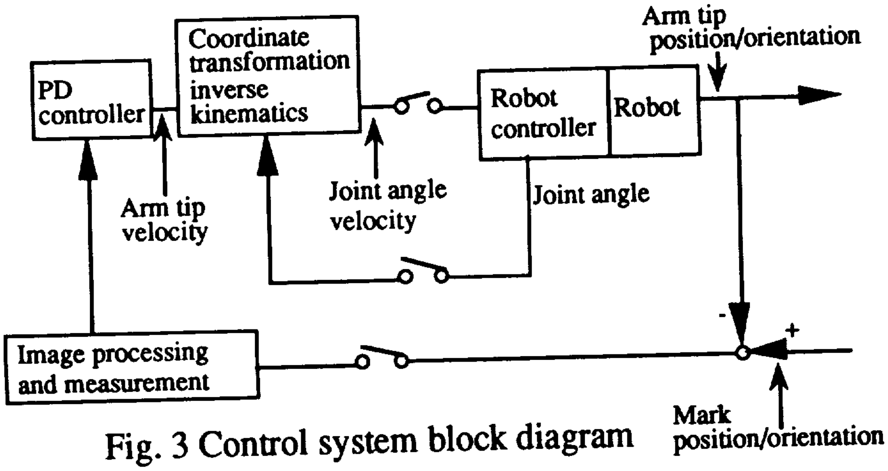


Fig. 3 Control system block diagram

the joint coordinate system as a joint angle velocity command. The joint angle velocity command is fed back to the robot controller as a velocity command.

EXPERIMENTS

Experimental System Setup

Figure 4 shows the Experimental System Setup. The robot arm has the same six degrees of freedom as the orbiting arm on the ETS-7. The arm is manipulated by inputting joint angle velocity commands to the robot controller. For mark movement, an XYZ- θ stage capable of moving the mark with four degrees of freedom was used to simulate the behavior of a slow spinning satellite.

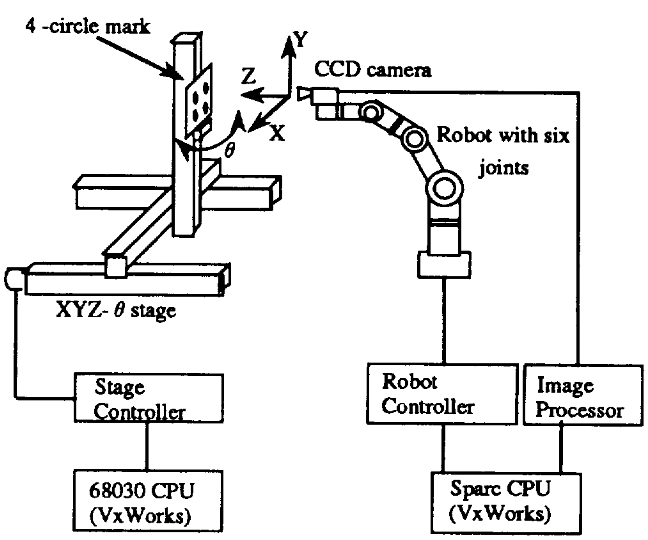


Fig. 4 Experimental system setup

Experiment Results and Discussions

Figure 5 shows the measurement accuracy with respect to the distance to the mark. The mark has a circle-to-circle distance of 100 mm. Translation errors are less than 5 mm up to the distance of 700 mm in the x and y axis directions, and less than 2% of the distance in the z axis direction. Orientation errors are less than 2 degrees in the roll, pitch, and yaw rotations. These errors are sufficient for the robot arm, which has force control to approach an object of interest.

Figure 6 shows the measurement accuracy at a distance of 500 mm when the mark rotates around y axis (pitch angle). Accuracy drops as the pitch angle increases. At a pitch angle of 20 degrees, measurement errors are about 2% of the measurement distance in the x,y, and z directions, and about 2 de-

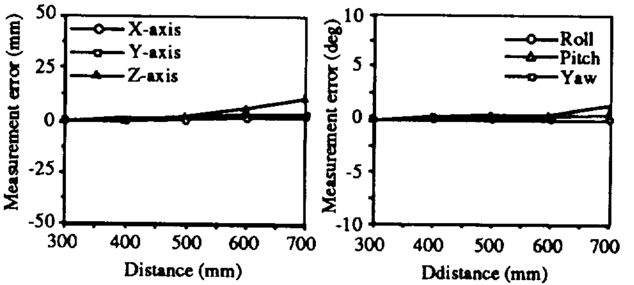


Fig. 5 Errors in position and orientation measurement with respect to the distance (Circle-to-circle distance: 100 mm)

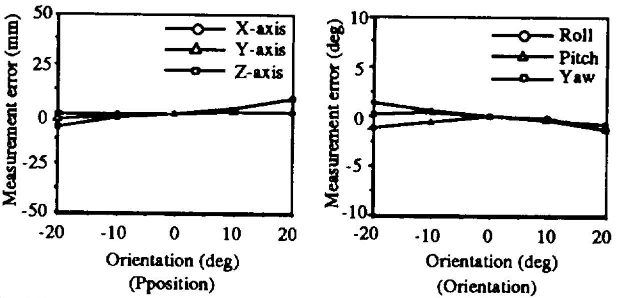


Fig. 6 Errors in position and orientation measurement with respect to the orientation (Circle-to-circle distance: 100 mm)

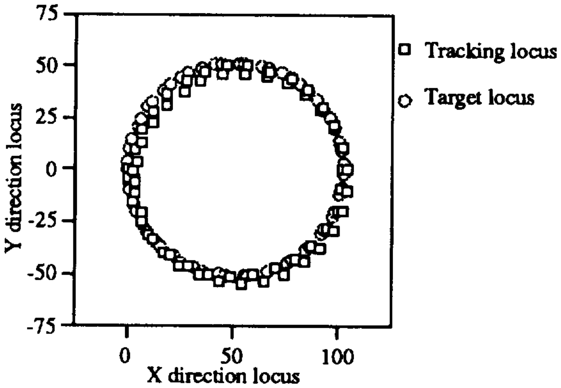


Fig. 7 Target locus and tracking locus in the X-Y plane

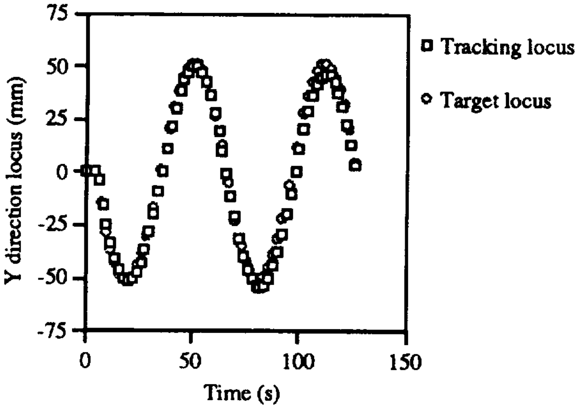


Fig. 8 Time response of the target locus and tracking locus in the Y-axis direction

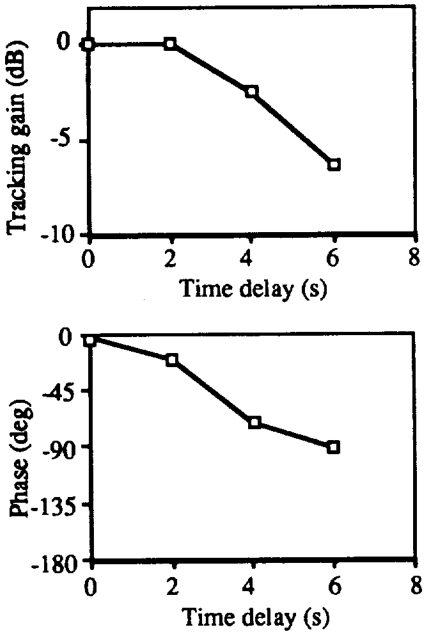


Fig. 9 Tracking gain and phase by time delay
mark velocity: 60 s/cycle
With a time delay of 8 seconds, tracking fails and the mark moves outside the image

grees in orientation angle. This loss of accuracy is caused by distortion of the circles in the image plane, if the pitch angle is large. Accuracy at an orientation angle around 0 degrees is sufficient for the robot arm to approach the mark perpendicularly.

Based on our findings, we conducted a four-circle mark tracking experiment by simulating the behavior of a slow spinning satellite as a target for tracking and capturing. The slow spinning satellite has the mark on its tip. As the mark moves, it draws a circular locus with a diameter of 100 mm. With a circle-to-circle distance of 50 mm, the mark travels in a circular locus at a rate of 60 seconds per rotation. Figure 7 shows the target locus and the tracking locus in the X-Y plane with respect to the behavior of the mark at a distance of 500 mm. Figure 8 shows the tracking locus in the Y direction. The results show that the robot arm tracked the locus of the mark accurately, with a phase delay of about 5 degrees. The sampling time from image processing to computing the joint angle velocity was about 0.2 s at a distance of 500 mm. These performances enable the robot arm to track and capture floating objects.

Figure 9 shows tracking gain and phase delay as time delay increases. The robot arm could track the target locus accurately with time delays up to 2 seconds, and the phase delay was about 20 degrees.

CONCLUSIONS

We developed a real-time position and orientation measurement method that uses a four-circle mark. These simple image processing and measurement algorithms will be used in orbit.

The measurement method was accurate enough to enable the space robot to approach an object of interest.

We built a visual feedback control system using the four-circle mark and conducted mark tracking experiments. The robot arm tracked the locus of the mark with a phase delay of about 5 degrees with respect to the locus of the mark. Accuracy was therefore good enough to track and capture objects.

We also performed a mark tracking experiment that used tracking data with a time delay to simulate teleoperation from the ground. The robot arm tracked the target locus accurately with phase delays for time delays of up to 2 seconds. Furthermore, predictive tracking control will be effective in tracking objects accurately when the tracking data has large time delay.

REFERENCE

[1] Oda, M., et al., 1992., ETS-VII, The World First Telerobotic Satellite(Mission and its Design Concept), *Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Toulouse, France: 307-318.

[2] Machida, K., et al., 1992., Space Experiment Plan of Advanced Robotic Hand for ETS-VII, *Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Toulouse, France: 219-223.

[3] Shimaaki, M., 1979., Some Discussions on Inverse of Projection", *IEICE Technical Report, IE79-15*: 83-92.

[4] Machida, K., et al., 1988., Prototype of Short Range Sensor for Proximity Operation in Space, *Proc.of 2nd Space Artificial-intelligence /Robotics / Automation Symposium*, Tokyo, Japan: 51-54.

[5] Morikawa, H., et.al., 1990., A Prototype Space Telerobotic System, *Proc. of IEEE international Workshop on Intelligent Robots and Systems* : 421-428.

[6] Paul, R.P., 1981., *Robot Manipulators: Mathematics, Programming, and Control,* The MIT Press

A Fuzzy Structural Matching Scheme for Space Robotics Vision

Masao NAKA[1], Hiromichi YAMAMOTO[1], Khozo HOMMA[1], Yoshitaka IWATA[2]
1)Computational Sciences Division, National Aerospace Laboratory
7-44-1 Jindaiji-higashi-machi, Chofu, Tokyo 182, JAPAN
Phone;(0422)47-5911, ext.2525,Fax;(0422)42-0566, Email;naka@nal.go.jp
2)Scientific and Technological Systems Dept.II, Fujitsu F.I.P.Inc.
4-14 Kamiyama-machi , Sibuya, Tokyo 150, JAPAN
Phone;(03)3481-4186, Fax,(03) 3481-9554

N95- 23737

KEYWORDS AND PHRASES

Fuzzy logic, stereo matching, robot vision, space robotics

ABSTRACT

In this paper, we propose a new fuzzy structural matching scheme for space stereo vision which is based on the fuzzy properties of regions of images and effectively reduces the computational burden in the following low level matching process. Three dimensional distance images of a space truss structural model are estimated using this scheme from stereo images sensed by Charge Coupled Device (CCD)TV cameras.

INTRODUCTION

The importance of advanced space vision processing has increased in the space station era for servicing, maintenance, repairs and assembly by space teleoperated /robotics systems. A stereo vision system with passive optical image sensors (CCD TV cameras) is the simplest method of sensing and perceiving the three dimensional (3-D) distances and attitude parameters of targets objects using the stereo process of matching and the principle of triangulation. Because of the simplicity of the imaging equipment and recent advance in the processing speed of computers, this stereo vision system is expected to become a key technology for space automation and robotics [Ref. 1,2]. However, the techniques of stereo images processing are insufficient so far, only the disparity map of targets objects can be extracted from their stereo images of uncooperative (without special reflector or pattern marker) targets objects in the space environment.

In this paper, we propose a new fuzzy structural matching scheme and give an example of its application for a space truss structural model. This paper consists of two parts. In the first part, a scheme for higher level structural stereo matching algorithms is discussed in terms of the fuzzy based properties of labeled coarse regions of stereo images. In the second part, the evaluation experiments for the fuzzy stereo matching scheme and fuzzy-based feature extraction are carried out using CCD images of simple objects (a space structural model).

HIGHER LEVEL STRUCTURAL MATCHING ALGORITHMS

Matching of the left and right images, the so-called correspondence problem, is one of the most critical subjects in the field of computer vision processing. Finding conjugate points of images can be performed in several ways such as gray-level matching, local correlation methods, edge matching, and dynamic programming. However, these methods are less reliable and less efficient than higher level structural methods as such segment- or region- based methods, because these methods depend basically on point matching.

A proposed matching algorithm is fundamentally based on the matching boundary-representation (B-rep) of stereo images as proposed by F. Tomita et al [Ref. 3]. To apply this method to the images of objects resulting in space optical environments, we modified this matching scheme using the fuzzy set theory for region matching as a higher level structural correspondence [Ref. 4]. An outline of the proposed matching scheme with fuzzy properties of regions is as follows. The properties of each region of segmented images consist of perimeter L, averaged gray level X, and aspect ratio T. The similarity measure of the perimeter for correspondence between the left region (i) and the right region (j) of images is defined as follows.

$$P_L(i,j)=\begin{cases} 1 & \Delta l_{ij}\le m_l-\sigma_l \\ -\dfrac{1}{2\sigma_l}[\Delta l_{ij}-(m_l-\sigma_l)]+1 & \Delta l_{ij}\ge m_l-\sigma_l \end{cases} \qquad (1)$$

$$\textbf{where}\quad \Delta l_{ij} = abs(l_{Li} - l_{Ri})$$

$$m_l = \frac{1}{MN}\times\sum_{i=1}^{N}\sum_{j=1}^{M}\Delta l_{ij}$$

$$\sigma_l = \frac{1}{MN}\times[\sum_{i=1}^{N}\sum_{j=1}^{M}(m_l-\Delta l_{ij})2]^{1/2}$$

l_{Li}; perimeter of $i-th$ region of left image

l_{Rj}; perimeter of $j-th$ region of right image

M,N; number of regions

The characteristic curve of membership function of this equation is shown in Figure 1.

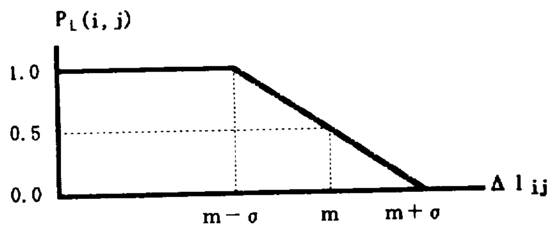


Figure 1. Membership function of PL(perimeter)

This similar equation is also used for other property X. The similarity measure of the aspect ratio T is defined as follows.

$$P_t(i,j)=\begin{cases} \Delta t_{ij} & \Delta t_{ij}\le 1 \\ \dfrac{1}{\Delta t_{ij}} & \Delta t_{ij}>1 \end{cases} \qquad (2)$$

$$\textbf{where}\ \Delta t_{ij} = t_{Li}/t_{Rj}$$

t_{Li}; aspect ratio of i - th regions of left image

t_{Rj}: aspect ratio of j - th regions of right image

The Total similarity measure of region properties for correspondence between the left and right regions of images, R(i,j), is calculated as follows:

$$R(i,j) = Min(P_L(i,j), P_X(i,j), P_T(i,j)) \qquad (3)$$
$$\text{and}\ R(i,j) > \text{threshold}\ R_O,$$

This higher level structural correspondence (region based) is performed at the first phase of matching computations. In this way, the search space of the second phase for low level structural matching (segment based) can be narrowed. Total processing flow of this hierarchical matching is shown in Figure 2.

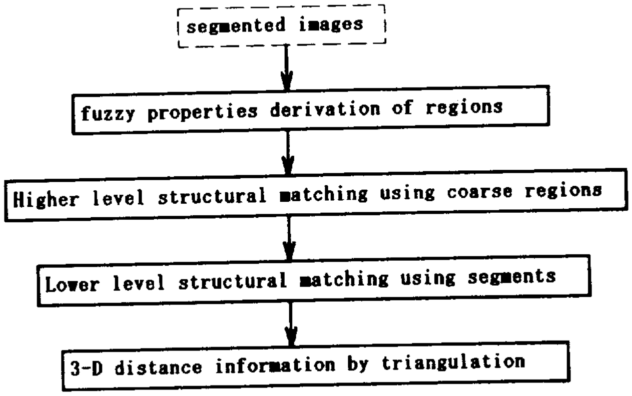


Figure 2. Hierarchical structural matching scheme using B-rep of stereo images

LABORATORY EVALUATION EXPERIMENTS

The evaluation experiments of the fuzzy-based edge feature extraction and stereo matching algorithms were performed using CCD images from models of target

objects. As a simple model of target objects, we use a building block model and a space truss structural model setup on an optical bench in the laboratory. The low contrast and/or blurred digital images of the target objects are generated by a CCD TV camera onboard a x-y traverse device and a Sun work station equipped with an image digitizer.
A digital image is represented by an array of 512X480 (=MxN) pixels, and 8-bit gray levels. Image processing flow for evaluation experiments is shown in Figure 3.

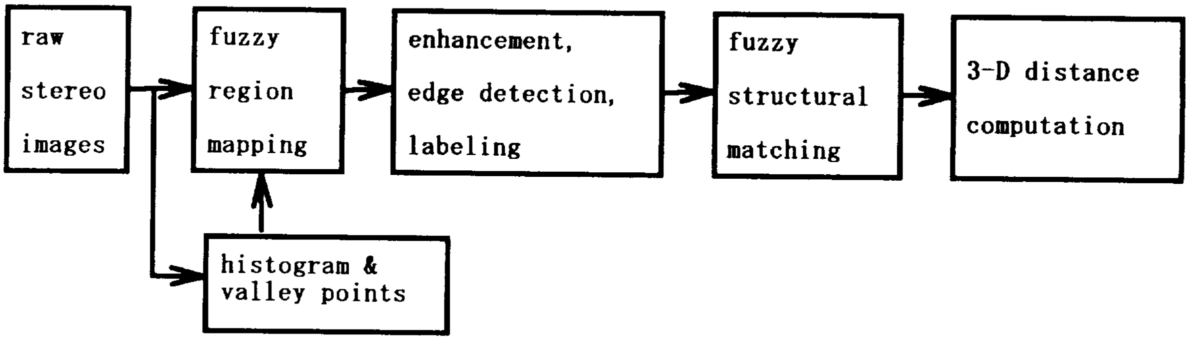


Figure 3. Image processing flow for estimation of 3-D distance informations

Figure 4(a) shows an original stereo image of a space truss structural model. The edge feature detected and labeled image from the original image are shown in Figure 4(b). A 3-D distance image of the object estimated by the proposed matching scheme and triangulation for labeled images is shown in Figure 4(c). In this experiment, the number of coarse regions correspondence was reduced to almost half by using a proposed structural matching scheme.

CONCLUSIONS

As a part of the research on stereo vision processing algorithms for aerospace applications, a stereo matching scheme based on the fuzzy set theory has been studied. It has been concluded that: (1) A higher level structural matching scheme for labeled images is very effective for reducing computational burden, (2) Three dimensional distances are relatively well estimated in a block model and a space truss structural model.

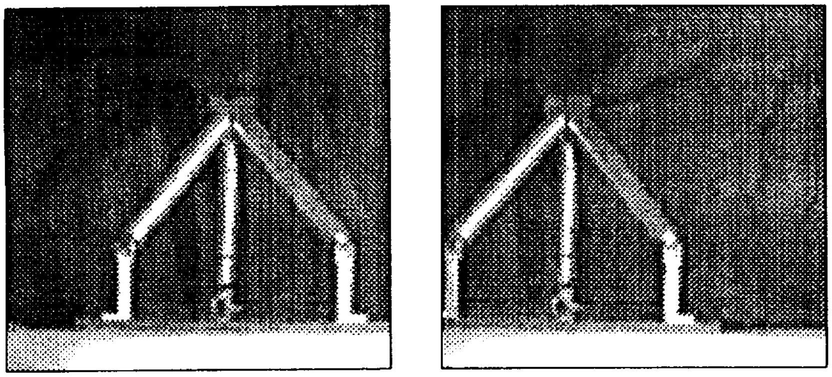
REFERENCES

[1] Krishen, K.,1987.*Vision technology/ algorithm for space robotic applications*, NASA CP, 441-453.

[2] Montemerlo, M.D.,1990.*Automation and Robotics for Space Operation and Planetary Exploration*, Proc. of i-SAIRAS '90.

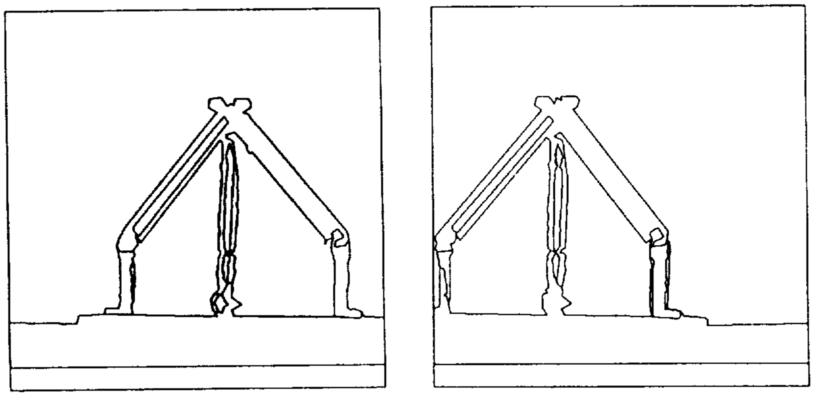
[3] Tomita, F.,et al, 1988.*Matching Boundary-representation of stereo images*, Trans.of IEICE,vol.J71-D,no6.

[4] Naka, M., et al, 1993.*Three Dimensional Distance Information techniques based on the concept of fuzzy set for future space intelligent vision system*, Proc.of Asian Conference on Computer Vision '93.



(left) (right)

Figure 4(a) Original stereo image of a space truss structural model



(left) (right)

Figure 4(b) Segmentation images of a space truss structural model

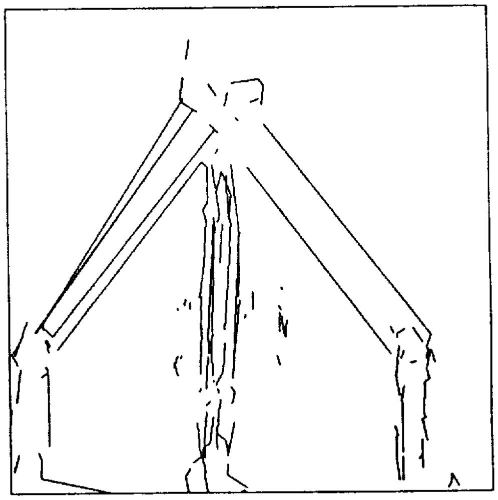


Figure 4(c) 3-D distance image of a space truss structural model

Active Vision in Satellite Scene Analysis

N95- 23738

Martine NAILLON
DASSAULT AVIATION
Research and Technology Cooperation
78, Quai Marcel Dassault
Cedex 300 92552 Saint-Cloud Cedex - FRANCE
Tel (33).1.47.11.52.87 Fax (33).1.47.11.32.19
E. mail : naillon@dassault-avion.fr

INTRODUCTION

In earth observation or planetary exploration it is necessary to have more and more autonomous systems, able to adapt to unpredictable situations.

This imposes the use, in artificial systems, of new concepts in cognition, based on the fact that perception should not be separated from recognition and decision making levels. This means that low level signal processing (perception level) should interact with symbolic and high level processing (decision level).

This paper is going to describe the new concept of *active vision*, implemented in Distributed Artificial Intelligence by Dassault Aviation following a "structuralist" principle. An application to spatial image interpretation is given, oriented toward flexible robotics.

TECHNOLOGICAL PRINCIPLES

In Cognitive Sciences, it is admitted that autonomous systems function following two main principles [12].

First Principle : Internal Organization

This means that an autonomous system has to be internally distributed and self-organized to catch the distributed knowledge of the outside environment. This principle determines the "structuralist" hypothesis.

Second Principle : External Interaction

This means that the system has to interact with its environment to actively adapt its own perception to unpredictable surrounding.

Those two principles are currently driving research in industry in complex system design, to establish a theory for adaptive cognitive distributed systems or multi-systems [9].

Knowledge Representation

In terms of knowledge representation, the classical assumption of the existence of a formal objective model *(cognitivist hypothesis)* becomes insufficient for autonomous system design. It is necessary, in the way this knowledge is processed by the autonomous system, to add a subjective link between the outside word and the system. This link is named an "eco-relation".

In other words, knowledge of an autonomous system cannot be defined only objectively, but has to be actively re-defined, on line, by the subject itself, in a phenomenological approach .*(constructivist hypothesis)* [8,11]. This is the foundation of active cognition, or active vision.

To implement active cognition principles, Dassault Aviation realizes fundamental studies in distributed systems, based on "structuralism" [2]. As it is proved that an adaptive autonomous system is structurally distributed (internal organization), the point is to study the relation between the connectivity of a large population of interactive internal components and the emergence of their collective adaptive behavior.

These scientific studies at Dassault Aviation are founded on Neurosciences [4] and non linear physics [1,10].

CURRENT REALIZATION : ACTIVE COGNITION MULTI-AGENT ARCHITECTURE

A "structural" generic multi-agent architecture is currently developed at Dassault Aviation. It is applied to different domains.

Architecture Main Characteristics

A Recursive Structural Organization (First Principle of Autonomy) The multi-agent architecture is recursively structured with a given organization composed of macro-structures and micro-structures (figure 1).

The macro-structures correspond to autonomous feedback loops which are groups of agents organized in a servo-control manner. Dedicated links between the agents structure the loops, interconnecting low level agents (subsymbolic signal processing) and high level decision and interpretation agents (symbolic processing).

The micro-structures are sub-agents wich correspond to the decomposition of the macro-structure agents. The structural links between macro and micro structures implement the semantic links between global and local perception, as it is known to be in the cerebral cortex [3].

The multi-agent architecture developed by Dassault Aviation differs from the other ones by a dedicated organization and a recursivity from macro to micro structures. This implements and controls the emergence phenomena in interactive micro-structures, and their collective behavior in a non linear dynamics.

Multi-Constructivist Agents (Second Principle of Autonomy). Each autonomous feedback loop is driven by a *distributed control* implemented in a "constructivist" agent attached to the loop. The "constructivist" agent has just enough knowledge so that one feedback loop can be autonomous for the function it processes.

The adaptation of the system to unpredictable environment is realized by a cooperative behaviour between the loops, via their "constructivist" agents. This cooperative processing realizes the necessary interaction between autonomous macro-structures and their environment.

First Results in Earth Observation Domain

A first prototype of structural multi-agent architecture has been realized. It is composed of four autonomous macro-structures. They are implemented on 4 SUN-workstations, each structural loop corresponding to one station. The 4 workstations constitute a cooperative multi-system.

Each loop implements a cognitive function as recognition, localization or scanning. The active recognition process has been realized in the recognition loop, where the sub-symbolic agent (agent LINE) is composed of image processing algorithms (Visilog software). The recognition agent (agent RECO) is a neural network (Perceptron membrane) [5] trained to recognize pieces of communication ways in an image (road, railways, rivers). The two agents, LINE and RECO, work cooperativelly together. For each new piece, e.g. a piece of road, the feedback recognition loop re-defines, on line and subjectively, the local pattern of a road, so that it can be eventually recognized by the neural network : phenomenological approach. This method has been proved efficient especially in the critical cases as, for example, road crossing points or road-railways junctions.

Each loop works under the control of a "contructivist" agent. The total communication ways are recognized using the cooperation between several loops (or several workstations). Among those is a SCAN structure which follows, piece after piece, a complete communication way (figure 2). Dassault Aviation has developed a new theory in servo-control shape tracking based in advanced control theory [7]. The theory has introduced the new concept of "Shape Lyapunov Functions" to control the derivative of a shape observed by a moving camera [6]. These theoritical results are currently applied at Dassault Aviation in active vision for image interpretation. This process is considered as an active target tracking.

TOWARDS ACTIVE ROBOTICS

The structural multi-agent architecture is designed to simulate perception-based control multi-system as active vision system. Each macro-structure represents one autonomous system, which is by definition in interaction with its environment. Each macro-structure could be embedded in an active autonomous robot. This active robotics processing is simulated in satellite scene analysis on 2 macro-structures, the RECOGNITION and the SCAN structures, as already shown in figure 2. The SCAN structure works as a mobile robot.

The structural architecture implements also the multi-constructivist cooperative process between autonomous macro-structures. This could be used to implement and embed a cooperative work between active robots of a team in *collective robotics*. This active collective robotics principles could be applied to planetary exploration.

CONCLUSION

The *"structural" multi-agent architecture* is a first step toward flexible, modular, cooperative multi-system, built on autonomy principles. The *active vision* principle allows the system to adapt to unpredictable situations.

First experiments are at the moment performed in satellite image interpretation for ecological crisis management and military applications. Cognitive functions as recognition, localization, scanning, are implemented on autonomous macro-structures. Their cooperation is simulated on a network of four cooperative SUN workstations. The experiments could be extended toward active collective robotics, applied for exemple to earth observation or planetary exploration.

The recursive structural principles of the multi-agent architecture developed by Dassault Aviation could be generalized for the design of aerospatial multi-systems in which each system could embedded autonomous structure, all the structures cooperating together (application to aerospatial CIS, Communication and Information System).

[1] Amit D., 1989. *Modeling brain function.* Cambridge university press.

[2] Atlan H., 1991. *Intentional self-organization in nature and the origin of meaning. The role of interpretation.* Rossi, C. and Tiezzi, E., editors, Ecological physical chemistry. Proceedings of an International Workshop.

[3] Burnod Y., 1989. *An adaptive neural network : the cerebral cortex.* Masson, Paris.

[4] Changeux J.P., Dechaene S., 1991. *Neural models of cognitive functions.* Entretiens Science et Défense, Délégation Générale de l'Armement, Ministère de la Défense, France.

[5] Deffuant G., 1992. *Apprentissage des réseaux auto-construits.* Ph. D. diss. Dept. of Cognitive Sciences, Paris VI University.

[6] Doyen L., 1992. Shape Lyapunov functions. *Journal of Mathematical Analysis and applications*.

[7] Doyen L., 1993. *Evolution, contrôle et optimisation de formes*. Ph. D. diss. Dept. of Mathematics, Paris XI University.

[8] Dreyfus H., 1991. *Being-in-the world. A commentary on Heidegger's Being and Time*. Division I. Cambridge, MIT Press.

[9] Naillon M., 1992. *Eco-systémique et futurs systèmes de défense*. DGT 48248 (Internal document), Dassault Aviation.

[10] Nottale L, 1993. *Fractal space-time and microphysics*. World Scientific.

[11] Piaget J., 1967. *Logique et connaissance scientifique*. Paris, Gallimard, La Pléiade.

[12] Varela F., 1989. *Autonomie et connaissance. Essai sur le vivant*. Paris, Seuil.

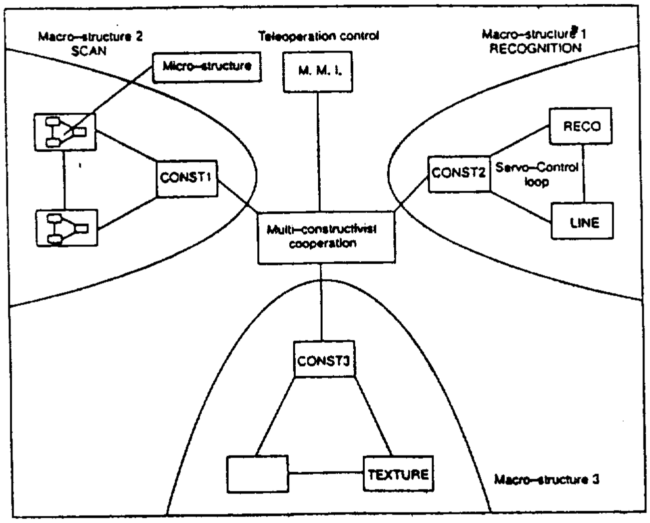


Fig. 1 : Recursive structural organization where each servo-control loop is an autonomous macro-structure driven by a constructivist agent, the loops cooperating together.

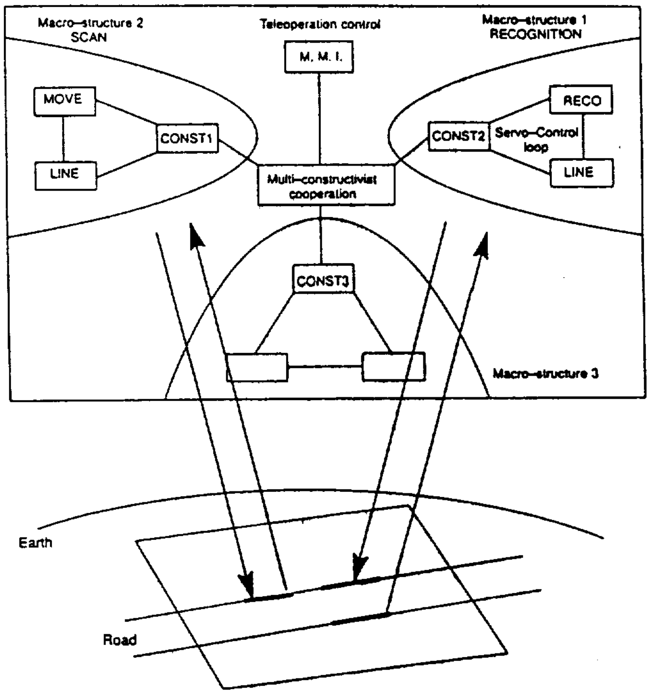


Fig. 2 : Functional cooperative processing between two autonomous macro-structures for active road tracking. The SCAN structure follows the track while the RECOGNITION structure adapts the shape of a road to the recognition process.

Real-Time Tracking Using Stereo and Motion: Visual Perception for Space Robotics

H. Keith Nishihara* **Hans Thomas*** **Eric Huber**† **C. Ann Reid***

*Teleos Research
576 Middlefield Road
Palo Alto, CA 94301 USA
Tel (415) 328-8800 Fax (415) 328-8880
E-mail {keith, hans, car}@teleos.com

†The Mitre Corporation
1120 Nasa Road One, Suite 600
Houston, TX 77058
Tel (713) 483-9939
E-mail huber@aio.jsc.nasa.gov

KEY WORDS AND PHRASES

Computer vision, EVAHR, robotic manipulation, visual perception.

INTRODUCTION

The state-of-the-art in computing technology is rapidly attaining the performance necessary to implement many early vision algorithms at real-time rates. This new capability is helping to accelerate progress in vision research by improving our ability to evaluate the performance of algorithms in dynamic environments. In particular, we are becoming much more aware of the relative stability of various visual measurements in the presence of camera motion and system noise. This new processing speed is also allowing us to raise our sights toward accomplishing much higher-level processing tasks, such as figure-ground separation and active object tracking, in real-time. This paper describes a methodology for using early visual measurements to accomplish higher-level tasks; it then presents an overview of the high-speed accelerators developed at Teleos to support early visual measurements. The final section describes the successful deployment of a real-time vision system to provide visual perception for the Extravehicular Activity Helper/Retriever robotic system in tests aboard NASA's KC135 reduced gravity aircraft.

LOW-LEVEL MEASUREMENTS FOR HIGH-LEVEL VISION TASKS

Computer vision systems typically exist as a primary input to some higher-level process. Although many systems have been constructed where there is limited or no feedback from the high-level process to the vision system, there is an emerging belief in the vision community that incorporating powerful feedback mechanisms will greatly increase the capability and durability of various vision algorithms; this new area of vision research has been termed active vision.

Many new issues are raised when we start to think about visual perception as an active, dynamic process interacting closely with higher-level goal directed behavior. For example, what makes a good measurement in this context? Clearly, a perceptual aid for machine vision ought to recover some basic useful information [1]. Furthermore, it should have an easy-to-model behavior that allows its user to employ it intelligently in new situations.

Two particularly important qualities of a visual measurement are meaningfulness and minimality.

Meaningful. A visual measurement device should derive useful information from the visual scene. This usually means recovering something about the physical surfaces that gave rise to the visual images. Range from stereo, surface orientation, and local image velocity are examples. In addition, there is considerable latitude in how information can be presented as an output, and this can significantly influence the effectiveness of the device for solving perception problems. As far as possible, output from the measurement device should exhibit a consistent, dynamic behavior that encourages the learning of strategies for making more specialized measurements. For example, in the case of a stereo correlator, static estimates of range would be enhanced by information about the shape of the correlation peak used to derive that range and the stability of that information across time and spatial position.

Minimal. A user's ability to exploit a measurement device effectively in a wide range of sensing environments depends to a large extent on how well that user is able to anticipate what the device will do in a new situation. This is easier to do with devices that have consistent, easy-to-model behaviors, and this, in turn, tends to be easier to achieve with simpler measurements. For example, a sensing device that tries to do a lot in one shot, (e.g., a sophisticated but monolithic face recognition system) typically operates on a restricted range of inputs and exhibits extremely non-linear behavior. This makes it difficult to apply in novel imaging environments because one does not have a good model of what it would do, for example, on non-face images. As a side effect, this minimality criterion encourages the use of computations that consume fewer resources and this boosts overall performance.

The combination of these two criteria leads to the question: What is the minimal measurement that produces meaningful information? In the stereo and motion sensing domains, this has led us to some new perspectives on how to define these computational problems. For example, instead of attempting to compute a dense stereo range map, we are focusing on the problem of computing and communicating the results of a single range measurement over a patch of surface. This distinction can be significant when issues of interaction with higher-level knowledge and control are considered.

In stereo matching, for example, a measurement over a small sensing area may fail due to the absence of matchable features. To recover, the calling agent can try switching to a larger measurement window, or it could move the original measurement patch to a slightly different position, or it could decide to move the sensor head to a better vantage point. In any case, the calling agent is aware of the changes made and their implications for the measurement. It is in possession of knowledge of the task to be accomplished, and it is aware of the measurement difficulty and the character of the possibly degraded information obtained. At the same time this agent does not have to know much about the detailed workings of the measurement algorithm itself. As long as it exhibits a consistent and predictable behavior, it can be effectively treated as a black box.

Sign-Correlation Algorithm

The first class of computations studied extensively in this context has been image matching algorithms applicable to stereo range finding and optical flow field measurement. We have developed a computational theory for measuring stereo and motion disparity that is consistent with the measurement-tool objectives and we have had some success at demonstrating the validity of that model for biological systems.

Binocular stereo, the measurement of optical flow, and many alignment tasks involve the measurement of local translation disparities between images. Marr and Poggio's zero-crossing theory made an important contribution towards solving this disparity measurement problem. The zero-crossing theory, however, does not perform well in the presence of moderately large noise levels as has been illustrated by the inability of zero-crossing-based approaches to solve transparent random-dot stereograms—which, interestingly, can be perceived correctly by the human visual system. The sign-correlation algorithm builds on Marr and Poggio's ideas, addressing many of the weaknesses of the original work.

The sign-correlation algorithm continues to use the zero-crossing primitive for matching, but the matching rule is changed. Instead of matching zero contours, we correlate the signal's *sign* in an area. This subtle change makes a significant difference in the behavior of the matcher. Sign-correlation continues to provide useful disparity measurements in high-noise situations long after the zero-crossing boundaries surrounding the signed regions cease to have any similarity. An intuitive explanation of why the two approaches perform so differently follows from the fact that the sign of the convolution signal is preserved near its peaks and valleys long after increasing noise has caused the zero contours to be fully scrambled. Thus, area correlation of the sign representation yields significant correlation peaks even with signal-to-noise ratios of 1 to 1. Since sign-correlation still operates off the zero crossing representation, the key strengths of Marr and Poggio's theory are preserved.

PRISM-3

The sign correlation algorithm has been implemented in the PRISM-3 real-time vision system. A pair of stereo cameras has been mounted on an active pan-tilt-vergence mechanism. The cameras have a stereo baseline of 22.2 cm and the camera vergence angle is computer controlled. The head can move through a 180 degree rotation in under a second and exhibits a positioning repeatability on the order of 50 arc seconds standard deviation in pan, 20 arc seconds in tilt, and 6 arc seconds in vergence.

The two video cameras share the same pixel clock in order to minimize timing skew between the cameras that would result from using only horizontal and vertical video synchronization signals. The left and right camera video is digitized using commercial (DataCube) digitizer hardware, and parallel digital video streams are fed to two dedicated Laplacian-of-Gaussian convolvers (developed by Teleos). These convolvers allow video rate convolution with operator center diameters ranging from 1.6 pixels to 16.6 pixels.

The convolved video signals are fed from the two convolvers to a binary correlator board (also developed at Teleos) which carries out high-speed correlations on the sign bits of the input video streams.

The PRISM-3 correlator board performs 36 correlations in parallel on rectangular windows of adjustable size. The correlator board is operated by an external control processor (currently a 68040 single board computer). At the start of a measurement cycle, this processor writes the pixel coordinates of the next measurement to be made into registers on the correlator along with information about the disparities at which correlation measurements are to be made. A set of correlations with 32 by 32 pixel windows at 36 different disparities takes 100 microseconds to complete. The correlation results are then read into the control processor. If a well formed peak is identified in the data, quadratic interpolation is used to refine the peak disparity. These steps on the CPU take an additional 200 microseconds.

With correlations taken at even pixel disparities at a single vertical disparity, the above 300 microsecond cycle allows a disparity peak to be located in a 72 pixel disparity search range with a third to a tenth of a pixel resolution. Vertical disparity errors between 1 and 2 pixels are well tolerated.

The correlator hardware is also configured to allow correlations to be computed between successive frames from a single camera, allowing optical flow measurements to be made. In the tracking application described below, the system has been programmed to handle image velocities as large as 50 pixels per frame in any direction with subpixel measurement resolution.

The dedicated hardware incorporates standard off-the-shelf TTL components and makes extensive use of field-programmable gate arrays (FPGAs) to achieve high performance while maximizing flexibility in reconfiguring the hardware design.

Tracker Module

Tracking and control applications require fast, low-latency response from the sensor to be of value. A natural limit on speed is the frame rate of the camera system; for most commercially available cameras this is either 30 or 60 frames per second.

At 30 Hz, a person three meters from a camera walking across the field of view at 1 meter per second will traverse about 38 arc minutes per frame. With a 50mm lens the interframe motion disparity will be on the order of 30 pixels. This estimate is for one set of parameters—disparity magnitude varies approximately linearly with lens focal length, subject distance, subject speed, and frame rate—but it gives an indication of the kind of matching performance that will be required to follow human scale motions.

Similarly, the head position control must be responsive to velocity commands at the 30Hz rate with maximum acceleration and velocity limits set sufficiently high to allow smooth pursuit tracking motions.

A tracking system designed to meet these performance specifications was implemented on the PRISM-3 architecture as three subsystems, a low-level electronic tracking system, a mechanical servoing system, and a figure stabilization system. These individual mechanisms operate as loosely coupled parallel process threads. The electronic tracker makes high performance image-based measurements of optical flow and stereo range and attempts to follow electronically an externally designated patch of surface so long as it remains within the camera field of view. The mechanical tracker operates the active camera head in velocity mode using a PID control algorithm. This system attempts to keep the head pointed so that the coordinates of the surface patch tracked by the electronic tracker are kept close to the center of the camera field of view. The figure stabilization submodule uses stereo measurements to assess the extent of the figure associated with the tracked patch. If the tracked patch is not centered on that figure, this module sends an error bias signal to the electronic tracker in an attempt to push it back to the center of the figure. This helps to maintain tracking on figures undergoing rotation that would otherwise lead an optical-flow-based tracking scheme astray.

VISUAL PERCEPTION FOR SPACE ROBOTICS

The Automation and Robotics Division in the Engineering Directorate at the Johnson Space Center recently used PRISM-3 in a successful demonstration of autonomous, vision-guided grasping of a simple target. Testing took place during a flight on NASA's KC135 Reduced Gravity Aircraft as part of Phase 3A of the Extravehicular Activity Retriever/Helper Project (EVAHR). These tests are the first to prove that autonomous robots can use computer vision to guide robotic manipulation and grasp of moving objects in microgravity.

The EVAHR is equipped with a 7-degree-of-freedom robot arm and a dextrous hand consisting of three active and two passive fingers. The PRISM-3 vision system provides the EVAHR's control system with continuous measurements of the position and velocity of a given object, enabling the arm to move to intercept the object. During tests aboard the KC135, a four-inch ball was released to move freely in space during the brief periods of microgravity induced on the aircraft. PRISM located and tracked the ball, enabling the EVAHR to catch it seven times in a number of tries.

Vision-guided grasping of moving objects is a basic skill both in space helper [2] and retrieval tasks and in making the transition from flying to attachment to a spacecraft. Making this transition is particularly demanding as the spacecraft is moving relative to the robot even if the robot is station-keeping with the spacecraft.

Plans are under development to use PRISM-3 in a follow-on EVAHR grasping experiment using more complex targets.

Additional space-related applications are under consideration in two areas: in-space assembly (for example, for operations involving the Shuttle Remote Manipulation System), and in the use of visually-guided Rover navigation for autonomous and/or supervised planetary exploration.

ACKNOWLEDGEMENTS

This report describes research done at Teleos Research supported in part by NASA under contract NAS2-13326, by the Advanced Research Projects Agency under contracts NAS213229 and DACA 76-92-C-0005, by a joint research project supported by the MITRE Corporation, and by Teleos Internal Research and Development funds. The work of Eric Huber at Mitre on the application of stereo machine vision to autonomous grasping as a member of the EVAHR team was carried out in the Laboratories of the Automation and Robotics Division at Johnson Space Center supported by NASA contract NAS9-18057. Jeff Kerr designed the active head mechanism. Many of the ideas reported here were improved through discussions with Teleos staff members especially Stan Rosenschein and Phil Kahn and with Ken Baker at the Johnson Space Center.

REFERENCES

[1] Nishihara, H. K., 1991. *Minimal Meaningful Measurement Tools*. Technical Report, TR-91-01, Teleos Research

[2] Erickson, J. D.; Grimm, K. A.; and Pendleton, T. W., 1994. An Intelligent Robot for Helping Astronauts. In *Proceedings of the Conference on Intelligent Robotics in Field, Factory, Service & Space*, NASA Conference Publication 3251, vol. 1, 421-433.

SESSION PS

Planning and Scheduling Workshop

The Planning and Scheduling Workshop is a single track within the overall i-SAIRAS 94 meeting. It focuses on planning and scheduling as they apply to space exploration, with specific attention to practical, working systems. The workshop includes papers of particular technical interest because they describe fielded planning or scheduling systems and emphasize the reasons for a particular system's success or failure.

The workshop combines formal presentations with opportunities for questions, discussion, and debate among speakers and workshop participants. A number of panels throughout the workshop allow participants to air their views and to exchange ideas about important topics in the area of planning and scheduling.

The theme of the workshop is technology transfer, with specific attention to possible "dual uses" of technology. The workshop attempts to establish connections between technology developed for space and that developed for nonspace (often private industry) markets — especially the manufacturing and airline industries, since they have many characteristics in common with space applications. Presentations in this track include discussions of technology developed in government research labs for particular space applications that can apply to nonspace applications, as well as technology developed for nonspace applications that can sometimes work perfectly for space.

The Planning and Scheduling Workshop comprises the following sessions:

- Session PS-AT *Astronomy Planning and Scheduling*
- Session PS-DS *Decision Support Aspects*
- Session PS-MS *Mission Support*
- Session PS-NT *New Techniques*

Session PS-AT

Planning and Scheduling Workshop: Astronomy Planning and Scheduling

Astronomy Planning and Scheduling

N95- 23740

Planning and Scheduling the Hubble Space Telescope: Practical Application of Advanced Techniques

Glenn E. Miller
Space Telescope Science Institute
3700 San Martin Dr.
Baltimore MD 21218, USA
410-338-4738 410-338-1592
miller@stsci.edu

KEY WORDS AND PHRASES

PLANNING, SCHEDULING, CONSTRAINT SATISFACTION PROBLEM, ARTIFICIAL INTELLIGENCE, ASTRONOMY

INTRODUCTION

NASA's Hubble Space Telescope (HST) is a major astronomical facility that was launched in April, 1990. In late 1993, the first of several planned servicing missions refurbished the telescope, including corrections for a manufacturing flaw in the primary mirror. Orbiting above the distorting effects of the Earth's atmosphere, the HST provides an unrivaled combination of sensitivity, spectral coverage and angular resolution.

The HST is arguably the most complex scientific observatory ever constructed and effective use of this valuable resource required novel approaches to astronomical observation and the development of advanced software systems including techniques to represent scheduling preferences and constraints, a constraint satisfaction problem (CSP) based scheduler and a rule-based planning system. This paper presents a discussion of these systems and the lessons learned from operational experience.

PLANNING

An astronomer wishing to observe with the HST competes for time in a peer-review process. If a proposal is selected, the astronomer submits a detailed observing program which gives specific exposures, instrument configurations and constraints on exposures. There are a variety of scientific reasons why an astronomer might place constraints on exposures: they may be constrained to be executed in a certain order or within a designated time interval. In the case of time-variable phenomena (e.g. variable stars), the proposer may require repeated observations at specific time intervals.

In addition to the constraints imposed by the proposer's scientific program, there are a large number of other constraints which must be considered. Many orbital factors exert a strong influence on scheduling: targets are occulted (blocked) by the Earth for up to 40 minutes each 95 minute orbit. The telescope cannot point too closely to the Sun, Moon or bright Earth limb. The roll orientation of the spacecraft is constrained in order to maintain correct power and thermal balance.

HST observing proposals are prepared using the Remote Proposal Submission System (RPSS) [2]. The astronomer prepares a proposal file, which is a text file in keyword-value format. The entries in this file specify the astronomical targets, exposures, instrument parameters and scientific constraints. The proposer then runs the RPSS *Validation* program which detects problems with the proposal file such as syntax errors, typographical errors, improper values on parameters and missing information.Validation can be performed by logging into a computer at the STScI or downloading the program via Internet and running locally. To our knowledge, RPSS was the first system of its kind for a major scientific installation (it has been in use since early 1986).

The RPSS format proposal describes the observations at a high level. The actual activities which are planned and scheduled by the downstream systems are called *scheduling units* and are specific realizations of the observations including details relating to spacecraft and orbital parameters and instrumental operational scenarios. The process of creating scheduling units from the proposal is called *transformation* and is a planning process. The STScI developed a rule-based expert system to implement transformation. When first proposed in 1984, the concept of an automated transformation of scientific proposals to implementation parameters was quite novel. Since that time, the system has demonstrated the capability to routinely perform this task and allows STScI staff to focus more attention on innovative and difficult observations. Additionally, as improved implementation strategies are devised, Transformation is quickly modified and allows us to re-transform proposals in order to benefit from these improvements. Transformation was originally implemented as a production rule-based system in OPS5, but was rewritten in Lisp as a procedural planning system [1].

Once a proposal is transformed to scheduling units, STScI staff members examine the scheduling opportunities for the proposal using the Spike system (discussed in the next section). Problems found at this stage are fixed by modifying the proposal, e.g. relaxation of observing constraints or choosing an alternate implementation strategy.

We are currently developing a second-generation RPS system which provides two major improvements over the existing system: greater insight into the planning and scheduling process and support for changes to proposals after execution has begun.

Greater insight into the planning and scheduling process is accomplished by providing the proposers with essentially the same tools as used by STScI staff, including Transformation and Spike. Graphical output will show proposers the layout of exposures and telescope activities during each orbital viewing period and the scheduling opportunities during the year, allowing them to see the implications of their choices of observing constraints, instrument parameters, etc. Proposers will also be given explicit control over the assignment of exposures to scheduling units. Previously this was determined by Transformation on the basis of a set of rules. However this was not visible to the proposer and often required several iterations with the STScI to achieve the desired groupings. Transformation will still be used to determine the detailed implementation of activities within a scheduling unit.

In addition, the proposal syntax has been enriched to allow the proposer to specify how observations should be expanded or contracted to make best use of the actual observation time (which cannot be accurately predicted more than a few months in advance).

A severe shortcoming of the current system is that once execution has begun, change to a proposal is a labor-intensive, manual process. The original ground systems were built with the assumption that most proposals would not change after submission. This has turned out to be a very poor assumption - scientific observations often require adjustment based on the results of other observations or to adjust for changes in instrument or telescope performance. Change responsiveness is being addressed in several ways. First, the overall time from proposal preparation to execution is being shortened (by about a factor of two). Second, proposal data and tools are being redesigned to be more modular so that a change to one scheduling

[redacted]

unit or target can be processed independently of others.

SCHEDULING

Scheduling the HST is a challenging problem for several reasons: A year's observing pool consists of tens of thousands of exposures for a few thousand astronomical targets. There are a large number of interacting constraints with timescales covering several orders of magnitude (minutes to years). Scheduling is many months in advance of execution and many constraints cannot be predicted in detail in advance. There is no one overriding factor which determines the schedule so that complex trade-offs between competing factors is necessary. Continuous modification of the schedule is necessary as observations are executed and proposals are changed.

A two-level, hierarchical approach has been used for HST science scheduling by dividing the problem in to long- and short-term scheduling. Long-term scheduling allocates observations over a 1-2 year interval, while short-term scheduling covers a one-week period and creates a detailed timeline of activities. Feedback from the weekly plans is used to update the long-term plan and to reschedule as needed. Long-term scheduling is performed with Spike [3] (developed at the STScI), while detailed short-term scheduling is performed with the Science Planning and Scheduling System (SPSS) which was developed by TRW and extensively modified by the STScI. Important features of Spike include:

- A constraint representation and propagation mechanism (suitability functions) which includes the ability to express human value judgements as well as strict constraints that can never be violated.
- Proposal evaluation tools that allow planners to display and manipulate observations and constraints on workstations.
- Automated and manual scheduling tools based on constraint satisfaction problem (CSP) techniques and a high-level scheduler that combines evidence from competing factors [3,4].
- Automated tools to track the status of the planning and scheduling process at all stages.

Spike is used in two ways. First as an analysis tool for individual proposals and second as a scheduling tool to produce a multi-year plan for an HST observing cycle. As an analysis tool, Spike shows the user (via a graphical interface, postscript plots or alphanumeric reports) the effects of scheduling constraints. It also has an explanation facility which can help a user understand why an observation is unschedulable so constraints can be modified.

As a scheduling tool, Spike is used to create and maintain the long-term plan. As observations are executed and proposals are created or modified, automated and manual tools in Spike are used to update the plan.

Spike was designed with generality in mind and has been adapted to about a dozen other satellite or ground-based observatories. Several of these were feasibility prototypes, but two are in flight operations: the Extreme Ultraviolet Explorer (EUVE) and the X-ray observatory ASCA. Observations with EUVE are sufficiently long (2-3 days) that a division into long- and short-term scheduling is not needed. ASCA uses a two-level hierarchy with Spike performing the long-term scheduling.

Adaptation of Spike to a new system is straightforward and largely consists of defining methods which describe mission-specific elements such as constraints. The core system which includes the constraint representation, propagation, scheduler and user interface are largely unchanged.

The adaptability of Spike was shown in another way as well - prototype short-term schedulers have been implemented for HST, ASCA and the X-ray Timing Explorer satellite. The major changes required to implement short-term scheduling included: the development of a new task which has a variable duration depending on when it is scheduled and that can be preempted (e.g. by Earth occultation or radiation belt passage); more accurate implementation of short-term constraints; a post-scheduler which adjusts task durations to utilitize small gaps remaining in the schedule.

For initial HST operations, long-term scheduling allocated each scheduling unit to a particular week. However this approach was sensitive to perturbations in the short-term schedule: If a scheduling unit could not be executed in the chosen week, this would leave a gap in the schedule which required additional effort to fill. These disruptions were sometimes caused by the fact that short-term scheduling has more information available to it and therefore can uncover problems which cannot be seen at a higher level. Another, more important, factor contributing to this was the large degree of change to HST proposals after submission - for a variety of reasons most proposals where changed after long-range planning began. The first response to this problem was to "oversubscribe" the long-range plan, that is, allocate an excess of observations to each week. In practice an oversubscription level of ~100% was necessary to ensure well-filled weekly schedules, and this made it impossible to predict with reliability when an observation would occur and required a large amount of rescheduling. We have recently developed an alternate long-term strategy which solves this problem. The long-term plan allocated each scheduling unit to a range of weeks (called a plan window). This range provides for each week an implicit oversubscription to maintain short-term efficiency, yet there is a high degree of certainty that the observation will be executed within the plan window. Our initial studies indicate that with plan windows as small as 4 weeks over 95% of the observations are executed within the plan window.

SUMMARY

HST science operations introduced several novel concepts for astronomical observation, including distributed proposal preparation tools, abstraction of the scientific program from the specifics of the implementation, and fully interleaved scheduling. To support this, a number of advanced planning and scheduling systems were developed and have supported HST throughout four years of operations. Current major enhancements to these systems include making more tools available to proposers and re-engineering the systems to better support proposal changes. Several tools have been adapted to other space- and ground-based observatories.

The Space Telescope Science Institute is operated by the Association of Universities for Research in Astronomy for NASA.

REFERENCES

[1] Gerb, A. et. al. 1991. Transformation Reborn: A New Generation Expert System for HST Operations. In *Proceedings of the 1991 Goddard Conference*, 283-295. NASA CP 3110.

[2] Johnston, M.; and Miller, G. Spike: Intelligent Scheduling of HST Observations. in *Intelligent Scheduling*, ed. M. Zweben and M. Fox. San Francisco: Morgan-Kaufman.

[3] Jackson, R. et. al. 1988. The Proposal Entry Processor. In *Proceedings of the 1991 Goddard Conference*, 197-212. NASA CP 3009.

[4] Minton, S. et. al. 1992. Minimizing Conflicts: A Heuristic Repair Method for CSP Scheduling. *Artificial Intelligence* 58: 161-205.

N95- 23741

A Constraint-Logic Based Implementation of the "Coarse-Grained" Approach to Data Acquisition Scheduling of the International Ultraviolet Explorer Orbiting Observatory

Bruce McCollum
Science Programs, Computer Sciences Corporation
Code 684.9, Goddard Space Flight Center
Greenbelt, MD 20771
tel: (301) 286-7537 fax: (301) 286-7642
e-mail: mccollum@iuesoc.gsfc.nasa.gov

Mark Graves
Department of Cell Biology
Baylor College of Medicine
One Baylor Plaza, Houston, TX 77030
e-mail: mgraves@bcm.tmc.edu

KEY WORDS AND PHRASES

Artificial Intelligence, Constraint Logic Programming, IUE Astronomy Satellite, Satellite Scheduling

BACKGROUND

In 1994 there are six large, long-lived astronomy satellites in operation. Additional missions are planned by the U.S. and other countries by the end of the decade. The general problem of setting up a yearly schedule of science observations for an astronomy satellite is a challenge which will exist in various incarnations for the foreseeable future.

Every year, each orbiting observatory typically carries out observations for several dozen different science programs, collecting data on up to a few hundred different objects in the sky by the end of the year. The number of distinct observations ("exposures") carried out each day with each satellite varies from one to more than ten, depending on the particular satellite. For every satellite, an annual schedule for these hundreds of observations must be set up which obeys the physical and operational constraints of the satellite and the scientific constraints of the many different science programs. In general terms, the problem of science scheduling in a satellite mission is usually cast as attempting to find the best schedule out of an enormous number of possible schedules.

The International Ultraviolet Explorer (IUE) satellite observatory has been in operation continuously since 1978. It typically carries out several thousand observations per year for over a hundred different science projects. These observations, which can occur in one of four different data-taking modes, fall under several satellite-related constraints and many other constraints which derive from the science goals of the projects being undertaken.

One strategy which has made the scheduling problem tractable has been that of "coarse-graining" the time into discrete blocks of equal size (8 hours), each of which is devoted to a single science program, and each of which is sufficiently long for several observations to be carried out. We call it "coarse-graining" because the schedule is done at a "coarse" level which ignores fine structure, i.e., no attempt is made to plan the sequence of observations occurring within each time block. Planning science observations on a "fine" level, within each time block, is done by the guest investigator whose program has been allocated that time block.

Coarse-graining the schedule has several advantages. It reduces the number of time blocks composing a schedule from several thousand to 730. Because most time blocks can be scheduled independently, it permits rapid rearrangements of the schedule with a minimal effect on the overall schedule. It also gives guest investigators the freedom to make last-minute changes in their observations based on new results or new thinking, which can significantly enhance the quality of science; although important in science, such qualitative human judgement cannot readily be represented in any scheduling algorithm.

Another advantage is that coarse-graining increases the observatory's ability to make significant changes in the schedule on short notice with minimal impact to the rest of the schedule, because the time blocks are usually mutually independent. In a fine-grained schedule where a linear sequence of (e.g.) a thousand distinct observations must be planned, moving the time of one observation causes a change in the time of all the rest. This is due to the fact that the time required to obtain one data set of one target (the "exposure time", as in photography nomenclature) varies from seconds to hours. In a fine-grained schedule, if a short exposure time (small time slot) is replaced by a long exposure time (large time slot), then every observation for the rest of the schedule must be moved later in time if the schedule is to avoid "gaps". In a coarse-grained schedule, however, the schedule is divided into (fewer) time blocks having equal length which are usually independent and interchangable. Thus an observer can change the length of exposure or number of exposures within his or her time block without affecting the other time blocks in the schedule. Some ability to make schedule changes on short notice is necessary in most astronomy satellites because a number of important classes of astronomical objects (e.g. novas, supernovas, transient X-ray sources) appear suddenly and unpredictably and may fade rapidly so that data must be obtained quickly before the target is no longer detectable.

SOFTWARE SOLUTION

We have incorporated the IUE's coarse-grained approach in new software which examines the science needs of the observations and produces a limited set of alternative schedules which meet all of the instrument and science-related constraints. With this algorithm, the IUE can still be scheduled by a single person using a standard workstation, as it has been. We believe that this software could be adapted to a more complex mission while retaining the IUE's high flexibility and efficiency. This has the potential of improving the efficiency and scientific return of future satellite missions.

Our first step was to develop a representation for the constraints sufficient for scheduling relatively simple satellites and implement a constraint logic program which accesses the representation and discovers a set of coarse-grained schedules [2]. Our coarse-grained scheduler can find a collection of schedules which satisfy the overarching spacecraft, instrument, target, and scientific program constraints. A human scheduler can then choose from the set an optimal schedule which maximizes the quality of science, after consulting with guest investigators, if necessary, about priorities and trade-offs.

Data Structures for Domain Representation

Developing an appropriate representation is the first step in developing a system which is data driven or knowledge based. We set up a representation for the programs, targets, instruments, and instrument exposures and a representation for general constraints on them, including constraints on the spacecraft operation.

A schedule consists of a collection of investigator programs which are assigned a number of shifts. The investigator's shifts are scheduled into one of 730 shifts during in the year. The shifts are 8 hour blocks of time which define the level of granularity for the coarse-grained approach.

An investigator requests observations of certain targets. Each observation may consist of one exposure or a set of exposures, and each exposure has a specified instrument (data collection mode) and exposure time. Each target may be viewed only on certain days which depend upon its angle to the sun.

Our satellite scheduling language can represent constraints specific to particular science programs, such as various types of temporal constraints, target observation constraints, and instrument exposures. The constraints which must be represented generally fall into two categories: constraints inherent in the spacecraft and instrumentation which are always in effect, and constraints which reflect the scientific needs of the different programs.

In the case of the IUE, spacecraft/instrument constraints include, among other things:

1. There are four data-taking modes, which can be used only one at a time.
2. The spacecraft is restricted in the directions it can point relative to the Sun. This has the effect of restricting the time of year during which a particular object can be observed.
3. Only short (less than 1 hour) exposures can be taken during part of each day due to high background radiation. This period occurs at the same time each day.

Typical constraints which are due to the specific objectives of the various science programs being carried out include:

1. Each science program is annually allotted a fixed, exact amount of time in which it may use the instrument.
2. The choice of detectors, targets, and exposure times is specified by the guest investigator. This information is solicited at the beginning of the scheduling year.
3. For science reasons, some observations must include or avoid certain dates or ranges of dates.

Overview of the Algorithm

The scheduling algorithm is a constraint logic program which finds all valid schedules and presents them one at a time to the human scheduler. The human scheduler determines whether the presented schedule is sufficient or whether the program should attempt to find an alternative one. The input to the algorithm is a series of constraints which must be met to create a valid schedule. The output is a collection of valid schedules.

Searching the space of all possible schedules is not tractible. Constraint logic programming addresses this issue by restricting the search space. Constraints on the observation date restrict the schedules which are considered, thus reducing the search space. It is useful to reduce the search space even more using techniques such as priority scheduling, which is described in the next section. After the reduced search space is determined, a simple logic program with backtracking is used to create a list of possible schedules and present them to a human scheduler.

The input to the system is a collection of investigator programs. Each investigator's program consist of a collection of target observations. There are two main constraints on the day which an observation may be scheduled. They are the angle the target has in relation to the sun and the request of the investigator. These constraints are combined to create a constraint on the day of observation for the target. These observation constraints are combined into constraints on possible days to which the entire program can be assigned. The combining of constraints occurs by unifying them in a constraint logic program, which is explained in [2].

Sometimes there are problems with the input, such as inconsistencies. Investigator programs may be inconsistent in two ways: they may be inconsistent individually or as part of a collection. When a program is approved by the IUE review process for observation it is assigned a specific number of shifts, which may not be sufficient to observe all proposed targets. In addition, two or more programs may have conflicting constraints. In either case, the investigator(s) must decide which target observations are more important and inform the human scheduler which ones have higher priority. The algorithm can assist in the process by listing alternative schedules.

Even in consistent schedules, there are additional scheduling tasks which must be supported. In a program with many targets, the investigator's program will generally have to be split across multiple days to meet all the constraints, though this does not have to be handled solely by the algorithm. The constraints used in creating a complete schedule of all observations can also be used to generate options for each investigator's program. There are two kinds of listings that can be created. The first is a list of all possible scheduling days for a program. The second option is to use an existing schedule and list all the ways it could be changed to reschedule an investigator's program. This option is especially important for use when unexpected observation opportunities arise.

Implementation

Our algorithm is implemented in the constraint logic programming language LIFE, which is a fusion of object-oriented, functional, and logic programming paradigms developed at Digital Equipment Corporation [1]. We have selected LIFE for this project because it is especially well suited for handling constraints in a declarative manner.

Because there are usually several schedule changes a week, it is important that the scheduling algorithm have an efficient implementation and support incremental updates. One way to make scheduling more efficient is to prioritize the scheduling of observations based on how constrained the observation day is. All programs are still scheduled and the scheduling priority has no relation to any priorities the investigator may set within a program. The search is made faster by considering observations in order of the severity of constraints on possible observation days, with the most time-restricted observations placed in the schedule first. This prioritization is independent of the dates which are allowed. For example, if the constraints limit one observation to one of five days in the year, then that observation is scheduled before observations which can occur during one of 60 days. There can and will be conflicts which will require backtracking, but there will generally be fewer conflicts than with an arbitrary ordering. (This is the same strategy which has already been used successfully for the IUE.)

The technique used to implement the prioritization is priority scheduling. Priority scheduling is a technique from operating systems research where each "job", in this case an observation request, is placed on an ordered set of queues. All the jobs on the first queue are scheduled before any job on the next one, and the process repeats until all requests are scheduled. If a scheduling year contains 365 days, then the algorithm creates a priority queue with 365 levels. Each observation is placed in the level corresponding to the number of days during the year in which it could be scheduled. The observations are scheduled beginning with the first level and proceeding through all 365 levels. If a conflict occurs, then the algorithm backtracks to remove the conflict.

Other satellite scheduling programs exist which use constraints, but none make use of the coarse-grained approach. Some, such as SPIKE [3] use constraint satisfaction to create fine-grained schedules. However, constraint-logic programming has an advantage over constraint-satisfaction programming in that constraint logic programming provides a mechanism for solving constraints within the framework of a high-level programming language. Within constraint satisfaction, there are typically a limited collection of domains and operators which can be used. Within constraint logic programming, constraints can be placed on any variables which occur in a logic formula. We have chosen to use constraint logic programming because it is easier to develop more complex systems of constraints within it.

DISCUSSION

For the past fifteen years, the International Ultraviolet Explorer (IUE) astronomical satellite has been successfully scheduled by "coarse-graining" the time into large (8-hour) discrete blocks, each of which is devoted to a single science program, and each of which is sufficiently long for several sets of data to be acquired. This approach has worked well. The IUE has established a reputation for a high quantity of observations per year as well as high quality of science resulting from them. Coarse-graining greatly simplifies the scheduling problem by not seeking to find the optimum schedule of all possible schedules, but instead to develop a schedule which meets a minimum but adequate set of scientific and instrument constraints.

Our implementation in its present form would have to be modified to work for satellites other than the IUE. However, as discussed above, the coarse-grained approach has some advantages (flexibility, inclusion of scientific judgement) which would be desirable in most space observatories. Furthermore, most of the science and instrument constraints of the IUE are shared to some degree by most space observatories. We believe that adapting our IUE implementation to other satellites would in most cases be possible without too much difficulty. A modified form of our algorithm could be used to schedule a ground-based telescope by a single person using a standard workstation. Whether our approach would be the best implementation to schedule a particular future mission would require further study in the context of planning that mission.

References

[1] Ait-Kaci, Hassan, 1991. "An Overview of LIFE". In *Next generation information system technology: Proceedings of the First International East/West Data Base Workshop*, 42. Vol. 504 of *Lecture Notes in Computer Science*, eds. J.W. Schmidt and A.A. Stogny, Springer-Verlag.

[2] Graves, M. and McCollum. B. 1994. "Coarse-Grained Scheduling for Astronomy Satellites", 1. In *Knowledge-Based Artificial Intelligence Systems in Aerospace and Industry*, SPIE Technical Conf. 2244.

[3] Johnston, Mark D., 1990. "SPIKE: AI Scheduling for NASA's Hubble Space Telescope", 184–190. In *Proc. 6th Conference on AI Applications*, IEEE.

N95- 23742

Robust Telescope Scheduling

Keith Swanson
NASA

John Bresina
Recom Technologies

Mark Drummond
Recom Technologies

AI Research Branch, Mail Stop: 269-2
NASA Ames Research Center, Moffett Field, CA 94035-1000
E-mail: {swanson, bresina, drummond}@ptolemy.arc.nasa.gov
Phone: 415-604-6016, *Fax*: 415-604-3594

KEY WORDS AND PHRASES

Automatic telescopes, execution errors, Just-In-Case scheduling, robust execution, telescope scheduling.

ABSTRACT

This paper presents a technique for building robust telescope schedules that tend not to break. The technique is called Just-In-Case scheduling and it implements the common sense idea of being prepared for likely errors, just in case they should occur. The JIC algorithm analyzes a given schedule, determines where it is likely to break, reinvokes a scheduler to generate a contingent schedule for each highly probable break case, and produces a "multiply contingent" schedule. The technique was developed for an automatic telescope scheduling problem, and the paper presents empirical results showing that Just-In-Case scheduling performs extremely well for this problem.

INTRODUCTION

This paper presents and evaluates a technique for generating schedules that have robust execution behavior. The technique is called Just-In-Case scheduling, or JIC, and it implements the common sense idea of being prepared for likely execution errors, just in case they should occur. JIC handles schedule execution errors that are due to the presence of actions with uncertain durations. JIC was developed as part of a larger telescope management and scheduling project. This section outlines only key aspects of the problem; more details are available elsewhere [4; 6].

In this application domain, the telescopes are land-based and fully automatic; a telescope control computer opens the observatory at twilight and collects data through the night without human assistance (see Genet and Hayes [8] for details). We are implementing an overall automated management system [6] to enable participating astronomers to submit observation requests and obtain results from a remotely located telescope. This interaction occurs *via* electronic communication networks, without the necessity of human intervention. To ensure the telescope load is balanced over weeks or months, the system will also include a sophisticated long-term loader [2]. The long-term loader will assign observation requests to specific nights. Each night, the assigned observations are given to a scheduler to determine the specific times during the night they are to be executed. Producing ***robust*** nightly schedules is the role of JIC.

An observation request specifies both hard constraints and soft preferences. The most important hard constraint is the ***observing window.*** Each observation request, or action, can begin execution only in a specific interval of time within a night; this interval is defined by the astronomer who submitted the request. In the remainder of this paper we refer to each observation request as an ***action.***

The scheduling problem is to synthesize a schedule that satisfies all hard constraints and achieves a good score according to an objective function based on the soft preferences. A schedule is a sequence of actions, each with an ***enablement interval*** assigned by the scheduler. The assigned enablement interval of each action is a subinterval of the action's (astronomer-provided) observing window. A scheduler assigns the enablement intervals to further restrict when the actions will begin execution. This paper does not address the problem of generating a basic schedule (this is discussed in [7]) – we assume the existence of a scheduler that, given a set of requests and an objective function, produces a feasible observing schedule with a reasonable score. (Our current scheduler produces reasonable schedules in less than one minute.)

In this domain, a typical action has a duration of several minutes with action duration uncertainty occurring due to mechanical slop in the telescope drive train, software timing inaccuracies, and star centering. The amount of time it takes to center a star depends on how accurately the telescope is pointed when it starts the search and how clear the sky is while the centering is going on. Hence, it is impossible to accurately predict the duration of an observing action. All we can do is gather data from actual executions and then calculate the mean and standard deviation of each action's duration. Observation actions are typically executed many times over a period of weeks or months, so such statistics are easily available.

The telescope used in this application is fully automatic and runs unattended; thus, unlike many scheduling domains where printing a schedule is the final goal, our system must be able to automatically execute a schedule. Schedule execution proceeds by executing each action in the scheduled sequence. After an action finishes execution, if the current time is outside of the next action's (scheduler-assigned) enablement interval, then the schedule breaks and execution halts.

Schedule breakage is the central problem. The predicted start time of an action in a schedule is based on the sum of the estimated durations of the actions that precede it. Hence, the further into the future an action occurs in the schedule, the greater the uncertainty surrounding its actual start time. Given the way that uncertainty grows, it is possible for a schedule to break due solely to accumulated duration prediction errors.

There is a simple solution to the problem of duration prediction errors: make the start time of each action equal to a worst case estimate of the previous action's finish time and introduce a busy-wait in case the previous action finishes early. Unfortunately, introducing such busy-waits will waste observing time. Our goal is to avoid schedule breaks without sacrificing schedule quality.

Schedules can also fail for reasons other than duration prediction uncertainty. Clouds or wind can make star acquisition impossible, resulting in unavoidable schedule breaks. In our system, when the schedule breaks, the telescope invokes the scheduler to generate a new one for the current situation. Hence, while weather can cause a break in schedule execution, the system is robust enough to dynamically reschedule and try again. Dynamic rescheduling could also be used, in place of JIC, to handle breaks due to duration prediction errors. However, the problem with dynamic rescheduling is that it wastes valuable observing time whenever the telescope controller is waiting for a schedule. There is limited observing time available during the night, and we do not want to waste it!

JIC proactively manages duration uncertainty by identifying high probability schedule breaks and, for each one, generating an alternative schedule just in case the break occurs during execution. This proactive management can use off-line time during the day to compute and store alternative schedules in order to reduce on-line rescheduling time during the night. JIC produces a "multiply contingent" schedule that specifies what actions the telescope controller should take, conditioned by the current situation. Thus, if accumulated duration prediction errors force the telescope into a situation for which the nominal (*i.e.*, the initial) schedule is inapplicable, then an appropriate contingent schedule (if available) is automatically selected and execution continues. If an appropriate schedule is not available, the system resorts to dynamic rescheduling.

JUST-IN-CASE SCHEDULING

In overview, the JIC algorithm accepts a schedule as input and robustifies it as follows. First, using a model of how action durations can vary, the temporal uncertainty at each step in the schedule is estimated. Second, the most probable break due to this uncertainty is determined. Third, the break point is "split" into two hypothetical cases: one in which the schedule breaks and one in which it does not. Fourth, the scheduler is invoked on a new scheduling subproblem to produce an alternative schedule for the break case. Fifth, this alternative schedule is integrated with the initial schedule producing an updated multiply contingent schedule. This completes consideration of one break case; if there is more time before schedule execution begins, then the JIC process can be repeated with the current multiply contingent schedule as the new input.

Each action A_i has a duration mean μ_i and standard deviation σ_i. One of the preconditions of each action is the interval of time during which it can begin execution; let P_i be this ***precondition interval*** for A_i. (The precondition interval for observation requests is provided by an astronomer.)

A ***schedule*** is a sequence of actions, where each action is associated with an ***enablement interval***, E_i, assigned by the scheduler: $(A_0, E_0); \ldots; (A_n, E_n)$, such that for $i = 0, \ldots, n$, $E_i \subseteq P_i$. During schedule execution, as soon as action A_{i-1} is finished executing, action A_i is selected for enablement testing; A_i is enabled if the current time is within E_i. If A_i is enabled, then it is immediately executed; else, the schedule breaks.

A ***multiply contingent schedule*** can be thought of as a set of alternative schedules; to save space, our implementation uses a tree to represent this set of schedules. Let $\beta(i)$ be defined such that $A_{\beta(i)}$ is the predecessor of A_i in the schedule, if one exists. For simplicity, we assume that A_0 is the unique first action in a multiply contingent schedule.

Using a duration uncertainty model discussed below, JIC estimates the temporal uncertainty at each step in the schedule by starting at the beginning of the schedule and propagating uncertainty forward. This process involves estimating the time at which each action in the schedule will start and finish executing. The ***start interval***, S_i, is the set of possible execution start times for action A_i. Similarly, the ***finish interval***, F_i, is the set of possible execution finish times for action A_i. Let S_0 denote the interval during which schedule execution can start. For our telescope application, schedule execution always starts exactly at twilight; hence, S_0 is the degenerate interval [twilight, twilight].

A_i cannot start executing at a time outside its enablement window. Hence, if $A_{\beta(i)}$ finishes executing at a time outside of E_i, then either an action in an alternative contingent schedule will be executed or the schedule will break. Thus, S_i is computed to be $F_{\beta(i)} \cap E_i$.

Given that A_i's start interval, Si, is $[t_1, t_2]$, its finish interval, F_i, is computed to be $[t_1 + \mu_i - \sigma_i, t_2 + \mu_i + \sigma_i]$. The current duration uncertainty model simply uses one standard deviation of the mean when computing each finish interval – this has worked well in practice, as shown by the empirical results in the next section.

The break probability of an action is a function of the enablement probability of that action and of all preceding actions. Let $p(\text{enable}(A_i))$ be the *enablement probability* for action A_i; that is, the probability that A_i will be enabled when selected. It is computed to be the proportion of the previous action's finish interval during which A_i is enabled.

$$p(\text{enable}(A_i)) = \frac{|F_{\beta(i)} \cap E_i|}{|F_{\beta(i)}|}$$

For simplicity, this computation is based on the erroneous assumption that all of an action's possible finish times are equi-probable (*i.e.*, that $F_{\beta(i)}$ has a uniform probability distribution) and, hence, is only an estimate of the true enablement probability.

Let $p(\text{select}(A_i))$ be the *selection probability* for action A_i; that is, the probability that A_i will be selected for enablement testing. An action will be selected if the preceding action was both selected and enabled; the schedule's first action will always be selected.

$$\begin{aligned} \text{For } i = 0\text{:} \quad & p(\text{select}(A_i)) = 1.0. \\ \text{For } i > 0\text{:} \quad & p(\text{select}(A_i)) = p(\text{select}(A_{\beta(i)})) \times \\ & \qquad p(\text{enable}(A_{\beta(i)})). \end{aligned}$$

Let $p(\text{break}(A_i))$ be the *break probability* for action A_i; that is, the probability that the schedule will break at A_i when it is selected for enablement testing.

$$p(\text{break}(A_i)) = p(\text{select}(A_i)) \times [1 - p(\text{enable}(A_i))].$$

Note that the computation of break probabilities is similar to the computation of the conditional probabilities characterized by a Markov chain.

After determining the action with the highest break probability, JIC "splits" the associated uncertainty time interval into two subintervals. One subinterval will be the intersection of the finish interval $F_{\beta(i)}$ with the enablement interval E_i. The remaining subinterval (not part of the intersection) is split off as a break case and a new scheduling subproblem is formed using a point in the subinterval as the start time. JIC then invokes the scheduler on this subproblem and incorporates the returned alternative schedule into the original schedule.

EMPIRICAL EVALUATION

To evaluate the performance of JIC we performed an experiment using real telescope scheduling data. (Additional experimental results and algorithm details are available elsewhere [3; 5].) The observation actions were provided by Greg Henry of Tennessee State University [9; 11]. The scheduler used in this experiment deterministically hill-climbs on a domain-specific heuristic [1]. The experiment required collecting data from thousands of schedule executions; since this is impractical on a real telescope, we developed a simulator of the telescope controller's schedule executor. The simulator computes an action's execution duration by using a random variable with a normal (Gaussian) probability distribution whose mean and standard deviation are exactly those characterized by statistics obtained from a number of nights of actual execution on a telescope at the Fairborn Observatory (Mt. Hopkins, Arizona).

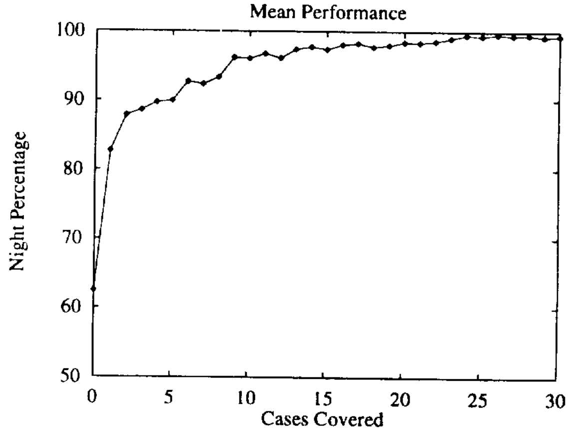


Figure 1: Mean performance, measured as night percentage, *vs.* cases covered.

The experimental question is: given real telescope scheduling data, can JIC provide a useful increase in schedule robustness within a reasonable number of contingent cases? To answer this question we varied the number of break cases considered and measured how far into the night a multiply contingent schedule would execute without dynamic rescheduling. The experimental procedure is as follows.

First, the scheduler is used to find a single nominal schedule. This schedule is executed 1000 times in the simulator; for each execution run we note the percentage of the night that the schedule executes before halting, either due to a break or schedule completion. Next, we allow JIC to find and fix what it deems to be the next most probable break case. We then run the augmented schedule through the execution simulator (again, 1000 times). In this manner, we allow JIC to cover up to thirty break cases.

Figure 1 illustrates the resulting performance. The independent variable is the number of break cases covered by JIC. The dependent variable is the percentage of the night that the schedule executes before halting, averaged over 1000 runs. It clearly shows that the mean percentage of the night executed increases with the number of cases considered by JIC. The performance increase is most dramatic early on, as we had hoped. After only ten cases, the schedule executes, on average, through 96% of the night. Although not shown, experimental results also indicate that schedule size (measured as the total number of actions contained in the multiply contingent schedule) increases linearly with the number of cases, as one might expect.

CONCLUSION

This paper has presented an algorithm for Just-In-Case scheduling. Using almost any scheduler and simple statistical models of duration uncertainty, the algorithm proactively makes a nominal schedule more robust. Despite some rather egregious modeling assumptions, the algorithm works extremely well for a real telescope scheduling problem. Traditional intuitions surrounding the management of uncertain action outcomes suggest the inevitability of large search spaces and intractable reasoning. Using a "splitting" technique, our algorithm makes action outcome distinctions only when necessary (see Hanks [10] for background to this idea). Further, most of the likely schedule breaks are covered in a few iterations of JIC.

While JIC works extremely well for our particular telescope scheduling problem, it will not necessarily fare so well on all domains. We have analyzed the nature of the schedule breaks in our domain in order to characterize the general conditions under which JIC achieves useful robustness increments in a few iterations. The results are suggestive, but not yet mathematically precise. Essentially, JIC appears to work well when the following three conditions hold.

First, there must be room for improvement. If the prior probability of successful execution of the schedule is close to 1.0, there is not much JIC can add. Second, there must be a small number of schedule breaks responsible for most of the total break probability mass. If this is so, then each break case covered by JIC stands a good chance of increasing the probability of executing the entire schedule. Third, each contingent schedule found must be no worse in its break characteristics than the nominal schedule. In some sense, this is simply a recursive application of the first two conditions; it requires that each contingent schedule be as easy to robustify as the initial one.

Finally, we recognize that a number of interesting issues remain outstanding regarding the applicability of JIC to other domains and how JIC compares to, or might integrate with, other existing scheduling techniques. This is an excellent area for further work.

ACKNOWLEDGEMENTS

Significant thanks to Will Edgington for helping make the telescope management and scheduling software a reality.

REFERENCES

[1] Boyd, L., Epand, D., Bresina, J., Drummond, M., Swanson, K., Crawford, D., Genet, D., Genet, R., Henry, G., McCook, G., Neely, W., Schmidtke, P., Smith, D., and Trublood, M. 1993. Automatic Telescope Instruction Set 1993. *International Amateur Professional Photoelectric Photometry Communications*, No. 52.

[2] Bresina, J. 1994. Telescope Loading: A Problem Reduction Approach. In *Proc. of i-SAIRAS 94*. JPL, Pasadena, CA.

[3] Bresina, J., Drummond, M., Swanson, K. 1994. Managing Action Duration Uncertainty with Just-In-Case Scheduling. *Proceedings of the AAAI Spring Symposium on Decision-Theoretic Planning*. Stanford, CA. March, 1994.

[4] Bresina, J., Drummond, M., Swanson, K., and Edgington, W. 1994. Automated Management and Scheduling of Remote Automatic Telescopes. *Astronomy for the Earth and Moon*, the proceedings of the 103rd Annual Meeting of the Astronomical Society of the Pacific, D. Pyper Smith (ed.).

[5] Drummond, M., Bresina, J., and Swanson, K. 1994. Just-In-Case Scheduling. In *Proc. of AAAI-94*.

[6] Drummond, M., Bresina, J., Swanson, K., and Edgington, W. 1994. The Associate Principal Astronomer Telescope Operations Model. In *Proc. of i-SAIRAS 94*. JPL, Pasadena, CA.

[7] Drummond, M., Swanson, K., and Bresina, J. (*in press*). Robust Scheduling and Execution for Automatic Telescopes. In *Intelligent Scheduling*, M. Fox & M. Zweben (eds). Morgan-Kaufmann.

[8] Genet, R.M., and Hayes, D.S. 1989. *Robotic Observatories: A Handbook of Remote-Access Personal-Computer Astronomy*. AutoScope Corporation, Mesa, AZ.

[9] Hall, D. S. and Henry, G. W. 1992. Performance Evaluation of Two Automatic Telescopes after Eight Years. *Automated Telescopes for Photometry and Imaging*, S. J. Adelman, R. J. Dukes, and C. J. Adelman (eds.) (San Francisco: Astronomical Society of the Pacific), p. 13.

[10] Hanks, S. 1990. Practical Temporal Projection. In *Proc. of AAAI-90*. pp. 158 – 163.

[11] Henry, G. W. and Hall, D. S. (*in press*) The Quest for Precision Robotic Photometry. *International Amateur Professional Photoelectric Photometry (IAPPP) Communications* 55.

N95- 23743

The Associate Principal Astronomer Telescope Operations Model

Mark Drummond* **John Bresina*** **Keith Swanson†** **Will Edgington*** **Greg Henry‡**

*Recom Technologies
†AI Research Branch, Mail Stop: 269-2
NASA Ames Research Center
Moffett Field, CA 94035-1000
Phone: 415.604.4710, *Fax:* 415.604.3594
E-mail: xfr@ptolemy.arc.nasa.gov

‡Center of Excellence in Information Systems
Tennessee State University
330 Tenth Avenue North
Nashville, TN 37203-3401
Phone: 615.251.1225, *Fax:* 615.251.1117
E-mail: henry@schwab.tnstate.edu

KEYWORDS AND PHRASES

Associate principal astronomer, automatic telescope, automatic telescope instruction set, principal astronomer, telescope operations.

ABSTRACT

This paper outlines a new telescope operations model that is intended to achieve low operating costs with high operating efficiency and high scientific productivity. The model is based on the existing *Principal Astronomer* approach used in conjunction with ATIS, a language for commanding remotely located automatic telescopes. This paper introduces the notion of an *Associate Principal Astronomer*, or APA. At the heart of the APA is automatic observation loading and scheduling software, and it is this software that is expected to help achieve efficient and productive telescope operations. The purpose of the APA system is to make it possible for astronomers to submit observation requests to and obtain resulting data from remote automatic telescopes, via the Internet, in a highly-automated way that minimizes human interaction with the system and maximizes the scientific return from observing time.

BACKGROUND

Research quality telescopes located at prime observing sites have always been a scarce resource, and astronomers have had to work with limited access to these telescopes. Typically, observing time is allocated to an individual astronomer a few times per year in short contiguous blocks of a few nights each. Furthermore, the astronomer has needed to be physically present at the telescope in order to operate his instrumentation for data acquisition. Limited access, block allocation, and local operation have restricted both the amount of data that can be gathered and the type of observational campaigns that can be accomplished.

More recently, sophisticated network and communication technologies have enabled a number of new approaches where astronomers may participate in an observation program from a remote location. These approaches range from remote verbal communications with the on-site telescope operations staff to actual remote control of a telescope with real time video feedback [4]. Such remote observations provide flexibility by allowing the observer to be physically distant from the telescope yet remain in direct control. However, even in this remote observing paradigm, the astronomer must still be involved during the execution of the observing program, and human presence at the observatory is often still required.

Fully automatic telescopes represent an extension to the remote observing paradigm, allowing an astronomer to be removed from the telescope both temporally as well as spatially. For example, Fairborn Observatory (Mt. Hopkins, Arizona) and AutoScope Corporation (Fort Collins, Colorado) have designed and built software and hardware systems for the control of modest-aperture telescopes equipped with photoelectric photometers to measure stellar brightness. These systems make it possible for a remotely located telescope to operate unattended for significant periods (up to a number of months). These telescopes execute commands provided by an astronomer in such a way that the astronomer is not required to participate in the execution of the observing program. It is in this sense that these telescopes are *fully automatic*.

While the majority of existing ground-based automatic telescopes are used for aperture photometry, automation support for spectroscopy and imaging has been increasing (primarily due to the efforts of R. Kent Honeycutt and Don Epand [3]). Genet and Hayes [6] describe automatic photoelectric telescopes in some detail.

For the sort of telescope we are considering, the language used to define observation requests is the Automatic Telescope Instruction Set, or ATIS [3]. In ATIS, a *group* is the primitive unit to be scheduled and executed. A group is a sequence of telescope commands and instrument commands defined by an astronomer to accomplish the observation of an object of interest. A group contains commands to move the telescope, to control the filters, and to gather data in a defined sequence. In the initial version, ATIS89, the only instruments accommodated were photometers, but the most recent version, ATIS93, also includes commands to obtain CCD camera images.

In addition to specifying the syntax and semantics for observation requests and results, the ATIS standard provides a set of *group selection rules* that are used to determine the execution order of groups during the night. The group selection rules provided by ATIS essentially implement a first-to-set-in-the-west policy: at any given point in time the telescope observes the star that will set next. It is possible to improve upon this default group selection policy by using more sophisticated scheduling techniques. Specifically, it is possible to improve the quality of the data by more precisely scheduling groups so that observations are taken at lower airmass (on average), and so that observations are obtained at astrophysically interesting times. Additionally, for a multi-user telescope, better scheduling can result in a fairer allocation of telescope time to requesting astronomers.

We were invited to be part of the International Astronomical Union ATIS93 standardization committee to assist with ATIS extensions in support of advanced scheduling. Along with other committee members, we designed a new group selection advice statement. This new statement is used to override the default ATIS group selection rules. The committee also agreed on a mechanism for communication with a telescope controller in terms of incremental ATIS93 partial input and partial output files. Together, these new features make it possible to implement a non-native (*i.e.*, external) scheduler that can effectively drive a telescope's controller to better serve the scientific objectives of participating astronomers.

Our new approach to the automatic management of remotely located telescopes is based on ATIS93. At the heart of our approach is automatic observation loading and scheduling software, and it is this software that is expected to help increase science quality and telescope productivity. Our goals are to provide software tools to assist managers of multi-user automatic telescopes and to make it possible for participating astronomers to have their observation requests scheduled on and their resulting data returned from remotely located telescopes, via the Internet, without the necessity of daily human intervention.

THE CURRENT APPROACH

Before we explain how we intend to improve telescope management and use, we need to briefly explain the current manner in which automatic ATIS-compatible telescopes are managed. This is illustrated in the left half of Figure 1 and briefly described by the following scenario.

First, an astronomer forms a set of groups consistent with the scientific goals of his or her observation campaign. For any given automatic telescope, there is a single *Principal Astronomer* or PA. The PA manages the set of requests that are loaded onto the telescope. Thus, once an astronomer has assembled a set of ATIS groups, these are sent to the appropriate PA, typically via e-mail, Internet FTP, or on floppy discs in the postal service.

The PA collects together the sets of requests from participating astronomers and attempts to ensure that the total set of groups is desirable – that the telescope load makes good utilization of observing time and is fair to all participating astronomers, that there are appropriate groups for quality control and data reduction, *etc.* Then the complete set of groups is sent to the computer controlling the telescope. Communication between the PA and the telescope controller is typically carried out using personal computers connected via the Internet or modems and phone lines. The important aspect of the communication is that the PA can be located anywhere on the planet (in principle) and need only have access to an appropriate communication link.

The telescope controller uses its built-in ATIS group selection rules to implement a form of heuristic dispatch scheduling. At any point in time, the rules recommend a single group to execute next. The groups are executed by the telescope controller for some number of nights (often months); eventually, the PA requests from the controller the results that have been collected thus far. The collected data are returned to the PA as a results file specified within the ATIS language. The results include the raw data obtained from the observations, as well as a chronological record of the groups that were executed and relevant observing parameters to help with data reduction. The PA edits the results file and sends each astronomer the pieces corresponding to his or her requested observations (again typically via e-mail, Internet FTP, or on floppy discs). In some cases the PA provides a data-reduction service, returning reduced results, not simply raw data.

THE APA MODEL

The goal of our project is to provide automation support for all aspects of ATIS-compatible telescope management. Our focus is on providing software tools to help a PA who represents a community of participating astronomers; however, the increased automation also improves the way in which the astronomers interact with a PA. The right half of Figure 1 and the following scenario illustrate a new way of doing business with ATIS-based automatic telescopes that we are in the process of making possible. More details on the APA operations model are available in Bresina *et al.* [1].

From an Astronomer's Perspective

An astronomer creates an ATIS93 observation request file and sends it via electronic mail to the PA's computer. Let us refer to this computer as the *Associate Principal Astronomer*, or APA. The mailed file is automatically received and parsed to check for syntax errors. If the file adheres to the ATIS93 specification, then the APA e-mails a message back to

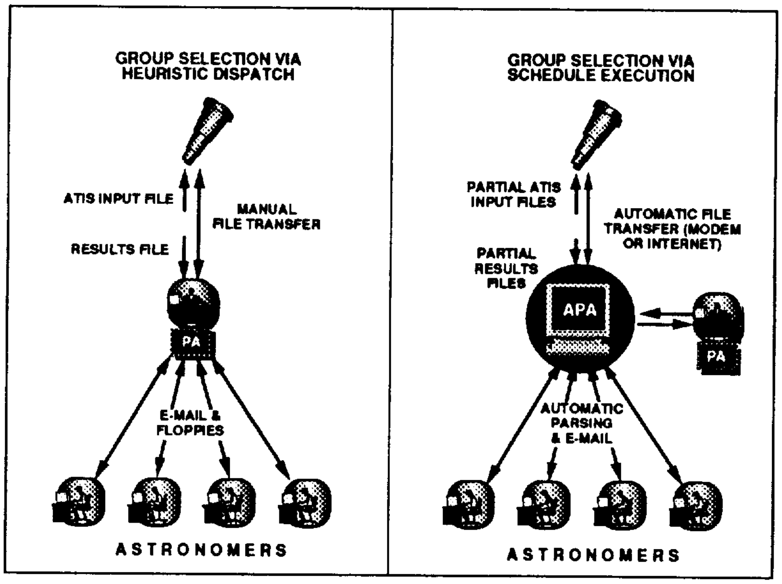


Figure 1: Operation of ATIS-compatible telescopes without (*left*) and with (*right*) the APA.

the astronomer acknowledging successful receipt of the request file; otherwise, a message is e-mailed back identifying the syntactic errors in the astronomer's file. At the end of each observing night, the APA e-mails the astronomer the results of those observation requests that were serviced that night, along with the results necessary for data reduction and data quality assessment.

From the PA's Perspective

The APA divides the overall problem of group scheduling into two subproblems: first, it assigns a group to execute on a given set of nights; second, for any group that has been selected for execution *tonight*, it assigns that group specific times through the night at which to execute. The first process is called *loading*, and its temporal scope covers many months. The second process is called *night scheduling*, and it is concerned with the seconds, minutes, and hours within a given night. After loading and night scheduling, a new combined ATIS93 file is automatically assembled by the APA. The PA can check how the controller will handle this new request file by displaying a prediction of telescope behavior for the night based on the best schedule found by the night scheduler (*i.e.*, what observations are likely to be made if the weather is ideal). If the PA is not satisfied with the prediction, then the manner in which the APA loads and schedules the observations can be modified. The next morning, the results of the night's observations are already stored at the APA. If the PA wants to assess the quality of the night's observation schedule and results, the actual telescope behavior can be displayed. Once the PA has tuned the APA to consistently produce high quality schedules, the APA takes care of routine observation loading and scheduling with only occasional supervision. A more complete description of the loader is given by Bresina [2], and a description of one of the techniques used in the night scheduler is given by Swanson, Bresina, & Drummond [9].

From the Telescope Controller's Perspective

Just before nightfall, the ATIS93 input file is automatically transferred to the telescope controller along with the observation schedule. The controller executes the schedule and, at the end of the night, transfers the ATIS93 output file back to the APA. This is the minimum amount of interaction between the telescope controller and the APA; however, the ATIS93 specification also allows for partial input and partial output files to be transmitted during the night. The partial output files enable the telescope behavior and status to be monitored during the night – either by a person (for example, to check the status of the telescope mechanics and optics) or automatically by the APA. The partial input files enable the APA to transmit new schedules and new groups during the night when necessary. For example, the APA can dynamically reschedule due to a change in the quality of observing conditions or due to an urgent observation request received during the night.

DISCUSSION

The overall goal of our project is to provide automation support for the management and use of remotely located, automatic telescopes. So far, we have focused on building the core of an Associate Principal Astronomer, or APA. This core consists of an automatic group loading and scheduling mechanism, together with a means for automatic schedule execution and dynamic rescheduling. While this core provides important functionality, there are many aspects of the PA's job that it does not address. In collaboration with other astronomers, we are currently expanding the set of functions offered by the APA to include automatic handling of ATIS request files, preliminary quick-look data reduction, and quality control measures. Experience gained with simulation tests and preliminary tests on an automatic telescope have been encouraging.

It is clear that the ATIS model is not the only one for automatic telescope management. Others have built APA-like systems [8]. The primary advantage of the APA is that it uses advanced scheduling techniques and operates with any telescope that adheres to the ATIS93 standard. Of course, NASA has a number of orbital telescopes that are operated remotely. The Hubble Space Telescope (HST), for instance, is operated in a way that is somewhat similar to our APA model. However, there is a significant amount of human infrastructure associated with the management of HST. Such infrastructure is expensive, and it cannot be replicated for every single telescope that is to be run automatically. Clearly, the human infrastructure surrounding HST performs useful tasks that our APA model ignores: for instance, helping users formulate their telescope requests and helping users make sense of the data they obtain. Our APA model leaves all such tasks firmly in the hands of telescope users (and their scientific community).

Our APA operations model requires one workstation (or a high-end personal computer), one experienced astronomer to act as the telescope PA, and one Principal Engineer / technician (PE) to fix the telescope and observatory control systems when things go wrong. A number of telescopes can be managed by a single APA, PA, and PE team.

One of us (GWH) has been working as a PA for a number of years with automatic telescopes. Together with Lou Boyd (of Fairborn Observatory) acting as PE, several telescopes have been operated automatically on Mt. Hopkins in southern Arizona to accomplish various scientific programs. The efficiency of operations for these telescopes has been estimated to achieve a dollar-cost-per-observation that is 30 to 40 times cheaper than previously possible using traditional manual telescope operations [7]. There has also been an enormous increase in observational throughput: the combined *yearly* output of the automatic telescopes managed by GWH would require a *lifetime* of effort to obtain by previous manual methods of operation.

To date, each of these automatic telescopes has been dedicated to a specific, long-term observing program. Thus, the operating schedule for each telescope has been extensively tuned by the PA (and sole user) to achieve acceptable performance. However, even small changes to the observing program make it very difficult to optimize the loading and scheduling. For multi-user telescopes, such extensive manual tuning is infeasible. In this context, our goal is to simplify and optimize the operation of single-user automatic telescopes and then to extend this simplified management structure to multi-user telescopes.

ACKNOWLEDGMENTS

Thanks to Bill Borucki, Lou Boyd, David Genet, Russell Genet, Butler Hine, George McCook, Ann Patterson-Hine, and Boris Rabin. Thanks especially to Ellen Drascher, who designed and implemented the APA's graphical user interface.

REFERENCES

[1] Bresina, J., Drummond, M., Swanson, K., and Edgington, W. 1994. Automated Management and Scheduling of Remote Automatic Telescopes. In *Astronomy from the Earth and Moon*, the proceedings of the 103rd Annual Meeting of the Astronomical Society of the Pacific, D.P. Smith (ed).

[2] Bresina, J. 1994. Telescope loading: a problem reduction approach. In *Proceedings of the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space*. Jet Propulsion Labs. Pasadena, CA.

[3] Boyd, L., D. Epand, D., Bresina, J., Drummond, M., Swanson, K., Crawford,D. Genet, D., Genet, R., Henry, G., McCook, G., Neely, W., Schmidtke, P., Smith, D., and Trublood, M. 1993. Automatic Telescope Instruction Set 1993. *International Amateur Professional Photoelectric Photometry (IAPPP) Communications*, No. 52, T. Oswalt (ed).

[4] Emerson, D.T., and Clowes, R.G. (eds.) 1993. *Observing at a Distance* (Proceedings of a Workshop on Remote Observing). World Scientific Publishing, River Edge, NJ.

[5] Genet, D. 1994. AutoScope Control System Reference Manual (Software Version 2.0). AutoScope Corporation, Ft. Collins, CO.

[6] Genet, R.M., and Hayes, D.S. 1989. *Robotic Observatories: A Handbook of Remote-Access Personal-Computer Astronomy*. AutoScope Corporation, Ft. Collins, CO.

[7] Henry, G.W. *forthcoming*. The Fairborn/TSU automatic telescope operations model. In *Proceedings of the symposium on Robotic Telescopes*, part of the 106th meeting of the Astronomical Society of the Pacific. Flagstaff, AZ.

[8] Richmond, M., Treffers, R., & Filippenko, A. 1993. The Berkeley Automatic Imaging Telescope. *Publications of the Astronomical Society of the Pacific*. No. 105 (October).

[9] Swanson, K., Bresina, J. and Drummond, M. 1994. Robust Telescope Scheduling. In *Proceedings of the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space*. Jet Propulsion Labs. Pasadena, CA.

Session PS-DS

Planning and Scheduling Workshop: Decision Support Aspects

N95- 23744

DTS: Building Custom, Intelligent Schedulers

Othar Hansson and Andrew Mayer
Heuristicrats Research, Inc.
1678 Shattuck Avenue, Suite 310
Berkeley, CA 94709-1631
(510) 845-5810, Fax: (510) 845-4405
{othar, mayer}@heuristicrat.com

KEY WORDS AND PHRASES

Decision theory, heuristic search, optimization, scheduling, user interface.

ABSTRACT

DTS is a *decision-theoretic scheduler*, built on top of a flexible toolkit—this paper focuses on how the toolkit might be reused in future NASA mission schedulers. The toolkit includes a user-customizable scheduling interface, and a "Just-For-You" optimization engine.

The customizable interface is built on two metaphors: objects and dynamic graphs. Objects help to structure problem specifications and related data, while dynamic graphs simplify the specification of graphical schedule editors (such as Gantt charts). The interface can be used with any "back-end" scheduler, through dynamically-loaded code, interprocess communication, or a shared database.

The "Just-For-You" optimization engine includes user-specific utility functions, automatically compiled heuristic evaluations, and a postprocessing facility for enforcing scheduling policies. The optimization engine is based on BPS, the Bayesian Problem-Solver [1,2], which introduced a similar approach to solving single-agent and adversarial graph search problems.

DTS SYSTEM OVERVIEW

The Decision-Theoretic Scheduler, DTS, is designed to support scheduling of over-subscribed, long-running projects. DTS is literally implemented as a program in a specialized language for the design of scheduling and optimization systems. This DTS Customization Language (DCL) is implemented on top of the public-domain TCL/Tk system [3].

DTS has been designed for science-planning on NASA missions. We are preparing to deploy the system as one component of a cost-reduction program within the Extreme Ultraviolet Explorer mission of the Center for Extreme Ultraviolet Astrophysics at the University of California, Berkeley [4].

We have explicitly designed DTS to be customizable by users, and thus transferrable to other missions. An easily customized scheduling system can reduce costs by eliminating the mission-specific paperwork and "workarounds" that result when a system does not address a scheduling scenario completely.

To reduce mission costs further, we have designed DTS so that such extensions can be made quickly and without corrupting existing code or functionality. For example, the current DTS interface provides much of the functionality of commercial project scheduling tools, but is implemented in under 7000 lines of DCL code. User modifications—such as an import "filter" for a pre-existing file format, or a specialized report writer—typically require only a few dozen lines of DCL code. Because DCL code is interpreted, programming errors are safely trapped.

Behind the scenes, the DTS "back-end" contains a sophisticated constraint-satisfaction search engine for use in automated scheduling. The use of decision theory permits user preferences and requirements to be modeled in a mathematically coherent way. The result is that DTS can typically find near-optimal solutions to the user's actual problem, with optimality measured in the user's terms. Many existing scheduling techniques restrict both the definition of optimality and the representation of the problem: the user is forced to use a system that provides a quasi-optimal solution to an approximation of the problem.

Our research goal in the DTS back-end has been to provide a rich representation for problems and preferences, and still find near-optimal solutions through the use of compilation, learning and decision-theoretic search.

In this paper, we describe customization in both the front-end and back-end, and then conclude with a description of future plans for applying DTS to NASA missions.

USER INTERFACE CUSTOMIZATION

The DTS interface uses objects and dynamic graphs to support customization.

All data in the system is represented within an object hierarchy. The hierarchy includes Task objects, Constraint objects, etc., as you would expect. These basic objects can be subclassed, or specialized, for the needs of an individual application: in the NASA version of DTS, an Observation object represents each Task that is an astronomical observation.

The system also includes "management" information objects such as (astronomical) Targets, (scientific) Proposals, and Principal Investigators. This information is linked to "problem" information such as tasks by the use of cross-reference attributes. For example, each Observation has an attribute named Target that is a cross-reference.

The DTS interface is centered on an object browser (Figure 2). Customization begins by defining a new object class, or redefining an existing object class. Each object class has an associated form, used to display and edit object instances in the browser. A simple default form is inferred from the "type" of each attribute (String, Date, etc.).

More complex forms require the use of DCL code. Figure 2 shows the form for a TemporalConstraint instance. This is the most complicated form in the system, but it requires only 40 lines to produce a specialized display

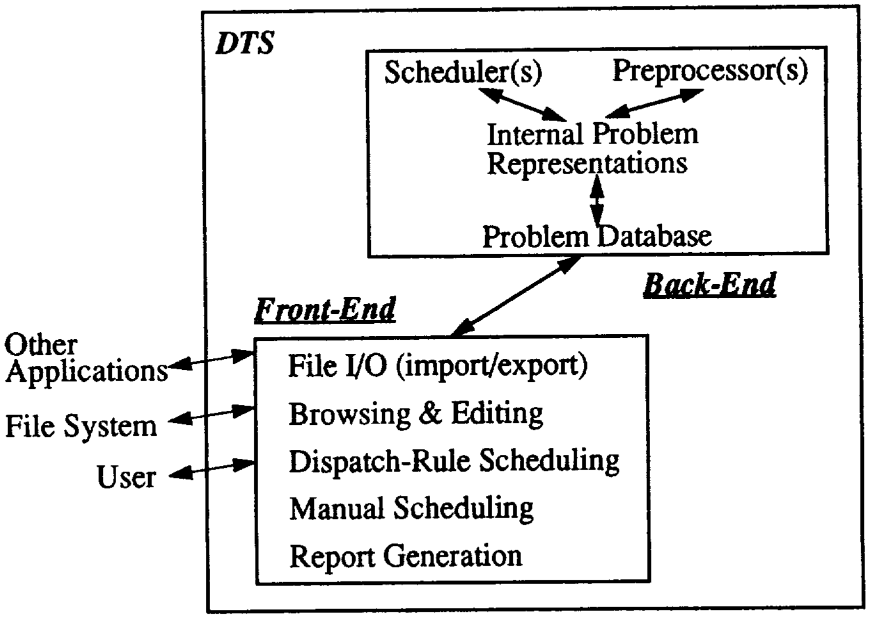
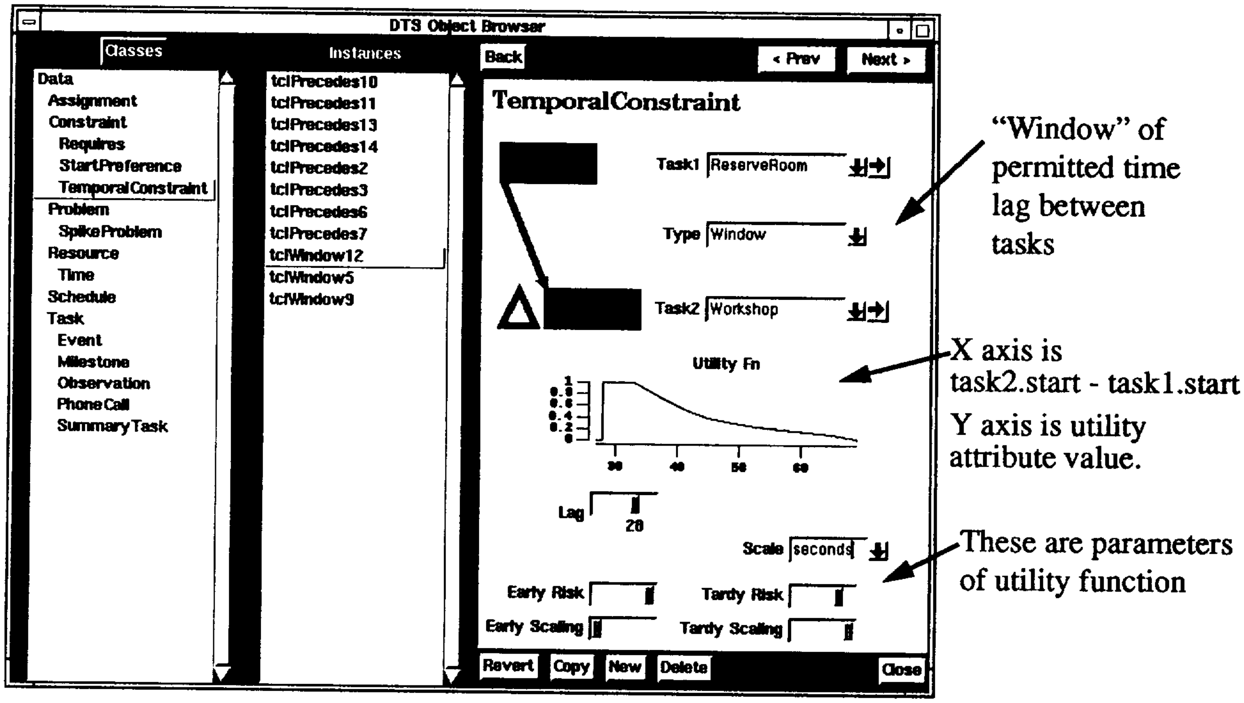


Figure 1. Overview of DTS System Architecture.



TemporalConstraint

Task1 ReserveRoom

Type Window

Task2 Workshop

Lag 20

Scale seconds

"Window" of permitted time lag between tasks

X axis is task2.start - task1.start

Y axis is utility attribute value.

These are parameters of utility function

Figure 2. Example Customized Form in the Object Browser.

for a number of interrelated attributes. Like most binary constraints, the temporal constraint has two task parameters. In addition, for constraints of type "window," a utility function is defined by the parameters at the bottom of the form. These parameters are "animated" in a utility graph. Finally, each type of constraint has an associated graphical mnemonic (the upper left of the form), which reminds the user of the nature of the constraining relationship.

The second major mechanism in the DTS user interface is the dynamic graph. Dynamic graphs are editable "views" of a number of objects, built using an X-Y graphing metaphor. For example, a typical Gantt chart is an X-Y plot of tasks (Y), using their start time and duration (X). The DTS dynamic graph permits views such as Gantt charts, PERT charts, constraint matrices and resource histograms to be specified easily. These graphs are dynamic in that callbacks can be associated with user actions (e.g., mouse events), and defined to modify the underlying data appropriately.

Each of the basic views implemented thus far has required approximately 250 lines of DCL code for layout and callbacks. Application-specific views (such as augmented Gantt charts, statistical summaries, etc.) should be implementable with similar effort.

OPTIMIZER CUSTOMIZATION

The DTS back-end includes C++ routines, callable through DCL, that perform basic preprocessing and scheduling tasks. This optimization engine uses decision-theoretic search mechanisms developed by the authors in previous and ongoing work with the Bayesian Problem-Solver [1,2,5].

The use of decision theory [6,7,8] enables the engine to guide its search by user-specific utility functions, in addition to heuristic evaluation functions. Many existing schedulers use heuristic functions alone, but heuristic functions can confuse the role of schedule evaluation (utility) and search control (heuristics).

DTS collects statistics that relate heuristic evaluations to attributes of the utility function. Because these statistics relate to inputs rather than outputs of the utility function, the function itself can be modified without invalidating the statistics that have been gathered. The use of statistical estimation and probabilistic inference in DTS also permits multiple heuristic evaluations to be combined to focus the search more effectively. For example, a general-purpose constraint-satisfaction heuristic might be coupled with a domain-specific heuristic [5].

In an early phase of development, we found that the costs of state generation and heuristic evaluation were a significant bottleneck to the development of sophisticated scheduling search control. DTS thus also employs an experimental compilation mechanism that derives a specialized data structure for search tree "states" from a formal specification of the heuristic function. Hand-coding of such data structures reduces the overall cost of search significantly, and we anticipate that the automation of these data structures will permit these benefits to be achievable for users relying on domain-specific heuristics. Hansson [9] describes the compilation mechanism in more detail.

Finally, the use of DCL permits a user to code a secure "audit" or "checker" routine to validate a finished schedule before execution, or to enforce certain scheduling policies that are hard to represent within the system.

Along with other DTS features, these three mechanisms—decision-theoretic search with user-specific utility functions, data structure compilation for fast heuristic evaluation, and postprocessing for schedule validity—have been designed to ensure that DTS finds solutions to the user's *real* problem with a minimum of search cost.

CONCLUSION

We are presently customizing DTS for possible use within current and future NASA missions (including EUVE and CASSINI), and collaborating with NASA researchers to reuse the DTS interface on top of their schedulers.

We feel that the customizability of DTS can permit future NASA missions to exploit "economies of reuse" and "economies of fidelity." Economies of reuse are well-known: they result when development costs are cut by reusing flexible software.

Economies of fidelity result when a system can be made to solve a large portion of an application task, without a great degree of simplification. Many search and optimization frameworks require the user to simplify or abstract their problem into a restricted modelling language. This increases the cost of using such systems, and reduces the benefits: the solutions found are not always executable, let alone near-optimal, solutions to the real problem. On the other hand, systems like DTS, and Muscettola's HSTS [10], attempt to provide a richer framework for modeling the problem. DTS focuses on preference modeling, while HSTS focuses on constraint and state-variable modeling. We anticipate that compilation and learning techniques will permit these rich representations to be searched efficiently.

BIBLIOGRAPHY

[1] Hansson, O. and A. Mayer. "Heuristic Search as Evidential Reasoning." In *Proc. of the Fifth Workshop on Uncertainty in Artificial Intelligence*, Windsor, Ontario, August 1989.

[2] Mayer, A. *Rational Search*. Ph.D. Dissertation, Univ. of California, 1994.

[3] Ousterhout, J. *TCL and the Tk Toolkit*. Addison-Wesley, Reading, MA, 1994.

[4] Center for EUV Astrophysics. *EUVE Guest Observer Program Handbook*. Appendix G of NASA NRA 92-OSSA-5. Berkeley, Jan. 1992.

[5] Hansson, O. and A. Mayer. "Decision-Theoretic Control of Artificial Intelligence Scheduling Systems." HRI Technical Report No. 90-1/06.04/5810, Berkeley, CA, September 1991.

[6] Pearl, J. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[7] Savage, L. J. *The Foundations of Statistics*. Dover, New York, 1972.

[8] von Neumann, J. and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[9] Hansson, O. *Bayesian Problem-Solving applied to Scheduling*. Ph.D. Dissertation, Univ. of California, 1994.

[10] Muscettola, N. *HSTS-DDL Manual*, NASA Ames Research Center, Code FIA, March 1994.

N95- 23745

Automated Synthesis of Image Processing Procedures Using AI Planning Techniques *

Steve Chien and Helen Mortensen

Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099

Tel: (818) 306-6144 Fax: (818) 306-6912
E-Mail: chien@aig.jpl.nasa.gov

KEY WORDS AND PHRASES

Automation, planning, science data analysis, large databases, science operations.

Introduction

This paper describes the Multimission VICAR (Video Image Communication and Retrieval[1]) Planner (MVP) (Chien 1994) system, which uses artificial intelligence planning techniques (Iwasaki & Friedland, 1985, Pemberthy & Weld, 1992, Stefik, 1981) to automatically construct executable complex image processing procedures (using models of the smaller constituent image processing subprograms) in response to image processing requests made to the JPL Multimission Image Processing Laboratory (MIPL). The MVP system allows the user to specify the image processing requirements in terms of the various types of correction required. Given this information, MVP derives unspecified required processing steps and determines appropriate image processing programs and parameters to achieve the specified image processing goals. This information is output as an executable image processing program which can then be executed to fill the processing request.

Currently, a group of human experts, called analysts, receive written requests from scientists for image data processed and formatted in a certain manner. These analysts then determine the relevant data and appropriate image processing steps required to produce the requested data and write an image processing program in a programming language called VICAR (LaVoie et al.1989).

Unfortunately, this current mode of operations is extremely labor- and knowledge-intensive. This task is labor intensive in that constructing the image processing procedures is a complex, tedious process which can take up to several months of effort. There are currently tens of analysts at MIPL alone whose primary task is to construct these VICAR programs. Many other users at JPL and other sites also write VICAR scripts, with the total user group numbering in the hundreds.

The VICAR procedure generation problem is also a knowledge-intensive task. In order to construct VICAR procedures, an analyst must possess knowledge of:

1. image processing and image processing programs (as of 1/93 there were approximately 50 frequently used programs, some having as many as 100 options)
2. database organization and database label information to understand the state of relevant data
3. the VICAR programming language to produce and store relevant information.

Because of the significant amount of knowledge required to perform this task, it takes several years for an analyst to become expert in a VICAR image processing area.

The MVP task targets automated generation of image processing procedures from user requests and a knowledge-based model of an image processing area using artificial intelligence (AI) automated planning techniques. In AI planning, a system uses: 1) a model of actions in a domain; and 2) a model

---

* This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Other past and present members of the MVP team are Christine Ying, Shouyi Hsiao, Alex Gray, Joe Nieten, and Jean Lorre.

[1] This name is somewhat misleading as VICAR is used to process considerable non-video image data such as MAGELLAN synthetic aperture radar data.

of the current state to reason about what actions to take to achieve some specified goals. By partially automating the filling of basic science requests, request turnaround time will be reduced, analysts' time will be freed for more complex and challenging science requests, and analysts' workload will be reduced.

VICAR is a general-purpose image processing programming language designed to promote the development and re-use of general-purpose image processing algorithms for MIPL needs. The primary function of VICAR is to allow individual image processing steps (called VICAR programs) to be combined into more complex image processing scripts called procedure definition files (PDFs). As one of their primary duties, MIPL analysts construct PDFs to perform image correction, image enhancement, construct mosaics, and to create movies and render objects. Individual processing programs perform functions such as:

photometric correction - correcting the image for lighting conditions due to the position of the sun relative to the imaging device and target,

radiometric correction - correcting for varying camera response depending on where in the field of view the image is read,

line fill-in - interpolating missing lines caused by data transmission errors.

By composing individual programs which perform these specialized functions, analysts can create complex image processing procedures (PDFs) to perform multiple types of correction and register the images to allow combination of multiple images into larger images.

The MVP Architecture

The overall architecture for the MVP system is shown in Figure 1. The user inputs a problem specification consisting of processing goals and certain image information using a menu-based graphical user interface. These goals and problem contexts are then passed to the decomposition-based planner which uses skeletal and hierarchical planning methods to classify the problem type and then uses this classification to decompose the problem into smaller subproblems. During this decomposition process, MVP determines which information on the database state is needed by the planner to solve the subproblems.

These subproblems are then solved by a conventional operator-based planner that uses the subproblem goals and initial states as indicated by the problem decomposition. The resulting plan segments are then assembled using constraints derived in the decomposition process. The resulting plan is then used to generate an actual executable VICAR PDF using conventional macro-expansion techniques.

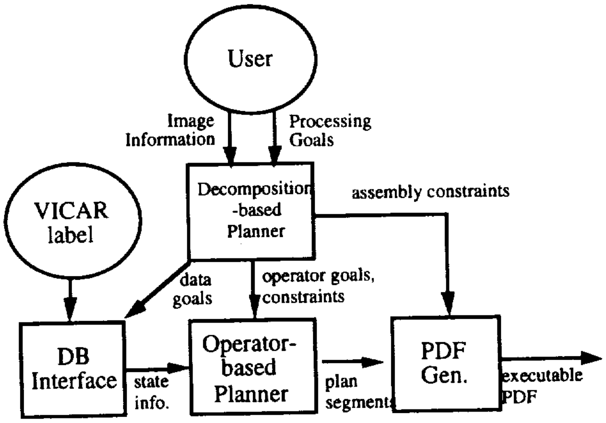


Figure 1: MVP Architecture

Plans in the MVP domain can be of considerable length (up to 100 steps) and each step (or VICAR program) can involve reasoning about numerous complex effects (many operators have tens of effects). Due to the large search space caused by this complexity, conventional operator-based planning approaches are not able to tractably construct plans in the VICAR domain without significant control knowledge.

Additionally, even if a purely operator-based planning approach were able to generate plans to solve the VICAR problems, these plans would be difficult for MIPL analysts to understand. Typically, analysts begin by classifying the general problem being addressed into one of a general class of problems, such as mosaicking, color triple processing, etc. They then use this classification and the problem context to decompose the plan into several abstract steps, such as local correction, navigation, registration, touch-ups, etc. A planning system which mimicked this approach to producing VICAR PDFs would be desirable.

Skeletal and Hierarchical Planning Using Decompositions in MVP

Skeletal planning (Iwasaki & Friedland 1985) is an approach to planning which casts planning as a structured classification problem. In skeletal planning, a planner identifies a new problem as one of a general class of problems, based upon the goals and initial state. This technique was originally developed as a model of experiment design in molecular biology; however, skeletal planning is also an accurate model of how expert analysts attack VICAR procedure generation problems. Typically, in a VICAR problem, there is a central goal for processing, such as mosaicking, which then dictates a decomposition of the overall problem into subproblems such as local correction, navigation, and registration. MVP attacks a VICAR problem by first determining the general problem class, and then using this problem class to perform an initial decomposition of the top-level image processing goals.

Hierarchical planning (Stefik 1981) is an approach to planning where abstract goals or procedures are incrementally refined into more and more specific goals or procedures as dictated by goal or procedure decompositions. MVP uses this approach of hierarchical decomposition to refine the initial skeletal plan into a more specific plan which has been specialized, based on the specific current goals and situation. This allows the overall problem decomposition to be influenced by factors such as the presence or absence of certain image calibration files or the type of instrument and spacecraft used to record the image. For example, geometric correction uses a model of the target object to correct for variable distance from the instrument to the target. For Voyager (VGR) images, geometric correction is performed as part of the local correction process, as geometric distortion is significant enough to require immediate correction before other image processing steps can be performed. However, for Galileo (GLL) images, geometric correction is postponed until the registration step, where it can be performed more efficiently.

MVP uses a decomposition-based approach (Lansky 1993) to perform Skeletal and Hierarchical planning. In a decomposition-based approach, decomposition rules dictate how in plan-space planning, one plan can be legally transformed into another plan. The planner then searches the space plans defined by these decompositions. Decomposition-based approaches are extremely powerful in that many other paradigms, such as modal truth criterion planning (Lansky 1993), can be implemented in a decomposition-based approach.

This decomposition-based approach to skeletal and hierarchical planning in MVP has several strengths. First, the decomposition rules very naturally represent the manner in which the analysts attack the procedure generation problem. Thus, it was a relatively straightforward process to get the analysts to articulate and accept classification and decomposition rules for the subareas which we have implemented thus far. Second, the notes from the decomposition rules used to decompose the problem can be used to annotate the resulting PDF to make the VICAR programs more understandable to the analysts. Third, relatively few problem decomposition rules are easily able to cover a wide range of problems and decompose them into much smaller subproblems.

Operator-based Planning in MVP

MVP uses classical operator-based planning techniques to solve subproblems produced by the decomposition-based planner. An operator-based planner uses: 1. a model of actions, A (in this case the model represents the requirements and effects of individual VICAR steps); 2. a specification of a current state, C (this corresponds to the current database state); and 3. a specification of a goal criterion, G (this corresponds to user request specification), to derive a sequence of actions, A', that when executed in the current state C, results in a state which satisfies the goal criterion G.

To illustrate this process, consider the following 5 simplified image processing operators shown in Figure 2. Preconditions are attributes which must be true of the image file before the step can be run, and effects are attributes which are made true by executing the step. This information can be summarized by the information shown below indicating the relevant programs for achieving the goals of missing line fill-in, spike removal, and radiometric correction for Voyager and Galileo images. When constructing a plan to achieve these goals, depending on the project of the image file (e.g., either Voyager or Galileo), MVP will know the correct program to use because the preconditions enforce the correct program selection.

Operator	VGRFILLIN	GLLFILLIN	ADESPIKE	FICOR77	GALSOS
Preconditions	VGR image EDR (binary header) present	GLL image	(GLL image) or ((VGR image) and (raw values))	VGR image	GLL image raw pixel values
Effects	missing lines filled in.....		spike removal	radiometric corr. blemish removal not raw values	radiometric corr. Reed-Solomon overflow corr. saturated pixel corr. not missing line fill-in

Figure 2: Simplified Planning Operators

However, determining the correct ordering of actions can sometimes be complex. In this case, the correct order to achieve the goals of line fill-in, spike removal, and radiometric correction is dependent upon the project of the file. In the case of Voyager files, ADESPIKE (spike removal) requires raw pixel values, and FICOR77 (radiometric) changes pixel values to correct for camera response function, so FICOR77 removes a necessary condition for ADESPIKE. This interaction can be avoided by requiring that ADESPIKE occur before FICOR77. VGRFILLIN requires a binary EDR header on the image file which is not maintained by ADESPIKE, this interaction can be avoided by requiring VGRFILLIN to be executed before ADESPIKE.

The Galileo case is slightly different. GALSOS undoes missing line fill-in so that it interferes with GLLFILLIN. This interaction can be avoided by enforcing GLLFILLIN after GALSOS. Additionally, GALSOS requires raw pixel values, and ADESPIKE alters the pixel values, so ADESPIKE interferes with this condition. This interaction can be avoided by requiring that GALSOS occur before ADESPIKE.

	Voyager	Galileo
fill-in missing lines	VGRFILLIN	GLLFILLIN
remove spikes	ADESPIKE	ADESPIKE
radiometric corr.	FICOR77	GALSOS
Execution Order:	VGRFILLIN	GALSOS
	ADESPIKE	GLLFILLIN
	FICOR77	ADESPIKE

This simple example illustrates the types of interactions and context-sensitivity that the VICAR image processing application entails. All of these interactions and context sensitive requirements are derived and accounted for automatically by MVP using the operator specification, thus allowing construction of plans despite complex interactions and conditions.

Current Status and Conclusions

MVP is currently operational and in use by analysts at JPL's Multimission Image Processing Laboratory (MIPL). Over a test suite of 5 typical mosaicking and color reconstruction tasks, an expert analyst estimated that MVP reduces effort to generate an initial PDF for an expert analyst from 1/2 a day to 15 minutes, and that it would reduce the effort for a novice analyst from several days to 1 hour.

MVP uses a combination of decomposition-based and operator-based planning paradigms to substantially automate the process of generating image processing procedures for radiometric correction and color triplet reconstruction. Current efforts involve expanding MVP to cover areas in filtering, stretching, and more complex relative navigation tasks.

References

(Chien & Mortensen, 1993) S. Chien and H. Mortensen, "MVP Domain Specification Document, Version 1.0 - Mosaicking with Absolute Navigation and Color Triplet Processing," JPL Internal Document, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 1993.

(Chien, 1994) S. Chien, "Using AI Planning Techniques to Automatically Generate Image Processing Procedures: A Preliminary Report," Proceedings of the 1994 Conference on AI Planning Systems, Chicago, IL, June 1994.

(Iwasaki and Friedland, 1985) Y. Iwasaki and P. Friedland, "The Concept and Implementation of Skeletal Plans,"Journal of Automated Reasoning 1, 1 (1985), pp. 161-208.

(Lansky, 1993) A. Lansky, "Localized Planning with Diversified Plan Construction Methods," Technical Report FIA-93-17, NASA-Ames Research Center, June 1993

(LaVoie et al. 1989) S. LaVoie, D. Alexander, C. Avis, H. Mortensen, C. Stanley, and L. Wainio, VICAR User's Guide, Version 2, JPL Internal Document D-4186, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 1989.

(Pemberthy & Weld, 1992) J. S. Pemberthy and D. S. Weld, "UCPOP: A Sound Complete, Partial Order Planner for ADL," Proceedings of the Third International Conference on Knowledge Representation and Reasoning, October 1992.

(Stefik, 1981) M. Stefik, "Planning with Constraints (MOLGEN: Part 1)," Artificial Intelligence 16,2(1981), pp. 111-140.

N95- 23746

Graphical Timeline Editing

Patrick E. Meyer and John P. Jaap
National Aeronautics and Space Administration (NASA)
Marshall Space Flight Center, AL 35812
Phone (205) 544-2265 Fax (205) 544-5873
Email: Patrick.Meyer@msfc.nasa.gov

KEY WORDS AND PHRASES

Conflict resolution, graphical editing, rescheduling, scheduling, space station.

INTRODUCTION

NASA's Experiment Scheduling Program (ESP), which has been used for approximately 12 Spacelab missions, is being enhanced with the addition of a Graphical Timeline Editor. The GTE Clipboard, as it is called, was developed to demonstrate new technology which will lead the development of International Space Station Alpha's Payload Planning System and support the remaining Spacelab missions.

ESP's GTE Clipboard is developed in C using MIT's X Windows System™ X11R5 and follows OSF/Motif™ Style Guide Revision 1.2.

Clipboard Concept

In ESP's GTE Clipboard concept, what is in the clipboard is *not* in the timeline. This reduces the permutations of potential conflict and allows for capabilities that would otherwise be too cumbersome or impossible. Activities to be edited must be moved to the clipboard, edited within the clipboard, and committed back to the timeline. When desired, a subset of the edited activities in the clipboard can be committed to the timeline. Activities from external sources can be added to the clipboard where possible conflicts can be resolved before they are moved to the timeline.

EDITING

"Editing," the key word in Graphical Timeline Editing, is a combination of rendering the timeline as graphics objects and supporting user manipulation of those objects. When *rendering* the data, pixel granularity inherent in graphic editors is overcome by using a "high resolution" screen. However, when *manipulating* data, the pixel granularity problem is exacerbated in the clipboard because ESP can have a scheduling horizon of 90 days or more and simultaneously maintain time accuracies to 1 second – a ratio of 1 to 10 million. In addition, users of ESP frequently need to maintain scheduling granularity at 1 minute. An eloquent solution to both problems was found and implemented.

The clipboard provides two methods for addressing time granularity problems. First, time quantization is used to round modified times to the nearest multiple of a user-specified constant. During an activity modification, feedback provides the quantized placement times so the user is constantly aware of the modified value. Second, a mechanism which allows the user to make micro-adjustments is provided. Micro-adjustments are always made in a quantization unit even when the unit is smaller than a pixel.

With a solution to the granularity problem in hand, fundamental graphical editing features such as selecting, adding, deleting, moving, modifying, undoing, aligning, and commiting back to the timeline can be straightforwardly implemented.

Activity Selection

For the clipboard, three methods and two modes of selection were identified. The methods include selecting the lowest integral part of a scheduled activity (a step), selecting a whole activity (a performance), and selecting all steps within a user-bounded box. For each of these methods, the modes of selection are additive (extending) and toggle.

Selection order may be relevant for some manipulations. For instance, to temporally align, both the activity which is aligned *to* and the activity which is aligned *with* must be specified. By using selection order, the alignment can be made consecutively; i.e. a successor can align with its predecessor.

Selected activities are rendered with angled appendages because it provides optimum viewability for overlapping and/or extremely small steps. Figure 1 below illustrates the usability of this method. Steps 1, 2, and 4 are shown as selected.

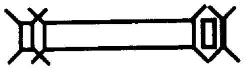


Figure 1 Selected Activity

Adding Activities

Activities can be added to the clipboard by moving them from the current timeline to the clipboard, copying from the current timeline, importing from an external timeline, including a template of a model, using the automatic scheduler, and keying in data.

Each of these methods has unique applications and advantages. Activities which could not be imported directly to the timeline may be imported to the clipboard, repaired, and then moved to the timeline. Adding a template of a model to the clipboard allows a user to start with a generic form of model and massage it to fit the timeline constraints. Adding to the clipboard via automatic scheduling provides conflict-free activities in the clipboard to which the user can make minor adjustments before moving them to the timeline. However, autoscheduling into the clipboard checks constraints only against the timeline and not against the clipboard.

Deleting Activities

Activities may be deleted from the clipboard. After deletion they may be restored until they are purged or until the entire contents of the clipboard is successfully committed to the timeline. As a shortcut, activities may also be deleted directly from the timeline.

Moving Clipboard Activities

Moving activities is the most used manipulation of a timeline and therefore should be robust and easy to use. In the clipboard, moving selected steps is initiated with a mouse button press while the pointer is within one of the selected steps. The initiating step's new start time is fed back to the user during the modification to indicate where the step will be placed. Other selected steps are also moved.

Modifying Activities

After moving or adding activities, small conflicts usually arise that require simple modifications for them to validly schedule as a group. For example, activity duration changes will routinely introduce overlap conflicts. By proper definition of activity duration changes, the user can prevent overlap conflicts. The definition chosen for the clipboard takes all succeeding (for modified start time) or all preceding (for modified end time) activities which are selected and shifts them to maintain the original time delay between the changed activity and any other *affected* activity. In Figure 2 below, the duration of B was changed causing C to be shifted.

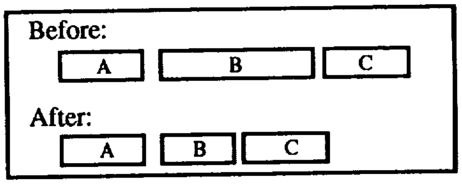


Figure 2 Duration Change

However, when changing the duration of an activity, the durations of overlapping selected steps are also modifed rather than being shifted. This overlap may or may *not* be due to scheduling concurrency constraints (see Figure 3 below).

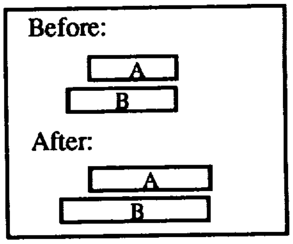


Figure 3 Overlapping Change

Modifying Crew Usage

A portion of the display can be used to display crew usage data for both the timeline and clipboard. Graphical manipulations within this area can remove, add, or reassign crew on an activity.

Undoing Modifications

Undo, as the name implies, allows the user to undo the *last* edit made to the clipboard. The last edit may be either a simple (one item) or compound (multiple items) edit. In addition, undo can also restore activities to their unedited state until they are committed to the timeline.

Assisted Temporal Aligning

The clipboard can align selected activity steps or activity performances in four ways:

- Start time to start time.
- Start time to end time.
- End time to start time.
- End time to end time.

Steps are aligned consecutively in the order in which they were selected.

Committing to the Timeline

As stated before, activities in the clipboard are not in the timeline and must be committed to the schedule. During committal to the timeline, activities are validated and any conflicts that the user has not resolved are reported. If no conflicts are present, the activities are moved from the clipboard to the timeline with no report.

USER INTERFACE

While building the basic editing commands into a new timeline editor, concerns of leaving behind old, but good, ways of editing arose. Therefore, the clipboard brings forward text-based editing features similar to ESP's previous timeline editing buffers. These include a table and command line which allow the user to enter specific values for important aspects of an activity and enter commands that can affect ranges of activities.

Tailorable Displays

Each Spacelab mission ESP supports has different objectives and resource utilization. To accommodate the varying demands on the clipboard, a tailorable display allows users to hone in on the information which is important to the type of missions they are scheduling. Some of the options included are —

- *Activity Breakdown* —Using Digital's Structured Visual Navigation (SVN) widget the clipboard is able to present the graphical data to the user in an expandable outline form. With SVN, the user can expand an activity to a Gantt chart of its steps.
- *Optional Command Line* —For the expert *graphical* editing user, the command line may be removed from the visible display.
- *Optional Crew Data* —For missions where activities are not crew-based, the crew timeline data can be removed.
- *Optional Crew Timeline Data* —Since what is in the clipboard is not in the timeline, the crew *timeline* data can be displayed in the same crew area with *clipboard* data.
- *Paned Windows* —The clipboard utilizes Motif's PanedWindow widget which allows the user to subdivide the display and view the most desired information.
- *Toolboxes* —The clipboard toolbox provides quick access to copy, delete, undo, auto-schedule (generate), move (bias), temporally align and other useful commands.

CONFLICT RESOLUTION/PREVENTION

The biggest drawback with many manual editors is their "guess-again" approach. When the user moves an activity, the scheduler response is to report resource constraint and scheduling conflicts and the user is forced to "guess again" where the activity can be validly placed to avoid the conflicts. The clipboard not only provides a report when edits are committed, it also displays graphics information to help the user eliminate existing conflicts and to avoid introducing conflicts when activities are manipulated. When fully implemented, the clipboard will also provide automatic scheduling assistance.

Visual Aid for Crew Conflicts

Since ESP enforces the Spacelab and International Space Station Alpha ground rule that a crewmember cannot be scheduled to do more than one task at a time, the clipboard provides a crew timeline area where the crew *timeline* data is displayed, the crew *clipboard* data is overlayed, and any conflicts are highlighted. Users can readily see where crewmembers are available or are in conflict and can modify crew usage for activities in the clipboard.

Visual Aid for Temporal Constraints

An activity can have temporal constraints relative to other performances of the same model or other models and can have fixed windows in which it must be performed. While other displays in ESP can show the data which determines where the temporal constraints are met, the clipboard can compute the feasibility windows and present them to the user so that temporal confícts can be avoided or removed. The clipboard uses an X Windows "work procedure" to compute and display these computationally-intense windows. Work procedures are executed when no events are pending execution; i.e., they use idle computer time. The clipboard re-launches the work procedure whenever the timeline is changed; therefore, the windows remain current.

Visual Aid for Resource Constraints

Each step of an activity can have many constraints which limit where on the timeline a step can be validly placed. In addition to crew requirements, these constraints include other shared resources and windows of opportunity. For each step, the clipboard can compare each requirement to the current timeline state, consolidate all the individual results, and present feasibility windows on a single line of the display. Windows which are not as long as the minimum step duration are not shown. By adhering to the feasibility windows, the user can easily avoid introducing conflicts or remove existing conflicts. A work procedure is also used for these feasibility windows. Figure 4 shows a step which is too long for its nearest feasibility window but which will fit within the next window.

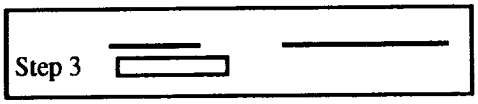


Figure 4 Feasibility Windows

Users may request that feasibilities for a step be presented on an individual requirements basis. This separate display will show each requirement of the selected step and the time periods when that requirement is met. The composite feasibility windows are also shown.

Automatic Assistance to Editing

As an enhancement of the manual editing process, the clipboard provides automatic assistance. In this mode, the user selects a performance that is in the clipboard but conflicts with some constraints. The user specifies a horizon over which scheduling is to take place, and an automatic scheduler adjusts step start times, durations, and crew assignments to avoid conflicts. An explanation facility which graphically explains what the scheduler did allows the user to understand what conflicts were addressed or why a solution was not found.

A future enhancement of automatic rescheduling is to allow the user to specify another horizon over which the program may automatically adjust activities already on the timeline. This process may proceed without user interaction, or user approval may be required for every change to the timeline. There are several strategies for directly modifying the existing timeline:

- Reassignment of crew.
- Adjusting step durations or delays.
- Substituting an alternate scenario of the same activity.
- Moving activities to non-conflicting times.
- Moving activities to new times where conflicts can be easily resolved.
- Deleting activities based on priorities or based on least impact to schedule.

Advanced techniques such as simulated annealing are also candidates for automatic assistance.

SUMMARY

The X Windows Graphical Timeline Editor currently being added to the Experiment Scheduling Program is based on a clipboard concept. Because activities which are in the clipboard are not on the timeline, robust capabilities which may not have been otherwise practical are being incorporated. The GTE Clipboard is being implemented using Motif™ and other widgets and includes a complete suite of graphic manipulation features and a complete suite of timeline editing features relative to Spacelab and International Space Station Alpha. The GTE Clipboard is much more than a "guess-again" editor in that it provides graphical assistance for conflict resolution. Future plans will implement automated conflict resolution.

N95- 23747

A Format for the Interchange of Scheduling Models

John P. Jaap & Elizabeth K. Davis
National Aeronautics and Space Administration (NASA)
Marshall Space Flight Center, AL 35812
Phone (205) 544-2226   Fax (205) 544-5873
John.Jaap@msfc.nasa.gov

KEY WORDS AND PHRASES

Activity modeling, activity scheduling, model interchange, space station.

INTRODUCTION

In recent years a variety of space-activity schedulers have been developed within the aerospace community. Space-activity schedulers are characterized by their need to handle large numbers of activities which are time-window constrained and make high demands on many scarce resources, but are minimally constrained by predecessor/successor requirements or critical paths.

Two needs to exchange data between these schedulers have materialized. First, there is significant interest in comparing and evaluating the different scheduling engines to ensure that the best technology is applied to each scheduling endeavor. Second, there is a developing requirement to divide a single scheduling task among different sites, each using a different scheduler. In fact, the scheduling task for International Space Station Alpha (ISSA) will be distributed between NASA centers and among the international partners. The format used to interchange scheduling data for ISSA will likely use a growth version of the format discussed in this paper.

The model interchange format (or MIF, pronounced as one syllable) discussed in this paper is a robust solution to the need to interchange scheduling requirements for space activities. It is highly extensible, human-readable, and can be generated or edited with common text editors. It also serves well the need to support a "benchmark" data case which can be delivered on any computer platform.

FILE FORMAT

The data which is interchanged via the model interchange format is contained in a data set or file. When the data is stored in a file on a platform which supports a file extension as part of the file name, the extension ".MIF" should be used.

A MIF file is arranged in lines or records. Each record contains *a single* keyword and may contain data values. Keywords are surrounded by vertical bars. They are case sensitive and should not contain characters, such as spaces or commas, which might be used as input delimiters in common user interfaces.

The information is organized as a hierarchy in parent, child, sibling relationships. No keyword can be the same as an ancestor or the sibling of an ancestor. Therefore, on any record, a keyword which is a child, sibling, ancestor, or sibling of an ancestor of the keyword of the previous record may be listed without ambiguity. But, while descending the hierarchy, ancestral keywords can*not* be skipped. To obtain the full meaning of the data on a record, the keyword on the record and all keywords (records) in its ancestry must be considered. The order of the records in a file is usually significant; arbitrary reordering of the information is not allowed.

The file format limits the use of vertical bars to keywords and disallows the use of the backslash (\) character throughout. Identifiers or names cannot contain a comma, parenthesis, or space. The file format also specifies formats for the following data types: single integer, multiple integers including range-of-integers, real numbers, time expressions, date expressions, and character strings.

FILE CONTENTS

At the time of this writing, over 200 keywords have been defined in three independent hierarchy structures: a data set description hierarchy, a mission model hierarchy, and an activity model hierarchy. Figure 1 shows the logical organization of the file, with hierarchies shown for the data set description, mission model, and several activity models.

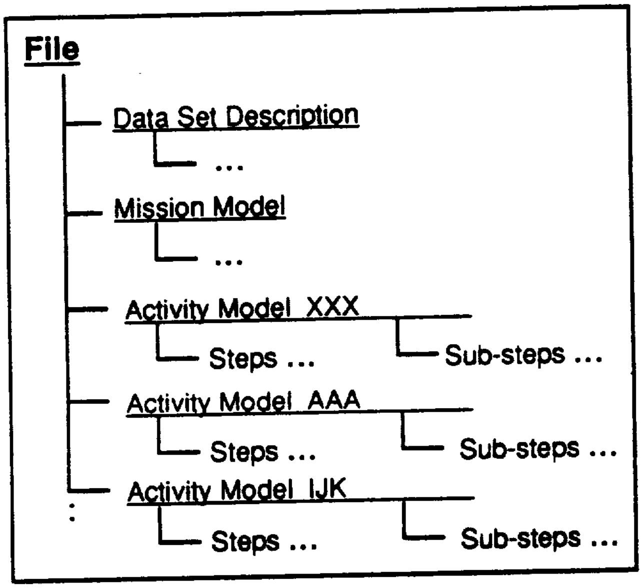


Figure 1. MIF File Organization.

Data Set Description

The data set description tells what is on the file, its source, and related data.

Mission Model

The mission model describes the availabilities of the resources for a particular scheduling task. The following items are included in the mission model:

- Identifiers and descriptive data.
- Resource availability profiles (including resource envelope definitions).
- Equipment reconfiguration data and crew relocation times.
- Pre-scheduled crew timeline and duty cycle data.

Activity Models

Activity models are used to describe the payloads or experiments to be scheduled. Each payload or experiment requires one or more activity models; for complex payloads, an activity model is usually included for each functional objective.

An activity model is the collection of constraint definitions describing a payload or experiment. Some of the constraints apply to the model as a whole, while others only apply to the model partitions, known as steps and sub-steps.

The smallest required, fully functioning, clearly delineated partition of an activity model is called a step. The steps of an activity model describe most of the resource constraints of the model. Each activity model in a MIF file must have at least one step.

The optional partition of an activity model which supports the execution of one or more steps is called a sub-step. Two classes of sub-steps are currently defined: crew monitoring sub-steps and resource carry-through sub-steps.

An execution of an activity model is called a performance. The performance of an activity model is generally considered to consist of the execution of the model's steps and sub-steps. A model may be performed multiple times to collect data. Each performance may contain a different set of steps and sub-steps, or they may be arranged differently when compared to other performances of the same model. Step-based schedulers usually require that each performance contain at least one step.

The model/step/sub-step structure for representing requirements was chosen because it was judged to be more robust than other representations. This representation observes well the axiom that models should exhibit high fidelity and flexibility; high fidelity means that an observer can correlate the model to the actual activity; high flexibility means that the model can represent the scheduling flexibility of the actual activity. The chosen representation also supports interchanges with schedulers which use models with requirement profiles attached directly to the model; in the model interchange format, a one-step model with requirement profiles on that step is used.

AVAILABILITY

Currently the Mission Planning Division at the Marshall Space Flight Center is the keeper of this format. As stated earlier, a growth version of this format is expected to be used for interchanging scheduling data for International Space Station Alpha (ISSA). The current definition may be superseded by the ISSA definition, and the ISSA configuration control function may become the keeper of this format.

Those wishing to have the format extended should contact the authors of this paper.

Documentation

The complete model interchange format is available in printed form, or electronically in PostScript, Bookreader, and possibly World View formats. Documentation can also be accessed on the World Wide Web via Mosaic.

Sample Data Cases

Sample files containing the subset of the defined format currently used by Marshall Space Flight Center are available for several Spacelab missions and some ISSA data cases.

Software Requirements

Implementers of software which reads a MIF file should allow for extensions of the format. Since at some future date new keywords may be defined, the software should contain code to ignore unrecognizable keywords. They must also develop mapping code to convert data in the MIF representation to their internal representation.

Implementers of software which writes a MIF file must develop mapping code to translate their data to the MIF representation.

SAMPLE MIF DATA

Figure 2 shows part of a two-step activity model with variable separations and durations. This figure also shows a sub-step (its keyword is `|-step|`). The sub-step could also have been positioned before step 1 or after step 2. Representing the requirements as two steps and a sub-step is considered a high fidelity representation because it closely matches the usual description of the activity. The model has high flexibility because it captures the variable step separation and duration, and the choice of crewmembers for step 2. As shown, the model does not require that high power be available immediately before shadow. The sub-step would not be scheduled whenever the separation between

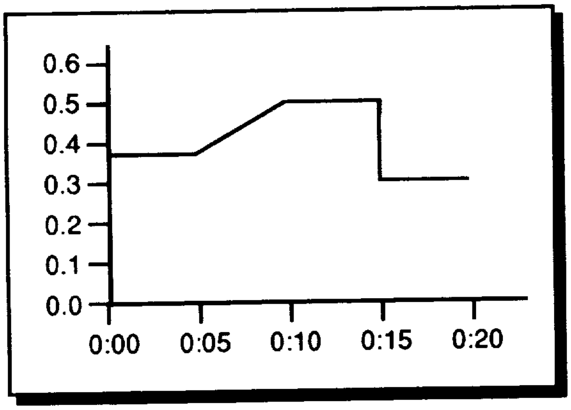
```
|Model| SEPAC-1
    |Comment| Beam firing (low power)
    |Partner| USA
    |step| 1
        |description| Capacitor charge
        |science_value| 0
        |duration| 0:15:00
        |nondepletable| POWER
            |profile|   2.750
        |carry-through|
            |sub-step| trickle
    |-step| trickle
        |-description| capacitor trickle charge
        |-nondepletable| POWER
            |-profile|  0.128
    |step| 2
        |description| Beam Firing (level 1)
        |separation|
            |min|   0:00:00
            |max|   0:30:00
        |duration|
            |min|        0:10:00
            |max|        0:20:00
            |preferred| 0:20:00
        |science_value|  4
        |crewlist|
            |type| FULLTIME
            |pick| 1
            |crewmember| PS1
            |crewmember| PS2
            |at_location| MODULE
        |crewlist|
            |type| FULLTIME
            |pick| 1
            |crewmember| Commander
            |at_location| MID-DECK
        |intersected_opp| SHADOW
        |intersected_opp| K-band
        |nondepletable| POWER
            |profile|   0.500
```

Figure 2. A Model with Variable Durations.

steps 1 and 2 is zero.

Figure 3 presents part of a one-step activity model with a power profile (shown in inset).

```
|Model| SEPAC/co
    |Comment| Check out hardware
    |step| 1
        |duration|  0:20:00
        |nondepletable| POWER
            |profile|
                |time|        0:00:00
                |usage|           0.381
                |time|        0:05:00
                |usage|
                    |initial|     0.381
                    |final|           0.50
                |time|        0:10:00
                |usage|          0.50
                |time|        0:15:00
                |usage|          0.30
```



POWER Requirement Profile

Figure 3. A Model with a Power Profile.

This type of model would be used in schedulers which use activity models with requirement profiles attached directly to the model. The current format limits profiles to constant values, ramps, and step functions.

SUMMARY

The four significant requirements that drove the formulation of the model interchange format were that it must be universal, extensible, portable, and human-readable. Since this format was developed chiefly for the interchange of data, rather than for the storage and manipulation of data within schedulers, these requirements were deemed to be more important than efficiency.

Universality

A format was needed which could be used by all space-activity schedulers. This format provides for all known constraints and requirements which affect the scheduling of space activities. The section entitled "File Contents" in this paper describes the current contents.

Extensibility

It was necessary that the format be one which can evolve as new capabilities are added to existing schedulers and as new schedulers are developed. This format may be extended by adding to existing hierarchies; i.e., by defining new children or siblings at any level. Entirely new hierarchies may also be defined; this is equivalent to defining siblings of the highest level in the current hierarchy.

Portability

Since currently available schedulers are on different platforms, a format which could be read or written on any platform was needed. To this end, the information is stored in a MIF file as ASCII characters only and is line- or record-oriented. A benefit of this characteristic is that the file can be edited with common text editors.

Human-readable

A person can easily read a MIF file or use a text editor to create/edit one by virtue of the following attributes:

- The syntax is simple. There are a limited number of rules and special characters.
- A cascading outline hierarchy is used. Each entry in the hierarchy resides on a separate line with no other keywords. Ancestry keywords are not necessarily repeated; the common conventions for outline presentation are followed.
- The format is free. Virtually nothing within a line is positional. The user can indent as desired to improve readability.

N95-23748

A Data Analysis Assistant

Amy L. Lansky

Recom/NASA Ames Research Center
MS 269-2, Moffett Field, CA 94035-1000
tel:(415)604-4431 fax:(415)604-3594
LANSKY@PTOLEMY.ARC.NASA.GOV

Keywords and Phrases: constraint satisfaction, data analysis, planning, software agent.

1 Introduction

This paper discusses the use of a domain-independent planner, COLLAGE, as a software assistant to Earth scientists working with remotely-sensed and ground-based data sets. The planner can be viewed as an *advisory agent* that helps scientists select appropriate data and create a suitable plan for data-processing that meets stated scientific requirements [4].

Though we have worked on this domain for some time, only recently have we come to view it as an instance of a much broader class of potential planning applications: helping humans to navigate through seas of software- and data-selection possibilities. In general, tasks that involve human interaction with visualization tools often manifest this particular kind of challenge. The human has very deep knowledge of their domain (e.g., the scientist knows about Earth science; the graphic artist knows what kind of image they are trying to produce). However, the tools available may be too vast or complex (e.g., there may be hundreds of possible data set options; there may be hundreds of data transform or image processing algorithms available). Thus, the human knows *what* they want to accomplish, but doesn't know *how* to use the software to accomplish it.

We believe there is great potential payoff in the development of planning applications of this kind. Human experts desperately want the kind of help such systems could provide, and there is a high likelihood that they can successfully implemented. Besides our own work, a few other planning applications in this class are being developed [1, 2].

This kind of domain has two other interesting characteristics:

- It would be almost impossible to imbue a planning system with enough deep knowledge about the domain to accomplish the desired task autonomously.
- It is feasible to imbue a planner with the kind knowledge that a user doesn't have or doesn't want to be bothered with: what data and data manipulations algorithms are available; what functions these algorithms perform (at a high level of abstraction); and what usage constraints and requirements are attached to algorithms and data sets. For example, our Earth scientist experts currently make use of numerous data bases and at least two or three data analysis packages, each providing tens to hundreds of functions, with a variety of constraints on their use. The size and complexity of these data bases and packages, as well as their interactions, can make the data analysis task a logistical nightmare.

These two factors lead to a natural functional role for the kind of application we are developing. The planner will provide advice to the scientist about what data sets are available and what sequence of processing algorithms may be appropriate for their task. However, it does not try to make data or algorithm choices that require deep scientific knowledge of the problem. Instead, the planner has a *dialogue* with the user, presenting useful information and plan options, interactively refining choices with the user, and performing constraint checking as appropriate, given its knowledge about domain requirements.

Thus, the role of our data analysis system is to give the of level advice a user wants and to stay well informed in order to provide that advice. Our planner must "sense" available data and algorithms, as well as feedback from the scientist. The system "affects" its environment by providing advice to the scientist. Notice that this role is much deeper than that provided by a smart interface. The kind of planning required is quite complex; scientists currently utilize human technicians to do much of what our system is being designed to provide.

The rest of this paper begins with a quick description of the data analysis task. Then, we provide a summary of the COLLAGE architecture and current project status. Finally, we discuss two issues relevant to this application: planning vs. execution and system utility.

2 The Data Analysis Task

The development and validation of Earth-system models and change-detection analyses require several kinds of inputs, including remotely-sensed images (taken by satellite instruments) and ground data (e.g., meteorological readings, soil maps, and vegetation surveys). After data sets are retrieved and before they can be used, they must all be *registered* so that they lie within the same coordinate system and scale – i.e., all coordinate values must accurately correspond to one another. Unfortunately, the scientist's task of selecting suitable data and acceptably registering them is more difficult than it might seem. This process is often a burdensome and tedious portion of the scientific cycle that can consume over half of a scientist's time.

One reason is that required data is often resident in several physically distributed data bases and is encoded in a variety of formats, densities, scales, and projections onto Earth's surface. In addition, the same kind of information may exist in several different forms, may have been sampled in different ways, or may be derived through models. Thus, a scientist has many possible information sources to choose from, each associated with its own tradeoffs.

Once sources of information have been determined and data sets have been retrieved, scientists must register them. Unfortunately, heterogenous data types are often not directly comparable. For example, sparse vegetation data collected on the ground is usually not directly correlatable to satellite image data. Thus, a methodology is utilized that registers all data sets for a particular application to a common base map. Figure 1 depicts a high-level view of this process. First, a target coordinate system and scale is chosen. This target system is typically one that is similar to a majority of the data sets to be registered and that meets scientific and data-related constraints. Next, a *base map* of the study area is chosen that conforms to the target system. Then, all data sets are registered to this map. Depending upon the base map and the original form of a data set, required preprocessing steps may include geometric corrections, projection and scale transforms, radiometric corrections, atmospheric corrections, data restoration, interpolation, image enhancement, and ground control point selection (points that are used to achieve a correspondence between a data set and base map). Each of the steps depicted in Figure 1 would typically be composed of several substeps or processes. For each step, there are often a variety of possible algorithms, programs, and computational platforms. The choices made for each step must meet a variety of constraints that encode dependencies on and between registration steps. If poor choices are made, the registration process may introduce unacceptable distortions into the data. In some cases, registration may be impossible.

Consider the (simplified) registration plan depicted in Figure 2. Suppose that we have already selected and must now register two data sets – Thematic Mapper (Landsat) image data of Oregon and ground vegetation data for Oregon supplied by the US Forest Service in latitude/longitude coordinates. Our goal is to filter the image data through an equation that computes a vegetation index value for each image pixel and then plot these values against the ground-based vegetation values. First we select a target projection system of Universal Transverse Mercator (UTM) coordinates at a 30 meter scale and retrieve a suitable base map. Because latitude/longitude and UTM are both universal coordinate systems, the meteorological data is fairly readily registered using existing programs.

Registering the Thematic Mapper data, however, requires the use of *ground control points*. Each ground control point is a physical location for which coordinate information is supplied from both the original data set and the base map. Using these coordinates, a transformation matrix can be computed that accurately translates all data set values into the target base map system. The challenge is finding adequate ground control point coordinates (both in number and accuracy) that are also uniformly distributed. If the points are skewed towards a certain portion of the study area, the transform matrix will yield unsuitably skewed results. Indeed, if the original data set or base map does not contain enough discernable features, ground control point selection may be impossible and other options must be considered. For example, an alternate base map may exist for which adequate ground control points can be found. Or, a useable base map may exist in some *other* coordinate system that is then easily registered to the target system. One might also decide to choose an alternate target system or an alternate data source that has more identifiable features.

The data selection and registration process we have just described is full of compromises and tradeoffs, which also make it time-consuming and error-prone. There is intrinsic conditionality and interdependency between steps, often resulting in backtracking and replanning. In some cases, failures or errors during execution may require portions of the plan to be modified "on the fly" (e.g., after data visualization, the scientist may realize that additional corrective transforms must be applied). Currently, scientists cope with the difficulties of this task by falling back on particular approaches they are familiar with, rather than those that are most suitable for a particular problem. As a result, they often end up using unsuitably flawed data sets. And because this process is rarely documented, it is quite difficult to diagnose the source of data distortions or to reuse previously successful plans.

However, these characteristics also make this domain amenable to automation. Besides helping to speed up an otherwise tedious and time-consuming task, automation enables the exploration of a much

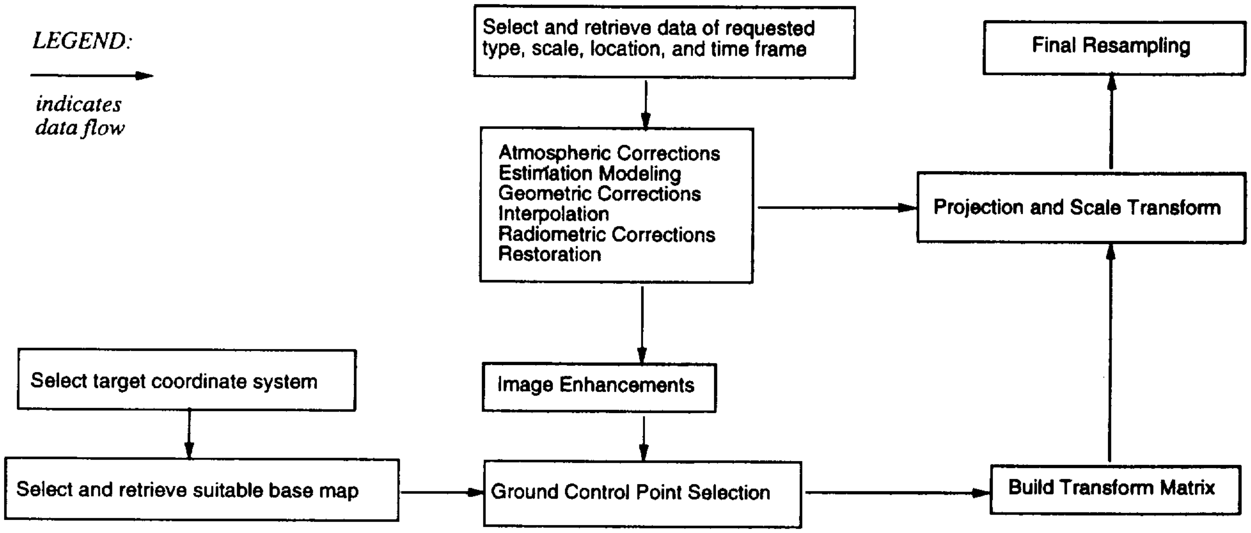


Figure 1: The Data Selection and Registration Process

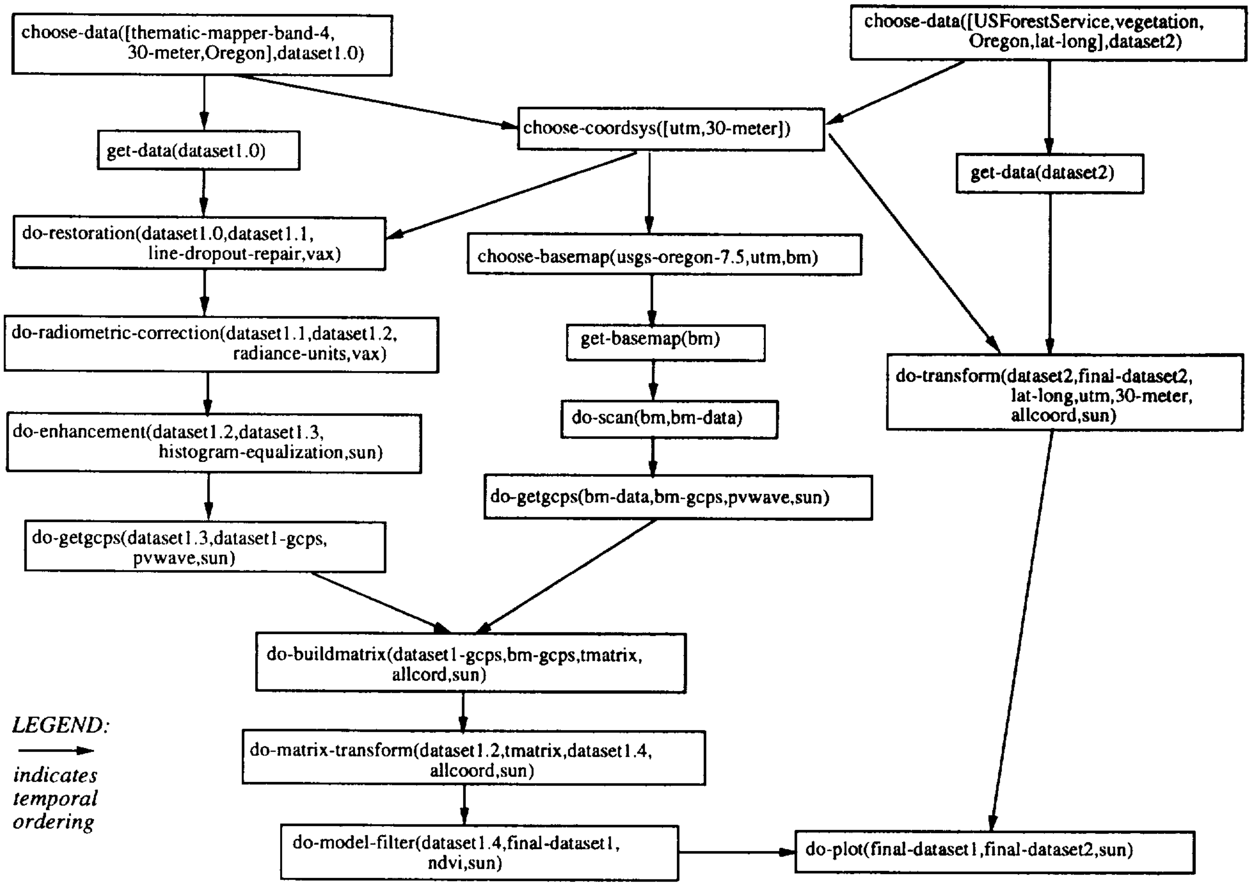


Figure 2: A Data Selection and Registration Plan

more complete range of data selection and registration possibilities and much more thorough constraint and integrity testing. Using an automated tool also enables documentation and justification of data selection and registration choices, thereby allowing for the possibility of diagnosis and plan reuse.

3 Current Status

COLLAGE[1] is a non-traditional domain-independent planner that may be viewed as a general-purpose constraint-satisfaction engine [3]. In the COLLAGE framework, the term "constraint" is used very broadly -- it is any type of requirement that the planner knows how to test and fulfill. Unlike the state-based encodings utilized by traditional planners, COLLAGE describes all domain requirements in terms of *action-based constraints*. Such constraints define domain characteristics strictly in terms of desired action interrelationships and action-parameter bindings requirements. The planner encompasses a wide variety of action-based constraint forms, each associated with constraint satisfaction algorithms that add new actions into a plan, decompose actions into subactions, impose ordering constraints, and constrain action-parameter bindings.

Instead of searching one large constraint-satisfaction search space, COLLAGE conducts its planning in a partitioned or *localized* fashion, searching a set of smaller (though possibly interacting) search spaces, each focusing on a subplan and a subset of the domain constraints. In the data analysis domain, these planning subproblems roughly correspond to the different data analysis subprocesses.

Over the past year, we have encoded the data analysis task in our constraint language and have extended the underlying COLLAGE planning framework and constraint library to meet the needs of this specification. We have also extended the system to include a static domain knowledge base that can drive and control aspects of the planning process. For this domain, the knowledge base includes facts about Earth's projection systems as well as information regarding available data processing algorithms. We are currently working our scientist experts to extend the domain knowledge base and create sufficient problem data to yield a set of planning problems for choosing and registering data for ecosystem models. We are also hooking COLLAGE up to the KHOROS image processing framework [5]. As part of this effort, we are developing mechanisms for automatically downloading information about the KHOROS algorithms into COLLAGE and for automatically visualizing and executing COLLAGE's plans in KHOROS's **Cantata** programming environment.

4 Discussion

Planning, Execution, and the User

This domain poses several interesting questions about planning vs. execution as well as the role of the user in the planning process. As we began to write the constraints for this application and deepen our understanding of the role of our planner vis-a-vis the user, we began to see traditional distinctions and roles becoming blurred. For example, in this domain, "execution" may be viewed in terms of data-retrieval and data-processing actions. Sometimes, the planner can autonomously execute these actions. In other cases, these actions must be performed by the scientist. This is because many image processing steps often require human interaction - for instance, to select image points with the naked eye.

As far as *when* planning occurs, much of the data analysis process must be planned in advance of execution; for example, scientists would be loathe to order expensive data sets or perform tedious manually intensive transforms unless they have created a data analysis plan that they are fairly sure will succeed. However, some forms of execution must take place during the planning process. For example, some preliminary information about data sets must be retrieved during "pre-planning" in order to enable reasoning about which algorithms are most appropriate to use.

However, some parts of the plan must also be filled out or modified during actual data processing. For example, the ground-control-point selection process is often iterative - new points must sometimes be added, others deleted in order to yield the best registered image. These plan extensions can't be determined until execution time, when an actual transform matrix is built and tried. Similarly, the most appropriate image enhancements for a data set often can't be fully determined until execution time, when the scientist can dynamically visualize those enhancements.

In summary, the domain requirements we have just described don't neatly fit traditional notions of reactive planning nor classical search-based pre-planning. Instead, the desired planning behavior can be viewed as a dialogue between the planner and the user, who are involved in a *collaborative* effort. The planner must be able to flow between classical deliberative reasoning, more dynamic forms of user-interaction and control over the planning process, and dynamic plan modification in response to the execution environment or user-directives.

For this reason, we have designed COLLAGE to enable a more fluid form of reasoning that we call ***flexi-time planning***. The system already allows for some forms of actions (e.g., choices, data retrievals, interactions with the user) to be performed durin "pre-planning." Soon, we hope to extend COLLAGE so that constraints can be triggered at any time relative

---

[1] Coordinated Localized aLgorithms for Action Generation and Execution.

to "execution." The COLLAGE constraint-triggering mechanism was intentionally designed to enable this kind of extension.

Utility

Given the advisory role of our data analysis planner, utility is critical. Does it provide good, up-to-date advice? Is it easy to use? We are addressing these issues in at least two ways. First, we are placing all forms of domain knowledge that are relevant and understandable to the scientist in a domain knowledge base that is distinct from the planning engine and domain constraint specification. Unlike domain constraints, the knowledge base may be viewed as static domain- and problem-specific factual information. For this domain, the knowledge base consists of information about Earth projection systems, constraints on usage of specific data types, projections, and scales, information about available data transform algorithms, and problem-specific data analysis goals. It also includes some domain-specific function definitions. The planner uses the knowledge base by conditioning the constraint-satisfaction process on knowledge-base contents and by using the domain-specific facts and functions to define binding requirements on plan variables.

Keeping the knowledge base distinct from the COLLAGE domain constraint specification and planning engine has several features that enhance utility:

- Planning functionality can be increased by extending the knowledge base rather than by extending the domain constraint specification.
- The same constraint specification can be used in numerous contexts with different knowledge bases.
- The knowledge base can be represented in a form amenable to viewing and extension.

The last feature is critical since we cannot possibly gather all domain-relevant information for this application. New data bases and algorithms are always being developed within the scientific community. To be truly useful, the system must be easily extendible by the user or via some other mechanism (such as automatically downloading information from KHOROS). Thus, a critical aspect of the utility problem is domain knowledge capture, which we hope to facilitate by making incremental knowledge easy to add and use.

A second aspect of utility is ease of use. We hope to foster this through our development of COLLAGE's integrated user interface, COLLIE. The COLLIE user can visualize the growing plan, inspect properties of each action, relation, and binding, and understand the relationship between plan structure and domain constraints. Features are provided for viewing a graphical representation of the domain structure and editing the domain specification and knowledge base. Eventually, we will extend COLLIE to include an improved interface to the knowledge base and allow users to modify the plan itself as well as interact more directly with the constraint search control mechanism.

Acknowledgements

We would like to thank our domain expert, Jennifer Dungan, and our NASA Goddard collaborator, Nick Short Jr. We would also like to acknowledge the contributions of past and present members of the COLLAGE project team: Lise Getoor, Andrew Philpot, Scott Schmidler, and Phil Chu.

References

[1] Boddy, M., J. White, R. Goldman, and N. Short,Jr. "Planning Applications in Image Analysis," *Proceedings of the 1994 Goddard Conference on Space Applications of AI*, pp. 17-28 (1994).

[2] Chien, S. "Using AI Planning Techniques to Automatically Generate Image Processing Procedures: A Preliminary Report," *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, Illinois pp. 219-224 (1994).

[3] Lansky, A. "Localized Planning with Diverse Plan Construction Methods," NASA Ames Research Center, Artificial Intelligence Research Branch, Technical Report FIA-94-05 (1994). Submitted to AIJ, Special Issue on Planning.

[4] Lansky, A. and A. Philpot, "AI-Based Planning for Data Analysis Tasks," Proceedings of the Ninth Conference on Artificial Intelligence for Applications (CAIA-93), Orlando, Florida, pp. 390–398 (March 1993). Also appeared in *IEEE Expert Magazine*, Volume 9, Number 1, February 1994.

[5] Rasure, J.R. and C.S. Williams, "An Integrated Data Flow Visual Language and Software Development Environment," *Journal of Visual Languages and Computing*, Volume 2, pp. 217–246 (1991).

Coordinating Complex Decision Support Activities Across Distributed Applications

Richard M. Adler
Symbiotics, Inc.
725 Concord Avenue
Cambridge, MA 02138
Tel (617)876-3635 Fax (617)876-0157
rma@symbiotics.com

N95-23749

KEY WORDS AND PHRASES

distributed computing, distributed decision support, intelligent coordination, work flow, work groups

INTRODUCTION

Knowledge-based technologies have been applied successfully to automate planning and scheduling in many problem domains [1,2]. Automation of decision support can be increased further by integrating task-specific applications with supporting database systems, and by coordinating interactions between such tools to facilitate collaborative activities. For example, end-to-end decision support for space missions involves a succession of interactions to transfer and manipulate data across diverse tools: deriving mission task, resource, and constraint networks via a planning engine; storing these results in a database; retrieving the mission plan for use as input to a scheduling engine; comparing the resulting schedule against current schedules for other missions to detect resource conflicts; and replanning or rescheduling to resolve problems. Ideally, no human intervention should be required to carry out such activity sequences, which, despite their complex distributed implementation, are otherwise well-defined and routine.

Unfortunately, the technical obstacles that must be overcome to achieve this vision of transparent, cooperative problem-solving are daunting. Intelligent decision support tools are typically developed for standalone use, rely on incompatible, task-specific representational models and application programming interfaces (APIs), and run on heterogeneous computing platforms. Getting such applications to interact freely calls for platform independent capabilities for distributed communication, as well as tools for mapping information across disparate representations [3]. Similarly, coordinating interactions dynamically presupposes capabilities for: identifying and locating required resources and capabilities across a network; capturing relationships between decision support activities such as task decomposition, data dependencies, and synchronization constraints; and autonomously controlling the execution of tasks across applications to reflect such relationships. These system engineering issues are largely orthogonal to the interests and skills of developers and end-users of decision support applications.

Symbiotics is developing a layered set of software tools (called NetWorks!) for integrating and coordinating heterogeneous distributed applications. The top layer of tools consists of an extensible set of generic, programmable coordination services. Developers access these services via high-level APIs to implement the desired interactions between distributed applications. Current API-based services enable developers to: register application services and information resources, their locations, and calling interfaces; model the decomposition or workflow sequence of composite decision support activities in terms of simpler units; and invoke automated control engines that execute composite models to carry out complex activities such as end-to-end decision support for space missions. The high-level coordination services are built on top of a communication substrate layer, which utilizes object-oriented technology to conceal the complexity of platform dependencies, data mapping, and network communication. The remainder of this abstract describes these various tools and how they interoperate as a nonintrusive, extensible framework for developing complex distributed applications.

DISTRIBUTED COMMUNICATION SUBSTRATE

NetWorks! is a communication tool that is based on object-oriented message-passing technology [4]. Messaging systems typically enable applications to interact by posting and retrieving messages from local queues that are connected transparently across network nodes. This minimal architecture tends to push more complex control behaviors (e.g., coordinating sequences of interactions) into the applications themselves, which impacts their modularity, maintainability, and extensibility. The NetWorks! Messaging Facility (NMF) provides the customary messaging queues, queue management and network transport services across heterogeneous platforms. However, NetWorks! also incorporates active objects called *Agents*, which mediate interactions between applications and local NMFs and isolate any additional behaviors required for integration or distributed control.

Agents consist of object methods that contain: conventional C or C++ code; calls to the native APIs of local applications; and calls to the NetWorks! API library for creating, sending, and retrieving messages. The messaging API provides both blocking and non-blocking (asynchronous) communication models. A supporting Data Management System (DMS) provides an extensible, machine independent "neutral exchange" representation for translating messages across incompatible application data models. Applications initiate distributed interactions via simple messaging API calls to Agents. Agents can: (1) manipulate and forward messages from applications to other Agents via NMFs; (2) interact with applications by injecting or extracting data and commands; and (3) provide other dedicated services. In particular, Agents can integrate generic distributed control models for coordinating interactions among other Agents and applications. The following sections review the three Agent-based coordination services that are already implemented [5].

BROKERING DISTRIBUTED APPLICATION RESOURCES AND SERVICES

A request broker is a dedicated control mechanism that mediates interactions between client applications needing particular resources or problem-solving services and server applications capable of providing them [6]. Brokers free individual applications from the burden of maintaining information locally as to where and how to obtain services that they may require, such as database queries or particular planning and scheduling engines. Instead, all applications within a distributed system register the services they support, their locations, and their calling interfaces with the broker, which typically maintains this information in a directory or naming service. Client applications can then query the broker to find and request the desired interaction. The broker uses the naming service to relay requests from clients to the relevant server applications, retrieve responses, and relay them back to the client.

The NetWorks! Service Request Manager (SRM) consists of an Agent that integrates a dedicated request broker application. The SRM control model also incorporates a shared memory bulletin board structure, which applications can use to post or retrieve information of common utility. The SRM supports a high-level API that includes functions for: dynamically registering applications and services; requesting services; adding and deleting information items from the bulletin-board; and querying the services directory and bulletin-board to search for particular items of interest. The SRM API is built on top of lower level NetWorks! messaging and DMS APIs.

COORDINATING DISTRIBUTED WORKFLOW

One approach to automating sequences of activities such as end-to-end decision support for space missions is to establish directed, data-driven control links between the relevant applications. Distributing control logic in this manner is cumbersome to maintain and extend, particularly in systems that support many composite activities and that evolve through incremental additions of applications and services.

The NetWorks! Process Planner provides an alternative process-oriented model for distributed coordination, which consists of a high-level scripting language and a control Agent that executes scripts. This model alleviates the difficulties of highly distributed schemes by capturing the coordination logic for workflows in centralized, compact scripts that are easily maintained and extended. Individual script steps take the form of "atomic" requests for specific services or tasks, such as transferring data between two applications or scheduling ground operations for a Shuttle mission. The scripting language also incorporates control structures to represent data dependencies, temporal ordering, and other synchronization constraints across atomic script steps, including binding variables to store input arguments or results of script steps for later use, conditional branching, and iteration. Mutually independent tasks can be grouped explicitly for concurrent execution. A nested invocation primitive enables scripts to be embedded in other scripts.

Clients use a simple message-based API call to invoke the Process Planner to initiate a specified script. The Process Planner Agent incorporates a script interpreter engine that instantiates that script and executes its constituent steps in the specified sequence. For example, the mission support script would input specified mission profile parameters to an intelligent planning engine, transfer the results to a database, and so on. Script steps are executed by sending messages requesting the specified services to an associated SRM. The SRM forwards requests to the relevant servers, and relays responses back to the Process Planner, which then reiterates these behaviors, updating interim variables, testing control constraints, and requesting any script steps that are ready to be executed. Upon completing the script, the Agent returns results to the client, such as a verified mission schedule. In essence, the Process Planner functions as a workflow driver to the SRM, which brokers individual task requests. The two coordination engines act in tandem to support process-oriented interactions between applications.

BROKERING DECOMPOSABLE SERVICES

The SRM mediates interactions between clients and individual servers, while the Process Planner coordinates the execution of multiple, interdependent services. A third coordination service, called the Server Group, coordinates multiple, independent services, such as system-level planning tasks that decompose into subplanning tasks for independent subsystems, or decision support queries that reduce to subqueries to independent databases. The Server Group is implemented as a specialized subclass of the SRM Agent. The Server Group Agent inherits most of the SRM's control behavior, but selectively extends the SRM's service registration and request services. The registration extension enables developers to register composite services in the Server Group directory. A composite service entry contains pointers to functions for (1) decomposing that defined service into other concrete services that are registered with the Server Group, and (2) combining results for those services. In response to requests for a composite service, the Server Group uses these functions to transparently: decompose that service into constituent tasks; dispatch requests to the appropriate servers for concurrent execution; and collect and combine results from those servers into a single response for the client. Examples of combination functions to merge results include voting algorithms, logical union, intersection, and relational join operations.

CONCLUSIONS

The NetWorks! tool suite enables complex coordination behaviors to be modeled and executed external to independent decision support applications, through a supporting layered infrastructure for distributed computing. Current generic control services include request broker, workflow, and group-oriented coordination models. The resulting partitioning of application and distributed behaviors results in improved modularity, maintainability, and extensibility of individual applications, whether intelligent or conventional. The infrastructure is extensible at both the control service (i.e., Agent) and message-passing layers. Individual control services are also interoperable, which means that they can be combined much like building blocks to match application-specific coordination requirements. These tools are directly applicable to domains other than decision support, including operations support, process control, concurrent engineering, and office automation.

ACKNOWLEDGMENTS

NetWorks! technologies have been developed in part with funding from the Small Business Innovative Research Program by the U.S. Army (contract DAAB10-87-C-0053) and NASA (contracts NAS10-11606, NAS10-11882, NAS5-31920, and NAS8-399905). Kirsten Kissmeyer, Tom Trautz, and Craig Hughes contributed to the development of the Agent-based coordination models.

REFERENCES

[1] M. Fox and S. Smith. 1984. ISIS: A Knowledge-Based System for Factory Scheduling. *Expert Systems*. 1: 25-49.

[2] S. Minton. ed. 1993. *Machine Learning Methods for Planning*. San Mateo, CA.; Morgan Kaufmann.

[3] G. Coulouris and J. Dollimore. 1988. *Distributed Systems: Concepts and Design*. Reading, MA.; Addison-Wesley.

[4] R. M. Adler. 1992. Object-Oriented Tools for Distributed Computing. *NASA Proceedings for Technology 2002 Conference*, NASA CP-3189.

[5] R. M. Adler. 1993. *Distributed Coordination Models for Client-Server Computing*. Technical Report, Symbiotics, Inc. Cambridge, MA.

[6] Object Management Group and X/Open. 1991. The Common Object Request Broker: Architecture and Specification. OMG Document No. 91.12.1 (revision 1.1), OMG, Framingham, MA.

Session PS-MS

Planning and Scheduling Workshop: Mission Support

N95- 23750

Modeling Actions and Operations to Support Mission Preparation

Jane T. Malin
NASA Johnson Space Center
Automation & Robotics Div., MC ER2
Houston, TX 77058
Phone: (713) 483-2046 FAX: (713) 483-3204
malin@aio.jsc.nasa.gov

D. P. Ryan and D. L. Schreckenghost
Metrica, Inc.
NASA JSC, ER2
Houston, TX 77058
Phone: (713) 244-6135 FAX: (713) 483-3204
schreck@aio.jsc.nasa.gov

KEY WORDS AND PHRASES

Planning, action representation, modeling.

ABSTRACT

This paper describes two linked technology development projects to support Space Shuttle ground operations personnel, both during mission preparation analysis and related analyses in missions. The Space Propulsion Robust Analysis Tool (SPRAT) will provide intelligent support and automation for mission analysis set-up, interpretation, reporting and documentation. SPRAT models the actions taken by flight support personnel during mission preparation and uses this model to generate an action plan. CONFIG will provide intelligent automation for procedure analyses and failure impact analyses, by simulating the interactions between operations and systems with embedded failures. CONFIG models the actions taken by crew during space vehicle malfunctions and simulates how the planned action sequences in procedures affect a device model. Jointly the SPRAT and CONFIG projects provide an opportunity to investigate how the nature of a task affects the representation of actions, and to determine a more general action representation supporting a broad range of tasks. This paper describes the problems in representing actions for mission preparation and their relation to planning and scheduling.

INTRODUCTION

We are developing methods and tools to provide intelligent automation and support for mission preparation tasks. These require the representation of mission preparation actions, and this representation is affected by the nature of the task being performed. We are investigating action representations for two distinct types of tasks, propulsion (PROP) consumables analysis and operations procedures evaluation.

The consumables analyses conducted by PROP consumables officers are a complex, time-consuming mission analysis task. Throughout the year preceding a flight, several types of mission changes initiate new cycles of analysis to determine how these changes affect consumables. Iterative evaluations are needed for nominal and contingency situations and for proposed mission plans and objectives, priorities, flight rules and procedures. These mission and situation what-if analyses are used to determine impacts to mission objectives and procedures. During missions, additional analyses are performed as needed.

Procedure evaluation has similar characteristics. It is important both in impact analysis during missions when an anomaly causes space system reconfiguration, and in procedure development and analysis during mission preparation. Operations personnel evaluate procedures against nominal and contingency configurations, to assess which procedures will be impacted and which should be altered. When procedures are altered to fit the current mission configuration, they are again evaluated against the current mission situation and related "next-worst" contingency situations.

These mission preparation tasks have common characteristics and problems. They both involve action representations, but for two distinct types of tasks:

- Scientific and engineering analysis: data generation and interpretation to answer specific questions; e.g., consumables analysis (SPRAT)
- Device operation and process control: monitoring and control of physical devices and processes in operations to achieve specific behaviors and to respond to failures; e.g., procedure evaluation (CONFIG)

Action representations developed for scientific and engineering analysis include the task of developing scientific models [1] and data analysis tasks for geological exploration [4].

Representations developed for device operation and process control tasks include malfunction procedures and process control operations. In the SPRAT and CONFIG projects, we are developing technologies to address both types of tasks, with the goal of developing a more general action representation. A mutual benefit is being gained by deriving an action and procedure representation which embraces both types of task domains.

SPRAT

The goal of the SPRAT project is to provide advanced technology support for flight design personnel and flight controllers to use when conducting analyses prior to a mission, and when performing new analyses in response to anomalous situations that occur during a mission. Initially, the project is focused on tools that support the management of mission preparation actions (the flight controller mission analysis "procedures" performed pre-mission).

Mission preparation actions include the execution of simulation and analysis software, the interpretation of results from these computations, and the generation of mission preparation reports summarizing decisions. Action management consists of creating and modifying an action item list, tracking the outcome of actions on the list, and creating and modifying action descriptions and their relations.

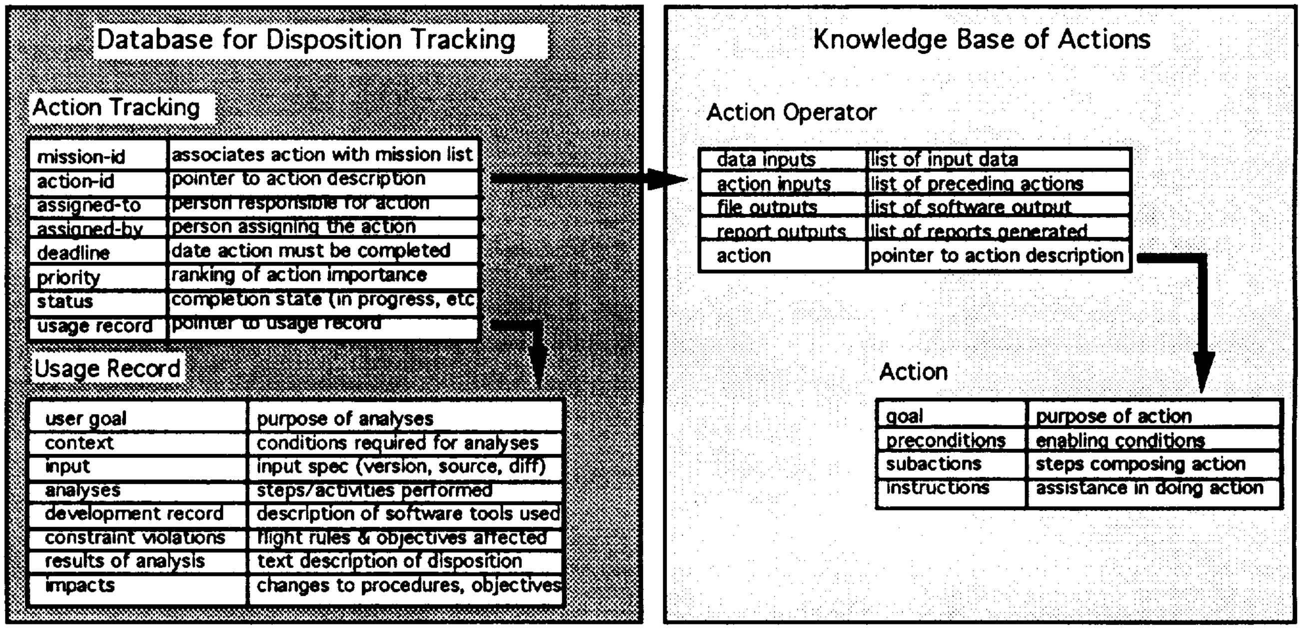
Action list creation can be viewed as form of planning, and action tracking as monitoring plan execution. A knowledge base of domain actions is defined in terms of goals and associated activities. Actions from this knowledge base are selected and placed on a managed list. The execution of actions on this list is monitored to determine how actions are dispositioned and to document the outcome of actions. This tracking information is stored in an action disposition "database". This separation of the knowledge base of available actions and the data base of action tracking objects permits multiple actions of the same type to be managed on one list. The Figure illustrates this distinction.

SPRAT's action representation has two parts:

- Description: goal of the action and conditions that must hold prior to action execution.
- Tracking Record: information about action assignment and disposition. The action tracking information is retained as part of a usage record stored in the action archive.

SPRAT provides for goal hierarchy and levels of abstraction in actions by permitting subactions (with subgoals) to be associated with an action. Subactions are viewed as constituent actions, or actions that must all be completed for a higher level goal to be satisfied.

SPRAT represents action dependencies in terms of inputs required by an action (data and information from other actions) and outputs generated by an action (software and manual). When a change in mission definition data occurs,



Database for Disposition Tracking

Action Tracking

mission-id	associates action with mission list
action-id	pointer to action description
assigned-to	person responsible for action
assigned-by	person assigning the action
deadline	date action must be completed
priority	ranking of action importance
status	completion state (in progress, etc
usage record	pointer to usage record

Usage Record

user goal	purpose of analyses
context	conditions required for analyses
input	input spec (version, source, diff)
analyses	steps/activities performed
development record	description of software tools used
constraint violations	flight rules & objectives affected
results of analysis	text description of disposition
impacts	changes to procedures, objectives

Knowledge Base of Actions

Action Operator

data inputs	list of input data
action inputs	list of preceding actions
file outputs	list of software output
report outputs	list of reports generated
action	pointer to action description

Action

goal	purpose of action
preconditions	enabling conditions
subactions	steps composing action
instructions	assistance in doing action

Figure. The SPRAT Action Representation

the dependency constraints are used to determine what new actions must be performed in response to this change. Simple ordering constraints are used to construct a list of actions. These constraints include delivery deadlines and priorities, and software precedence constraints.

The information needed to track the disposition of actions includes information about the intent of the action, the way the action was conducted, and the outcome of the action. The intent of the action is defined by the user's goal or purpose in performing the action and the mission context in which the action is relevant (e.g., rendezvous, mission definition data). The way the action is conducted is characterized by the activities/steps composing the action (e.g., analyses performed) and the characteristics of the tools used when performing these activities/steps (e.g., low fidelity model of gravity used). Information needed to track the action includes information about deadline, priority, completion status, and action responsibility.

The outcome of an action includes the results of the analyses (e.g., computation of consumables usage), the consistency of these results with flight rules and mission objectives, and the impact of these results on flight procedures. It also includes information about the execution of the action (was the action completed and how was it completed, was the action successful and if not why, why was the action canceled or aborted, what was done in response to an unsuccessful action). Information about analyses that were canceled or unsuccessful is useful, since knowledge about why an approach wasn't pursued or what caused it to fail may be useful when performing similar analyses in the future.

The SPRAT prototype is implemented in G2, extended with C routines for the data interface to analysis software. It runs on Unix workstations. Although the initial domain is consumables analysis, the SPRAT project will develop both flight-discipline-specific and generic tools for other disciplines to build similar systems.

CONFIG

CONFIG is a prototype software tool which provides integrated support for the modeling, simulation and analysis of the structure, behavior, failures, and operation of system designs [5,6]. System models are structures of connected component models, with embedded time-related behavior models partitioned into nominal and failure modes. The behavior of each device during a simulation depends on its current mode and on changes in its input caused by operations or from other devices via local connections or global flow path changes. These capabilities enable several types of evaluation of system operability, including analysis of impacts over time of faults, failures, and procedural or environmental difficulties.

CONFIG operations models support analysis of plans and procedures for operation of systems in nominal and contingency configurations. They can also support simulation and analysis of proposed changes (reconfigured systems and revised procedures) that are developed during operations in response to failures. The operations modeling approach integrates both with operations-execution-monitoring representations that are based on device and command states and with goal-based planning representations [3].

CONFIG operations models represent procedure actions and dependencies among these actions. CONFIG operations models are activity structure models that can be developed independently from system models, yet link and dynamically interact during simulation with system models. Activities are the basic components of a CONFIG operations model, and are connected together in action structures. These structures represent procedures or protocols that interact with the system, to control and use it to achieve goals or functions. Relations define sequencing and control between activities and connect devices with device-controlling activities.

CONFIG is implemented in the Common Lisp Object System (CLOS) language, and runs on Unix workstations. The current test model domain for CONFIG is thermal bus systems, including a model of a pump safing procedure.

PLANNING & SCHEDULING ISSUES

Action list management in SPRAT raises a number of issues related to both plan creation and plan repair. An objective of the SPRAT project is to provide a tool that permits the flight controller to create new actions dynamically, and to link those actions into the representation of precedence constraints. Such a capability minimizes domain knowledge engineering, since new actions can be added as needed. The ability for the user to create new types of actions (not yet developed) is related to the work by Martin and Firby [7] on human repair of robot plans "on the fly".

The dispositioning of actions on the action list includes archiving the outcome of actions for use in future missions. These action archives will be used as starting points for creating action lists for missions with similar issues and constraints. Accessing and modifying these archived actions remains an issue. These archived actions resemble case bases of partial plans [8].

An issue related to the disposition of actions is merging new items onto the list and deleting items on the list that no longer hold. To be added to the list, an action must be consistent with

- the mission definition (e.g., flight design data)
- the phase of the mission preparation
- the intent of the controller performing the analysis (e.g., orbital vs. ascent analyses)

As the mission definition and user intent change throughout mission preparation, actions items on the list may be no longer relevant (e.g., new flight design data). For SPRAT, the challenge is to provide an adaptable plan with a goal structure which models flight controller intent. The intent of an action is needed to track the action (did the action achieve the desired effect? was an observed change intended?), and to provide goals that can be manipulated using traditional replanning techniques [2,3].

Procedure modeling in CONFIG uses an action representation that interfaces with planning systems, and that will be able to use SPRAT-style action management. CONFIG and SPRAT action representations can become more powerful if action representations in planning and scheduling become integrated.

BENEFITS

SPRAT models the actions taken by flight support personnel during mission preparation. CONFIG models the actions taken by crew executing procedures. Jointly the SPRAT and CONFIG projects provide an opportunity to investigate how the nature of a task affects the representation of actions, and to determine a more general action representation supporting a broad range of tasks. Such representations can be applied to other types of activities (such as software development and analysis over large data bases). They also enable the development of more flexible tools for representing and reasoning about actions.

Application of CONFIG and SPRAT can reduce ground operations costs not only on console, but in a large and costly operations area, mission preparation. Increased automation and support for mission analysis and procedure analysis will reduce analysis time, make impact assessment quicker, reduce the number of unnecessary analyses, reduce training time and support better documentation. Common representations for procedures, action lists, plans and schedules can support the integration of several types of operations support tools.

ACKNOWLEDGMENTS

Thanks to JSC flight controller Matthew Barry, for his insight into mission preparation tasks and design concepts for support systems. Thanks to Tim Hill for his significant efforts in the design and development of the SPRAT action representation. Thanks to Land Fleming for CONFIG design and programming. Thanks to Barry Fox for his review comments.

REFERENCES

[1] Ellman, T., 1993. A toolbox and a record for scientific model development. *Proc. SOAR-93*, (NCP 3240). Houston, TX: NASA-JSC, 1993: 321-328.

[2], Elsaesser, C.; R. MacMillan., 1991. Representation and Algorithms for Multiagent Adversarial Planning. *MITRE Tech Report* 91W000207.

[3] Georgeff, M.; A. Lansky., 1987. Reactive reasoning and planning.*Proc AAAI*, 1987: 677-682.

[4] Kant, E., 1988. Interactive problem solving using task configuration and control.*IEEE Expert*, Winter 1988.

[5] Malin, J.; B.Basham; R. Harris., 1990. Use of qualitative models in discrete event simulation for analysis of malfunctions in continuous processing systems.*AI in Process Engineering*, M. Mavrovouniotis, ed., Academic Press: 37-79.

[6] Malin, J.; D. Ryan; L. Fleming., 1993. CONFIG - Integrated Engineering of Systems and their Operation, *Proc. 4th Nat'l Tech. Transfer Conf.*, Anaheim, CA, Forthcoming.

[7] Martin, C.; J. Firby., 1991. An integrated architecture for Planning and Learning. *SigArt Bulletin* 2(4): 125-129.

[8] Zito-Wolf, R.; R. Alterman., 1993. A framework and an analysis of current proposals for case-based organization and representation of procedural knowledge. *Proc AAAI*: 73-78.

CRI Planning and Scheduling for Space

N95-23751

Mads Aarup
CRI Space
Ground Based Systems Section
Denmark
tel. +45 45822100
fax +45 45813217
e-mail: maa@nov.cri.dk

Computer Resources International (CRI) has many years of experience in developing space planning and scheduling systems for the European Space Agency. Activities range from AIT/AIV planning over mission planning to research in on-board autonomy using advanced planning and scheduling technologies in conjunction with model-based diagnostics.

This article presents four projects carried out for ESA by CRI with various subcontractors:

- DI, Distributed Intelligence for Ground/Space Systems is an on-going research project,
- GMPT, Generic Mission Planning Toolset, a feasibility study concluded in 1993,
- OPTIMUM-AIV, Open Planning Tool for AIV, development of a knowledge-based AIV planning and scheduling tool ended in 1992,
- PlanERS-1, development of an AI and knowledge-based mission planning prototype for the ERS-1 earth observation spacecraft ended in 1991.

DISTRIBUTED INTELLIGENCE FOR GROUND/SPACE SYSTEMS

DI is short for Distributed Intelligence for Ground/Space Systems and the DI Study is one in a series of ESA projects concerned with the development of new concepts and architectures for future autonomous spacecraft systems. The kick-off of DI was in January 1994 and the planned duration is three years. The total budget is 600,000 ESA Accounting Units corresponding to approximately $720,000.

The background of DI is the desire to design future ground/space systems with a higher degree of autonomy than seen in today's missions. The aim of introducing autonomy in spacecraft systems is to:

- lift the role of the spacecraft operators from routine work and basic trouble shooting to supervision,
- ease access to and increase availability of spacecraft resources,
- carry out basic mission planning for users,
- enable missions which have not yet been feasible due to eg. propagation delays, insufficient ground station coverage etc,
- possibly reduce mission cost.

The study serves to identify the feasibility of using state-of-the-art technologies in the area of planning, scheduling, fault detection, model-based diagnosis and knowledge processing to obtain a higher level of autonomy in ground/space systems.

A demonstration of these technologies will be developed in the form of a prototype to run in a laboratory environment for the purpose of evaluating future ground/space system designs, and to experiment with the distribution of functionalities of the autonomous architecture between the ground and space segment. DI will use the ERS-1 earth observation mission as the reference mission for the study.

Reference Mission

Not all missions will benefit equally from AI and autonomy in space. AI is mainly applicable in complex domains where complicated decisions, based on several inputs have to be made. Autonomy on the other hand is beneficial in cases where human intervention is either inappropriate or directly impossible. Thus an interesting reference mission for this study should involve a complex spacecraft in an orbit that is either partly without ground contact or so distant that significant delays are inevitable. A natural choice is to select the ERS-1 mission as the reference since:

- ERS-1 is equipped with several scientific instruments with many operational constraints, implying very complex mission planning,
- ERS-1 is in a low polar orbit causing it to be out of ground contact during prolonged periods of time,
- operational experience has been gained, making it possible to quantify the advantages of on-board autonomy and AI,
- ERS-1 systems engineering expertise exists in the DI consortium,
- The ERS-1 simulator is available in the DI consortium.

Approach

The DI study is divided into two phases. In phase I, as a practical mean for obtaining a higher degree of abstraction, we have taken the rather provocative liberty to simply consider the ground and space segment as one combined system. This allows focusing on the essential user requirements on the overall system and on the interaction of the various modules of the autonomous ground/space system. Phase I creates a combined architecture that will be developed into the phase I prototype mock-up to ensure feasibility of integrating existing software developments.

In phase II the focus will be concentrated on the distribution aspects of the ground and space segments taking into account issues of distributed artificial intelligence. The development of the distributed phase II prototype will further improve the integrated software tools of the phase I prototype mock-up enabling the evaluation and demonstration of benefits.

The current status as of June 1994 is that a Draft User Requirements Document for the phase I prototype has been produced and the ERS-1 mission demonstration scenarios have been described. The prototype mock-up development has just begun with a clarification of the general MMI strategy.

GENERIC MISSION PLANNING TOOLSET

GMPT is a pilot study performed for ESA-ESOC, concerned with the development of a concept for a Generic Missions Planning Toolset in support of operations planning and scheduling. The main objectives are to provide a survey of general mission planning approaches, to define generic mission planning user requirements and standards, and to define a GMPT and develop a small prototype. The study was performed for ESA/ESOC with Computer Resources International A/S (Denmark) as prime contractor and with Science Systems Ltd (UK) as subcontractor. The kick-off was in January 1992 and the final presentation was held at ESOC in November 1992.

The study is divided into two phases with the following main objectives:

- provide a general survey of current mission planning approaches, analyze future ESA requirements for mission planning systems, define generic mission planning user requirements, as well as related interface standards,
- elaborate GMPT concepts, i.e. define baseline software requirements and overall architectural design, and develop a prototype demonstrating the feasibility of the elaborated concepts. As a bi-product, GMPT also resulted in a mission planning glossary list aiming at harmonizing the terminology used by the various mission planning teams.

Naturally there is a great variation in the extent to which elements in the mission planning process are generic, thus the GMPT modules have been categorized according to their degree of generality:

- *Fully generic* modules with no need for adaptation from mission to mission,
- *Configurable* modules, eg. a knowledge based system holding mission specific knowledge,
- *Mission specific* modules, eg. special external interface modules for translating non-standardized file formats to GMPT-compatible form.

The GMPT prototype was built around the OPTIMUM-AIV planning and scheduling tool previously developed for ESA, and aimed at demonstrating

- feasibility of implementing GMPT,
- schedule lifecycle under GMPT,
- efficient coding and decoding of state vectors,
- principle of activity-to-command conversion.

The GMPT is foreseen to operate in the ESOC spacecraft operations infrastructure. The current infrastructure is based on SCOS-I which will be replaced by SCOS-II after 1995. Therefore the architecture of the GMPT will depend on the structure of SCOS-II and the Mission Information Base defined within the framework of the SCOS-II project.

OPTIMUM-AIV

The size and complexity of the tasks involved in the Assembly, Integration and Verification (AIV) of a spacecraft, raises the need for efficient and flexible planning and scheduling tools. This lead ESA to award a contract to a consortium, with CRI as prime contractor together with Matra, AIAI and ProgEspace, to assess the applicability of AI and KBS techniques in a prototype AIV planning and scheduling tool. This study results in a set of user and software requirements and a demonstration system exploring some of the aspects of AIV planning.

The objectives of the OPTIMUM AIV project are four-fold:

- to develop an operational kernel of a planning, scheduling and plan repair tool consisting of a set of software functionalities for assistance in:
  - initial specification of AIV plans,
  - generation of valid plans and schedules for the various AIV activities,
  - interactive monitoring of the AIV plan execution,
  - identification of immediate effects and plan repair of problems,
- to embed external interfaces which allow integration with alternative scheduling systems and project databases,

- to provide facilities which will allow individual projects to customize the kernel to suit its specific needs,
- to implement knowledge-rich plan representations and enable the active use of this rich domain knowledge in the planning and scheduling process.

Artemis Interface

The system has embedded an interface to the widely used Artemis Project Management System. The interface is primarily intended for:

- importing space project data, i.e. activities and events, constraints, and resource data sets,
- exporting and displaying plans,
- report writing and graphics,
- aggregation, i.e. summarizing numeric data held in network data sets, e.g. resource requirements for all activities.

It can also be used for network construction, examination of the network logic, time analysis and updating, resource-limited or time-limited scheduling, and multiple network processing. However, in these latter uses of Artemis, it is not feasible to return the results directly to OPTIMUM-AIV.

Programming Interfaces

The system is designed as to allow external documentation programs to be written. It provides an interface that permits any user to develop their own documentation, in particular any new representation of the plan and schedule. That means that all activities, resources and constraints and any schedule will be accessible by any external program (written in C, Pascal or Ada).

Information can then be derived about alternative activities, soft constraints that may be relaxed, and potential activities that may be performed in advance.

PlanERS-1

The planERS-1 system was developed for ESTEC by a consortium primed by CRI together with Matra and AIAI, with the primary aim of assessing the applicability of knowledge-based and artificial intelligence techniques to planning and scheduling problems. The system was developed on SUN 3 workstations using CommonLisp and the Knowledge Engineering Environment (KEE). It has been used at ESTEC to evaluate the recorder and downlink strategy applied on the spacecraft.

Though conventional planning and scheduling systems have been used on a daily basis, they present various drawbacks. In general, these drawbacks appear because the scheduling domain is not static but evolves gradually; the cause may be the degradation of the spacecraft and its resources, changes to the satellite utilization, or increased demand for remote sensing data. One approach taken to handle these problems has been to manually pre-process the plan and over-constrain the input to the planning software. This, however, increases the risk of producing sub-optimal plans, and raises the question whether artificial intelligence technology can provide tools where the planning knowledge rather than the input data, is modified in order to reflect the changing environment of the satellite.

The planERS-1 system overcomes some of these problems by providing a flexible environment in which the modelling of the earth observation mission can be dynamically updated.

The planERS-1 System is a prototype system developed to assist payload planners of the ERS-1 satellite with constructing a Preferred Exploitation Plan (PEP) based on customer requests, background missions, and detailed modelling of the spacecraft.

BENEFITS OF ADVANCED SOFTWARE TECHNIQUES FOR MISSION PLANNING SYSTEMS

N95- 23752

A. Gasquet, Y. Parrod, A. De Saint Vincent

MATRA MARCONI SPACE
31 Rue des cosmonautes - Z.I. du Palays
31077 Toulouse Cedex - France

Tel +33 61.39.61.39
Fax +33 62.24.77.80
Email : gasquet@osiris.matra-espace.fr

KEY WORDS AND PHRASES

"Artificial Intelligence", "Mission Planning", "Flexibility, Genericity and Performance".

INTRODUCTION

The increasing complexity of modern spacecraft, and the stringent requirement for maximizing their mission return, call for a new generation of Mission Planning Systems (MPS). In this paper, we discuss the requirements for the Space Mission Planning and the benefits which can be expected from Artificial Intelligence techniques through examples of applications developed by Matra Marconi Space.

THE MISSION PLANNING PROBLEM

The term "Mission Planning" is used to refer to the process of planning and scheduling all activities and operations of the space segment (spacecraft platform and payload, e.g. power sub-system for the platform, optical instruments and tape recorder for the payload) and the ground segment (ground station activities, payload data processing and product dissemination) associated to a given mission.

The main inputs to the Mission Planning System are a set of requests of the following types :

- Spacecraft platform operation;
- End User request (e.g. observation requests for an Earth observation satellite);
- Other types of ground segment activities (e.g. data processing requests, dissemination requests).

The main outputs of the Mission Planning System are the *Service Utilization Plan* for satellite End Users, the *Final Operations Plan* uplinked to the space segment. Additional outputs include ground segments activities plans. From an operational point of view, the whole process is decomposed in the two following phases :

- Generation of the Operations plans: his phase is performed off-line and deals with the acquisition of User Requests and the detailed planning and scheduling of all space / ground operations. It includes :
  - The generation of the Preferred Exploitation Plan (PEP),
  - The integration of this first plan with the activities required by the Operations team for house keeping maneuvers, and the production of the final "executable" plan.
- Execution of the Operations plans : Once the whole planning and scheduling process has been completed, a schedule is available for execution and transmitted to the execution environment. During execution, monitoring is performed to control the evolution of the mission and detect eventual anomalies. If any disturbance on the current schedule occurs during its execution, rescheduling may be required and performed locally by the mission control center. If the rescheduling fails, a replanning session is entered on the Mission Planning System. Examples of anomalies include resource shortage (e.g. electrical power drop, unavailable ground station), activity execution failure (constraint violation, unexpected result), and changes in the satellite status due to some contingency (automatic or manual plan interruption, unexpected state transition).

THE MISSION PLANNING REQUIREMENTS

Based on experience learnt from past developments and current studies, both on operational Mission Planning systems and on advanced prototypes, three main types of requirements on the Mission Planning system can be identified.

Algorithmic performance

Generally, the Planning & Scheduling problem is characterized by an intrinsically high combinatorial complexity, reflecting the complexity of the spacecraft itself and the numerous utilization constraints (resource constraints, inter-instruments constraints, etc...). This is in particular the case for the first step of the Mission Planning process which deals with the definition of the PEP starting from a large number of End User requests. It is thus necessary to find powerful algorithmic techniques to deal appropriately with that complexity, in order to optimize as much as possible the utilization of the satellite, while taking into account the constraints on computing time.

Matra Marconi Space has conducted an internal study on this problem in order to evaluate the applicability of advanced algorithmic techniques on the planning & scheduling of an Earth Observation spacecraft . The objective was to optimize as much as possible the use of the satellite resources with an acceptable response time taking into account the following points :

- On one hand, the combinatorial problem due to the high number of requests to be scheduled makes the determination of a good solution difficult in a reasonable time (large space of potential solutions to be explored);
- On the other hand, the complexity of the spacecraft due to the management of tape recorders, the strategy used for ground station dump operations and the constraints imposed by the capabilities of the instruments in terms of transition between requests makes the determination of one feasible plan a time consuming step.

The activity performed in 1993-94 lead to the definition and implementation of a planning algorithm applied to the SPOT4 mission planning problem using an iterative and "any-time" optimization strategy [1]. This approach is characterized by two phases :

- Phase 1 : Determination of a first plan (without optimization) based on a simple heuristic strategy. This phase is considered as an initialization phase being responsible for the determination of a first potential solution.
- Phase 2 (The anytime phase) : The algorithm starts a loop which explores the initial plan elaborated in Phase 1 and then optimizes this plan. This operation is done by iteratively removing some requests and inserting new requests according to heuristics driving the plan evolution toward a better plan quality. In order to avoid looping in the remove / insert process, all generated plans (up to several thousands) are stored and each new plan is checked against the history of the already generated plans.

This algorithm was integrated into a mission simulator for analysis on real problems. Testing has been performed using operational scenarios and the analyses conducted during the testing phase have lead to the following conclusions :

- A first set of initial plans can be made available at the end of the first phase, in a very short time;
- Initial plans are improved regularly and solutions are available at any time (Several plans of approximately the same "quality" are available);
- The flexibility of the iterative approach allows late insertion into the plan of new requests, which is an important advantage from an operational point of view;

This approach thus proved to be quite successful; furthermore, it is general enough to be reusable for other planning and scheduling problems. Further developments in this area now concerns the application of these techniques to a new observation satellite.

Flexibility

The lifetime of modern spacecraft combined with the complexity of the current missions call for highly flexible and evolutive planning systems, enabling users to adapt the planning system to the evolutions of the planning problem (new planning constraints derived from satellite degradation, new planning strategies because of evolution of spacecraft utilization or increased planning experience, etc...). In conventional Mission Planning System, information is more or less hard-coded, making changes and corrections difficult. For instance, the evolutions of conceptual information concerning strategies for resolving conflicts cannot be modified by the operator and requires software modification. In order to solve this problem, Knowledge Based Systems (KBS) have a more declarative approach which brings a high degree of flexibility in the system.

An illustration of this approach is given by PlanErs [2]. PlanErs is a mission planning system developed by MMS (France), CRI (Denmark) and AIAI (University of Edimburgh) for the European Earth Resource Observation satellite ERS-1. It has been developed during an ESA R & D project from 1987 to 1990. Its first objective was the modeling of the planning & scheduling process in order to optimize various strategies (usage of recorder, record / dump strategy and selection of the ground station dedicated to the dump operation, priority mechanism between requests in order to cope resource shortage, etc). One of the main features of the system is the use of high level, user accessible formalisms for representing the different areas of the planning knowledge.

A simple example is the rule formalism used to define the transition modes for instrument: *From Mode Measurement_1 to Mode Measurement_2*

- *Goto Mode Standby _1 during 10 seconds*
- *Goto Mode Standby_2 during 20 seconds*
- *Goto Next_Mode*

Thanks to this approach, the PlanErs system has been used (in 1991-1992) by the European Space Agency (ESA) as a Mission Analysis tool for interactively simulating the impact of various strategies and constraints on the mission output of the satellite. PlanErs allowed to demonstrate a high potential in the adequation with the problem domain evolutivity by providing a very modular and declarative representation of the different types of knowledge involved in the scheduling problem, including for instance the possibility to account for evolutions in satellite utilization constraints, ground segment resources, tape recorder utilization strategies, etc.

PlanErs is going to be reused for the ERS-1 and ERS-2 mission analysis at ESA / ESRIN.

Genericity

The need to reduce mission-specific software development costs requires to develop Generic Mission Planning functions, from which a mission-specific Mission Planning system can be derived at low cost. In this case, the use of an object oriented representation for both the spacecraft model and the definition of the planning and scheduling methods participate to the genericity of the planning system by offering a more natural and reusable decomposition of the planning & scheduling world and of the methods governing the planning process.

This issue is addressed in the Generic Mission Planning Facilities (GMPF) project [3] which is currently performed by Cray Systems (UK) and Matra Marconi Space (France) for the European Space Agency (ESA/ESOC). The objective of this project is to analyze the commonalities between the large variety of Mission Planning Systems dedicated to specific missions and, by identifying the plan elements and the planning and scheduling process required by several types of mission, to define a common planning & scheduling kernel which can be customized to a given application. The GMPF project should contribute to the definition of the new generation of Spacecraft Control Center (SCOS II) which is conducted by ESA / ESOC.

The envisaged types of missions to be supported by GMPF are :

- Observatory Missions: The spacecraft has one main instrument. End Users are allocated observing time windows during which they have dedicated usage of the instrument.
- Survey Missions: The spacecraft has a single or a small number of payloads. The spacecraft and payload are normally operated by a centralized agency on behalf of a number of End Users who request specific observations that are planned according a high level mission definition.
- Multi-Instrument Missions: The spacecraft has a number of independent experiments, each provided by a separate Principal Investigator (PI). The platform is operated by a centralized agency but PIs are responsible for operation of their experiments, submitting requests to the control center.

- Telecommunication Missions: The spacecraft has a number of transponders to provide communications between ground stations (fixed service) or between another spacecraft and ground (data relay service). The spacecraft and its payload are operated by a centralized agency on behalf of the End Users. Transponders communication channels are allocated to Users.

The result of the GMPF study will be the definition and prototyping of :

- an objects library defining all the planning & scheduling elements and methods. These objects can be later reused or customized (by subclassing) for a specific application.
- a set of tools used to customize the library for a given application. These tools include a User Interface Builder, a Class Library Browser, a Mission Specific Information Editor and a Rule / Constraint Editor

At the current stage, the definition of the requirements for the GMPF tool kit has been performed. The project will lead to the implementation of those facilities and to a first application demonstrator.

CONCLUSIONS

In this paper, we have presented three main areas where advanced software techniques can contribute to solve the requirements raised by Mission Planning systems : performance, flexibility and genericity. These issues are taking an increasing importance with the growing complexity of space systems.

REFERENCES

[1] A. Gasquet : "*AI embedded in CPH*" Final Report, February 1994 (internal document), Matra Marconi Space.

[2] J. Fuchs, A. Gasquet, Jean Muller, Pierre Viau : "*PlanErs : Experimentation and Users Feedback*", Artificial Intelligence and Knowledge-based systems for space, May 1991, ESTEC, Noordwijk, The Netherlands.

[3] A. Gasquet : "*GMPF Users Requirements Document*", June 1994 (internal document), Matra Marconi Space.

A Scheduling and Diagnostic System for Scientific Satellite "GEOTAIL" Using Expert System

I.Nakatani*, M.Hashimoto*, T.Mukai*, T.Obara* and N.Nishigori**
* Institute of Space and Astronautical Science(ISAS)
3-1-1, Yoshinodai, Sagamihara-shi, Kanagawa 299, Japan
Tel/Fax: 81(427)51-3988
E-Mail Address: nakatani@ginga.eng.isas.ac.jp
** Fujitsu LTD.
9-3, Nakase 1-chome, Mihama-ku, Chiba 261, Japan

KEY WORDS AND PHRASES

"Scheduling satellite operation, diagnosing satellite troubles, actual scientific satellite operation"

ABSTRACT

The Intelligent Satellite Control Software (ISACS) for the geomagnetic tail observation satellite named GEOTAIL (launched in July 1992) has been successfully developed. ISACS has made it possible by applying Artificial Intelligence(AI) technology including an expert system to autonomously generate a tracking schedule, which originally used to be conducted manually. Using ISACS, a satellite operator can generate a maximum four day period of stored command stream autonomously and can easily confirm its safety. The ISACS system has another function -- to diagnose satellite troubles and to suggest necessary remedies. The workload of satellite operators has drastically been reduced since ISACS has been introduced into the operations of GEOTAIL.

INTRODUCTION

ISAS (Institute of Space and Astronautical Science) is receiving telemetry data from satellites and spacecraft on a daily basis. In recent years, satellite size has increased significantly and the mission objectives have expanded rapidly resulting in much more complex satellite functions. Therefore, ground system operators are required to have increasingly complicated and high-level knowledge of the satellite system. Moreover, it is becoming more difficult to keep as many high-level operators in the steady state phase of spacecraft operations as in the initial operation phase.

We have developed an artificial intelligence application software system for satellite monitoring and controlling on the ground to reduce the operators' workload by simplifying satellite operation and increasing reliability in satellite maintenance. Called ISACS, which stands for Intelligent Satellite Control Software, this system has been applied to the GEOTAIL satellite launched in July 1992.

Many reports on the application of expert systems to satellite operation have been published. However, most of them are just ideas or prototype systems needing verification. ISACS is one of the few instances where expert systems have successfully been applied to actual scientific satellite operation.

ROLES OF ISACS IN GEOTAIL CONTROLLING

GEOTAIL is a joint project between Japan and the USA and aims at the study of the geomagnetic tail region of the magnetosphere. This satellite is the largest and most complicated one that ISAS has ever launched and with many onboard scientific instruments. GEOTAIL is tracked using a 64 mø parabolic antenna at Usuda Deep Space Center (UDSC) in Japan, and is remotely controlled from Sagamihara Spacecraft Operation Center (SSOC) at ISAS. Three NASA stations are also used to receive the recorded data.

In these circumstances, GEOTAIL operators are required to have a wider variety of expert knowledge to monitor and control the satellite than those of other satellites that ISAS has launched. Furthermore, GEOTAIL is tracked for eight hours every night in real time because its purpose is to observe the night side of the magnetosphere. However, resources for night-shift operations at ISAS are limited.

For these reasons, it has been requested that the satellite be safely controlled by a small number of operators by applying AI technologies.

ISACS has following functions:

(1) ISACS-PLANNER (ISACS-PLN)

ISACS receives the tracking schedule from abroad, observation requests from both home and abroad, and orbit and attitude data through an online data feed system, then it generates the operation schedule autonomously. ISACS checks the safety of this schedule, and converts it to command codes.

(2) ISACS-DOCTOR (ISACS-DOC)

ISACS reads the telemetry data sent from GEOTAIL and watches the status of the satellite in real time. If the operator finds the satellite in trouble, ISACS supports the operator in diagnosing the problem and taking the necessary actions.

Expert systems have been applied to the generation of the satellite operation plan and also to the satellite diagnostic system.

AIMS OF ISACS-PLN

We can define 'the scheduling of the satellite operation' as follows:

> *"To schedule satellite operation is*
> *to put 'the requirements' in order*
> *under 'the restrictions'."*

'The requirements' and 'the restrictions' are classified as follows:

Restrictions:
- Orbit and attitude of the satellite,
- Time and duration of eclipses in which the satellite is shadowed by the Earth or the Moon,
- Communication link margin between ground stations and the satellite,
- Power consumption and thermal condition,
- Tracking schedule for each ground station,
- Requirement for range and range rate(R&RR) measurement,
- Requirement for maneuver operations,
- Priority of operations,
- Operations inhibited for the safety of the satellite, and
- List of command codes corresponding to the operations.

Requirements:
- Power on/off of the instruments,
- Observation mode,
- Rewriting of Random Access Memory (RAM),
- Bit rate of telemetry data,
- Ground station antenna selection, and
- Tracking schedule for each ground station.

It is easy to update the application software when the operational condition or mode is changed if the restrictions are defined in the knowledge base as logic or parameters and the requirements from each scientist are input from independent data files.

Basically, the GEOTAIL satellite is controlled and operated by an Operation Program (OP) which consists of a stream of stored commands. The OP commands are autonomously executed during invisible time from UDSC. Once an OP is transmitted to the satellite from SSOC via the 64 m0 antenna at UDSC, GEOTAIL is operated autonomously for three or four days. An OP sequence consists of 128 control elements called Organized Command (OG) to govern, for example, the record (REC)/replay (REP) cycles of the data recorders(DRs), the pointing of the high gain mechanical despun antenna to the ground tracking stations, and the control of the scientific instruments according to their observation plans.

It would be a heavy load for the operators at SSOC if they had to manually generate the OP because the restrictions and the requirements as shown above should be considered for scheduling the satellite operation. To overcome the difficulties of carrying out such complicated mission operations, and to safely and reliably generate the operation program, we have developed ISACS-PLN by applying AI technology.

OUTLINE OF ISACS-PLN

ISACS-PLN has been developed on a Sun Work Station using a scheduling expert tool. The function of this scheduling expert tool is to support a programmer in constructing a knowledge base using the Black Board (BB) model based on object-oriented programming. Figure 1 shows the system configuration.

This system has three major functions: initialization, inference engine, and command checking.

Following is their outline.

Initialization

To generate an OP for GEOTAIL, the ISACS-PLN needs the following data:
- Orbit and attitude data,
- Tracking schedule of DSN stations for receiving DR's playback data, which is provided by Jet Propulsion Laboratory (JPL),
- Tracking schedule of UDSC and SSOC, and
- Operation requests for onboard subsystems.

Though some of these data must be manually input at SSOC, most data come into the Work Station through the network and are input to ISACS-PLN. The operation requests for onboard subsystems are written in a simple computer language called ORL (Operation Request Language). Using ORL, the scientists both at home and abroad can give the operation requests freely without worrying about restrictions such as the difference in time, the place where they are, or the period of request time.

Inference engine

The inference engine part is developed using an expert tool. The knowledge base is described using the frame (~50), the data-class (~1500) and the BB's. The data-class is used for defining the inhibited operation mode and command code table. The BB's are used for both adjusting the events and converting the status data. Special events, difficult to describe using prepared functions of the expert tool, are described by CLOS.

The inference engine has three parts -- input processing, schedule processing, and output processing as follows:

<u>Input processing</u>. After setting time parameters such as start and end times for scheduling on a time control table, the instances concerning the following items are generated in the request lists.

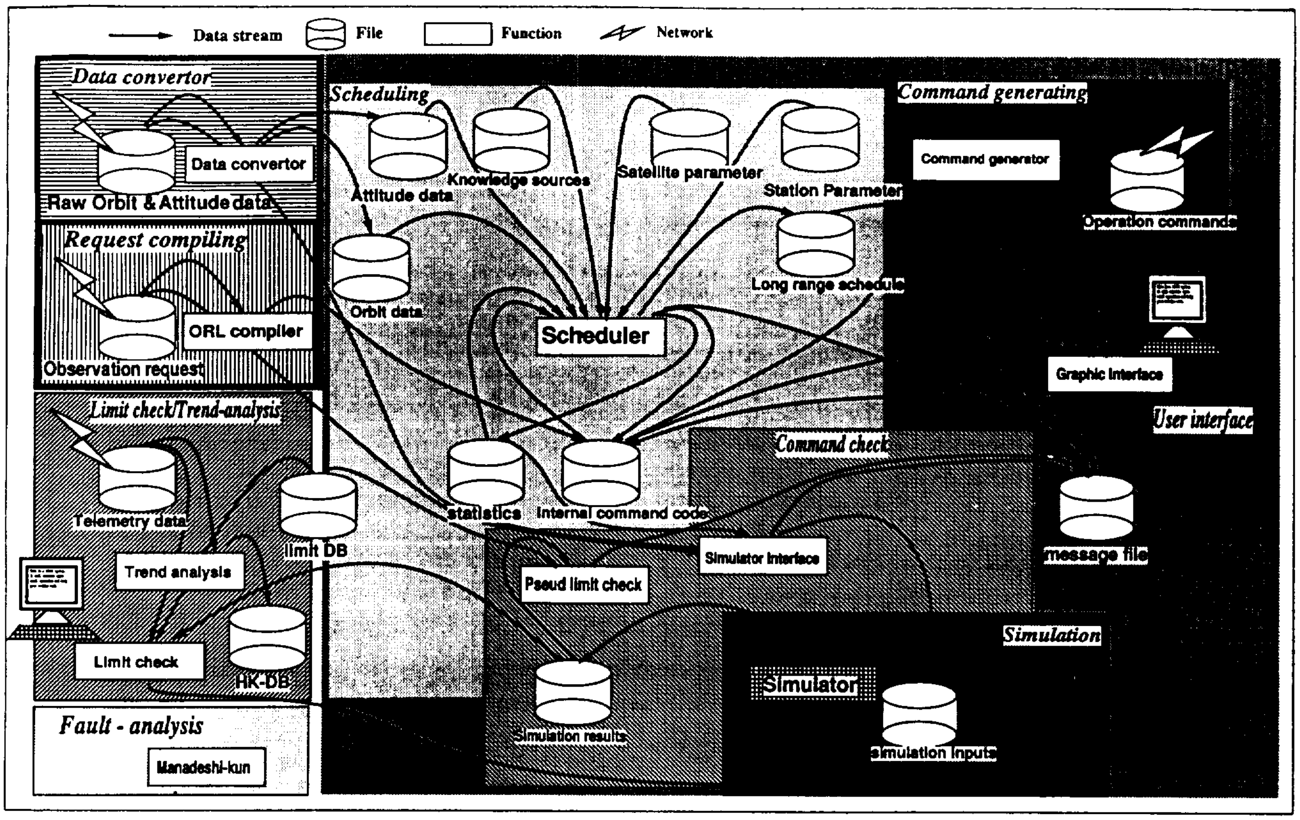


Figure 1 ISACS System Structure.

1) Input of the request file written in ORL.

The correspondent request items are cut out from the request file sent from each scientist according to the period of scheduling time.

2) Generation of the record cycle of the data recorders.

All the data obtained by scientific instruments onboard GEOTAIL is to be recorded by only one of two data recorders (DRs). In this function, the event-data for alternating the DR usage is generated by referring to the parameter data such as capacity of the DRs, overlap time for switching the DRs, and initial status of the DRs.

3) Generation of the tracking schedule of DSN stations.

The event data for receiving the DR's playback data or measuring R&RR data at the DSN stations is generated referring to the schedule data that have been planned at JPL based on the orbit data previously provided by ISAS and the initial status of GEOTAIL.

4) Generation of the standby sequence of UDSC station.

The command sequence required to be sent to the satellite at the beginning of every tracking pass is generated autonomously.

5) Generation of the control schedule of communication system.

The most suitable onboard antenna which provides enough link margin is selected from orbit information.

6) Generation of the switching cycle of the onboard heater system.

Power on or off schedule of the onboard heater system is generated.

Schedule processing. The schedule processing part has the following functions to adjust the requests.

1) The request list that has been generated in the input process is adjusted by considering the time control table data and priority of the command executions, and then the time series status list is generated.

2) In order to check the command sequence, the status list is searched for contradictions and prohibited command orders.

3) The request data that were canceled, due to conflict of time with other request data, are shifted within their permissible time span.

4) If ISACS-PLN cannot find a solution through adjusting request data, ISACS-PLN outputs an error message and entrusts the decision of which or what request should be selected to the satellite operator, since subjective criteria, such as importance of the observation or academic interest, can only be evaluated by scientists.

Output processing. A preliminary operation plan is thus generated in the inference part and then the list of command sequence is output.

Following is an example of how ISACS-PLN includes the preference of each scientists:

Figure 2-1 shows the time chart of the request data and figure 2-2 shows the result generated by ISACS-PLN autonomously. A, B, or C in both figures mean a subsystem onboard the satellite and each has an independent operation schedule. The numbers with parentheses indicate the priority of the observation previously defined in the knowledge

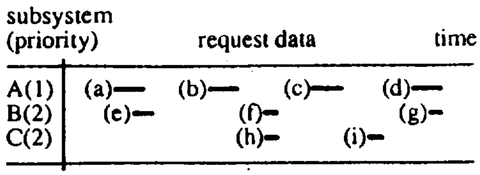


Figure 2-1 Input data to ISACS-PLN

base. The small letters in parentheses indicate the operation mode which has a series of commands, and the length of the line is proportionate to the time scale.

<Explanation of request data in figure 2-1>
Request from A: (a)(b)(c)(d) are operation requests of the Common Instruments (CIs) and their priority of execution is primary (=1).
Request from B: (e)(f)(g) are operation requests of the Physical Instruments (PIs) and their priority of execution is secondary (=2). The execution time of (e) has a scope, and the execution time of (g) is limited.
Request from C: (h)(i) are operation requests of the Physical Instruments (PIs) and their priority of execution is secondary (same as B).

ISACS-PLN generates the following results.

<Explanation of results in figure 2-2>
Output to A: All requests from A are accepted.
Output to B: The execution time of (e) is shifted back within its scope because a part of request (a) has a higher priority than (e) and conflicts with (e). Request (g) is canceled because (g) conflicts with (d) and the execution time of (g) is limited.
Output to C: Request (h) is canceled because the priority of (h) is same as (f) and (h) was put in to ISACS-PLN later than (f).

As mentioned above, ISACS-PLN does not fix the execution order among conflicting operation requests with the same priority because the rule of adjustment is affected by elements such as the importance of the observation or academic interest of scientists.

Command checking

The operation plan can be applied to the command checking function to simulate the temperatures, power consumption, and communication status of the satellite using mathematical modeling programs as follows:

1) Power analysis program.

The power consumption and remaining battery capacity are estimated from the power generated by solar cells and the load current predicted by the operation plan.

2) Thermal analysis program.

The temperature of each subsystem is predicted by the thermal analysis program.

3) Communication analysis program.

The antenna gain and span loss are evaluated from the satellite status estimated in the operation plan and orbit/attitude data. Then the receiving level, link margin, and C/N ratio are estimated.

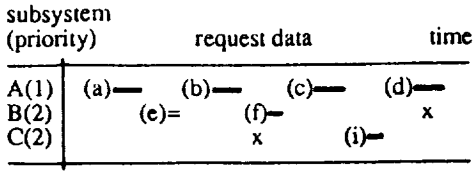


Figure 2-2 Output data from ISACS-PLN

OPERATION RESULTS

For the ISAS satellites launched before GEOTAIL, it took almost one day to manually generate an operation plan by adjusting the requirements of satellite operation using telephone or facsimile.

Now ISACS-PLN has shortened the processing time for operation plan generation to less than two hours. Moreover, ISACS-DOC can analyze the satellite status quickly using about 500 diagnosing rules defined in the knowledge base. About 80% of necessary information for diagnosis is fed on-line in real time. And, the scientists can send their observation requests in text file format through a network from home or abroad without concern for time limits. ISACS-DOC is very useful in protecting against overlooking satellite abnormality by checking the entire satellite's condition at least once every tracking pass.

CONCLUSION

From the viewpoint of AI technology, it is hard to say if the technique used in constructing the inference part of ISACS-PLN takes full advantage of AI technology, but this is not from a lack of skill in developing an expert system. We would like to emphasize that we have successfully merged, for the operation of a scientific satellite, 'the expert system' and 'the preference of scientists', so one is not emphasized over the other. This is a point much appreciated by both scientists and satellite operators.

REFERENCES

[1]Ramsay, B.H., "Anomalous Event Diagnosis for Environmental Satellite System", Proceedings of the 1993 Goddard Conference on Space Applications of Artificial Intelligence, pp.107-116, May 1993.

[2]Hughes, P.M., & Luczak, E.C., "The Generic Spacecraft Analyst Assistant (GenSAA): A Tool for Automating Spacecraft Monitoring with Expert Systems", Proceedings of the 1991 Goddard Conference on Space Applications of Artificial Intelligence, pp.129-139, May 1991.

[3]Ringer, M.J., Quinn, T.M., & Merolla, A., "Autonomous Power System Intelligent Diagnosis and Control", Proceedings of the 1991 Goddard Conference on Space Applications of Artificial Intelligence, pp.163-167, May 1991.

N95-23754

Automatic Commanding of the Mars Observer Camera

Michael Caplinger
Malin Space Science Systems, Inc.
PO Box 910148
San Diego CA 92191-0148, USA
phone: (619) 552-2650 fax: (619) 458-0503
mc@msss.com

KEY WORDS AND PHRASES

Conflict resolution, spacecraft imaging instrument operations, mission planning, sequencing and scheduling.

INTRODUCTION

Mars Observer, launched in September 1992, was intended to be a "survey-type" mission that acquired global coverage of Mars from a low, circular, near-polar orbit during an entire martian year [1]. As such, most of its instruments had fixed data rates, wide fields of view, and relatively low resolution, with fairly limited requirements for commanding. An exception is the Mars Observer Camera, or MOC. The MOC consists of a two-color Wide Angle (WA) system that can acquire both global images at low resolution (7.5 km/pixel) and regional images at commandable resolutions up to 250 m/pixel. Complementing the WA is the Narrow Angle (NA) system, that can acquire images at 8 resolutions from 12 m/pixel to 1.5 m/pixel, with a maximum crosstrack dimension of 3 km. The MOC also provides various forms of data compression (both lossless and lossy), and is designed to work at data rates from 700 bits per second (bps) to over 80k bps [2].

Because of this flexibility, developing MOC command sequences is much more difficult than the routine mode-changing that characterizes other instrument operations. Although the MOC cannot be pointed (the spacecraft is fixed nadir-pointing and has no scan platform), the timing, downlink stream allocation, compression type and parameters, and image dimensions of each image must be commanded from the ground, subject to the constraints inherent in the MOC and the spacecraft. To minimize the need for a large operations staff, the entire command generation process has been automated within the MOC Ground Data System [3].

Following the loss of the Mars Observer spacecraft in August 1993, NASA intends to launch a new spacecraft, Mars Global Surveyor (MGS), in late 1996. This spacecraft will carry the MOC flight spare (MOC 2). The MOC 2 operations plan will be largely identical to that developed for MOC, and all of the algorithms described here are applicable to it.

TARGET GENERATION

In advance, users define "time-independent observing plans" that consist of a specification of an area or feature edge on the surface of Mars, the type of acquisition(s) to be made there, any geometric and timing constraints (such as lighting angles, season, etc.), the image size, resolution, allowable compression types, and a single number indicating the priority of the observation. (By convention, priorities are non-negative and the more important an observation is, the larger its priority number.)

Daily during operations, these plans and spacecraft position are examined and a list of potential images is generated. This "strawman sequence" consists of image acquisition commands to be sent to the MOC; each command specifies the time a specific optical system is to be activated, and a set of parameters (image size, resolution, compression type, and downlink channel assignment) to be associated with that particular acquisition.

Since there are typically many thousands of active observing plans, and targeting is performed frequently, the algorithm that generates the strawman sequence must be very efficient. The algorithm we finally used treats NA and WA swaths in two different ways. Such a dual approach makes sense, since the clocking rates, and hence the accuracy requirements, are two orders of magnitude different between NA and

WA. Also, the very narrow field of view of the NA allows a much simpler geometric description of its swath to be used.

Narrow Angle

The NA swath is treated as a widthless polyline generated by sampling the ground track in equal time intervals and assuming linearity in lat/lon space between these points. The required accuracy is obtainable with fixed 5-second spacing, though a method using variable spacing, with more time resolution near the poles, is preferable and would be required for non-circular orbits.

The core of the algorithm is a loop over each segment of the ground track. A clip test between this line and the target box is done in lat/lon space, and the locations of the intersection(s), if any, are calculated, using the parametric representation of the line segment. These parameters are used to compute the first and last times of intersection, and the loop gathers the minimum and maximum time values. These values are used to generate the start time and dimensions of the image event, if there were any intersections. If there are no intersections, this area is not accessible on this orbit.

Because the majority of boxes on a given run are probably not accessible by the NA, some attention was paid to rejecting a line segment that did not intersect the box as soon as possible in the algorithm, using a trivial bounding-box calculation.

Wide Angle

The WA swath covers over 30 degrees of longitude and obviously cannot be treated as a widthless line. Instead, two ground track polylines are calculated -- one representing the maximum view angle of the WA in the +Y direction (plus) and the other in the -Y direction (minus). These curves represent the overall field of view of the MOC. Note that during the ascending part of the orbit, plus is east of minus; during the descending part, minus is east of plus. This ordering is used to remove the meridian-crossing ambiguity.

Rather than process the entire swath as a single polygon, it is broken up into four-sided quadrilaterals, called "quads", with sides connecting the four points defined by the plus and minus tracks at time t and t+delta. (Note that the lines connecting the plus and minus tracks at the same time are not the tracks followed by single scanlines, except near the equator, and they are not used as approximations to scanlines -- they are merely arbitrary lines.) Because quads are always convex, processing them is relatively simple.

The basic algorithm is a loop through all of the quads for a given orbit. At each point, the quad is tested against a target box. Two kinds of tests are performed and point coordinates are recorded. First, any box corner contained within the quad is recorded. Second, any point of intersection between the plus and minus edges of the quad with the box are recorded.

After all the quads are compared against a given box, the gathered test points are mapped to times and WA pixel coordinates using a separate iterative algorithm. (If a box generated no test points with the swath, it cannot be viewed on that orbit.) The ranges of the test point pixels and times are recorded and used to generate the timing and dimensions for a WA event.

The algorithm described so far ignores latitudes near the pole, because quads do not appear above a given critical latitude. (For example, suppose a quad had its bottom edge at latitude 85, and then the spacecraft passed over the pole and back down to latitude 85 before the quad's top edge was created. Then the quad would appear to have no latitude extent above latitude 85.) The polar areas can be handled by noting that if an orbit changes from ascending to descending or vice versa near a pole, then all boxes above a certain latitude are seen on that pass. In addition, if an orbit passes sufficiently close to a pole, then all boxes above a given latitude in a range of 180 degrees of longitude are also seen. Each pole can fall into at most one of these two categories (it is physically impossible to go from ascending to descending and back on the same orbit because the placement of the terminator cannot change that rapidly.) Thus, every vertex of a box that occurs in one of these polar regions is added to the list of test points.

Performance

Our initial performance goal was to process 3000 potential target areas for one orbit in less than 5 minutes. Operationally, we saw an average time of 2.1 minutes for this task on the Sun SPARCstation IPX (a roughly 20

SPECmark system), so we have exceeded our goal by over a factor of two. However, since we were processing a significantly larger task (13 orbits and nearly 10,000 plans) total time was about 1.5 hours, which can become burdensome.

Many solutions are possible without changing the algorithm, the simplest of which is to use a faster processor. Also, the algorithm is easily done in parallel either by splitting plans or orbits across processors, and so would benefit from the multiprocessor systems becoming available. We predict that a four-processor SPARCstation 10 system could perform the task above in less than 15 minutes.

CONFLICT RESOLUTION

Unfortunately, not all of the commands in a strawman sequence can be executed because of limited instrument resources. These resources include buffer space, CPU processing time, downlink rate, and power. Since the timing for each acquisition is fixed by the spacecraft's position, the sequence cannot be reordered. (This makes the MOC sequencing problem fundamentally different from other space application sequencing problems, such as Voyager-like or Hubble Space Telescope sequencing [4,5].) The only free parameters left to modify are whether or not to acquire each potential image, and the compression type and downlink channel assignment for each image. (While it is possible for science users to restrict compression or downlink channel to particular values, this may place limits on how well the automatic process can optimize the overall sequence. In some cases, resolution or image size can be altered as well, but the automatic program does not attempt such modifications.)

Thus, the MOC sequencing program seeks to maximize the number of images taken from the input sequence, while choosing images of higher priority, all other things being equal.

Obviously, the key to solving the problem is to generate alternative possible sequences and see which have conflicts. A critical problem is how to know if a given MOC sequence fits within the resource constraints. The solution is a fast event-driven simulator that mimics the behavior of the instrument and detects resource conflicts. Using this simulator as a black box, it can be determined if a given sequence is conflict-free, and if not, when and what the conflict is.

Additionally, we have the following desires for the algorithm:

(1) it should be applicable across all data rates (data rate assignments have changed several times and can be expected to change again)

(2) it should be insensitive to exact details of instrument behavior (during development, the performance and details of instrument operation were not known to any accuracy)

(3) it should allow "splicing" of daily sequences because planning is done piecemeal, not all at once

Our initial approach was to develop a series of heuristics that took a full input sequence and deleted or modified individual items until the sequence was without conflict. Though this worked after a fashion, it was extremely slow, because there was no systematic way to search for alternatives. Therefore, it was decided to invert the approach and develop a series of heuristics to take an initially empty input sequence and add items to it until no more can be added.

By "heuristic", we mean a rule intended to choose a favorable outcome without any analytical evidence that such an outcome would be chosen. One could have very specific heuristics, such as "when the data rate is higher than X, use predictive compression", or quite general heuristics, such as "choose the alternative such that the image is resident in the buffer for the shortest period of time." The more general heuristics are preferable, since they rely on less knowledge of the specifics of the process. In addition, specific heuristics may be derivable from the general heuristics, such that a system using only the general heuristic will appear to be operating under the specific heuristics as well.

In fact, we have obtained good results with a single heuristic, which we call "shortest-residence-time". This is used by the following algorithm:

```
sequence = $\varepsilon$ (empty sequence)
for priority = highest to lowest
 for images at this priority ordered by time,
 earliest to latest
  for each alternative
   given sequence so far, compute residence time
   of current image in instrument for this
   alternative. If alternative generates conflict,
   set time to $\infty$
```

if any time is not ∞, add this image, using the shortest-residence-time alternative, to sequence

The residence time of an image is the amount of time any fraction of either raw image data or any compressed or processed version of that data is stored in the MOC buffer. The alternatives examined by the program currently are from the set {predictive compression, channel 1; transform compression, channel 1; predictive compression, channel 2; transform compression, channel 2}; obviously, other alternatives could be easily added.

Performance

The requirement set for the MOC GDS was that conflict resolution for a 12-orbit strawman sequence containing 1500 potential acquisitions could be performed in less than 5 minutes; the existing system meets this performance goal on a Sun SPARCstation 1 (a roughly 10 SPECmark system.)

To give an idea of the size of a typical problem and the effectiveness of our algorithm, our standard test target set contains about 2500 planned areas. For a twelve-orbit period chosen at random, 111 images (50 WA and 61 NA images) were found to be accessible. At low data rate, 29 of the 111 images could be taken; at high data rate with 4 orbits of realtime passes, 75 of the images could be taken.

The relationship between the sequences found by our software and optimal sequences is not known, although the general problem has been shown to be NP-complete, meaning that the optimal sequence cannot be found without examining *all* sequences. We do note that our sequences utilize 90% or more of the available resources, indicating that little waste is present. For small sequences (about length 10) for which the optimal sequence could be found, our algorithm either finds the optimal sequence or at worst, fails to take one image.

CONCLUSIONS

The existing system is operational and has processed hundreds of simulated sequences that were then executed on the actual hardware (in ground testing) without conflict. We hope to use this system for instrument operations when Mars Global Surveyor goes into orbit around Mars in late 1997.

Some simple additions would make it possible to extend this system to missions in eccentric orbits, such as the "transition orbit" of MGS. These additions include the provision of a resolution requirement for time-independent plans, and removal of the reliance on geometric properties of the ground track for simplification of the targeting algorithm. While these additions would not completely solve the problem for missions which use a scan platform or spacecraft slewing to point their instruments, we believe this framework would be easily applicable even to such missions, by using a series of heuristics and resolution requirements to fix observations in time. We expect to experiment with this approach soon.

ACKNOWLEDGEMENTS

I especially thank Mike Malin, the MOC Principal Investigator, without whom the MOC would have never existed. Marc Sarrel implemented the first version of the targeting algorithm; Jeff Warren provided valuable insight into aspects of MOC geometry. This work was supported by the Jet Propulsion Laboratory, Contract 959060 to Malin Space Science Systems, for development and operation of the Mars Observer Camera.

REFERENCES

[1] Albee, A.L., *et al*, 1992. Mars Observer Mission. *Journal of Geophysical Research* 97(E5):7665-7680.

[2] Malin, M.C., *et al*, 1991. Design and Development of the Mars Observer Camera. *International Journal of Imaging Systems and Technology* 3(1):76-91.

[3] Caplinger, M., 1993. The Mars Observer Camera Ground Data System. In *Proceedings of the Ninth AIAA Conference on Computers in Aerospace*, San Diego, CA.

[4] Dias, W.C., *et al*, 1987. PLAN-IT: Scheduling Assistant for Solar System Exploration. *Telematics and Informatics* 4(4):275-287.

[5] Miller, G., *et al*, 1987. Expert Systems Tools for Hubble Space Telescope Observation Scheduling. *Telematics and Informatics* 4(4):301-311.

Artificial Intelligence Techniques for Scheduling Space Shuttle Missions

Andrea L. Henke, Richard H. Stottler
Stottler Henke Associates, Inc.
916 Holly Road
Belmont, CA 94002
(415) 595-1692 (415) 595-0358 (FAX)
E-mail: 76420,3622@compuserve.com

N95- 23755

KEY WORDS AND PHRASES

Artificial intelligence, intelligent entities, object-oriented, planning, scheduling.

ABSTRACT

Planning and scheduling of NASA Space Shuttle missions is a complex, labor-intensive process requiring the expertise of experienced mission planners. We have developed a planning and scheduling system using combinations of artificial intelligence knowledge representations and planning techniques to capture mission planning knowledge and automate the multi-mission planning process. Our integrated object-oriented and rule-based approach reduces planning time by orders of magnitude and provides planners with the flexibility to easily modify planning knowledge and constraints without requiring programming expertise.

MISSION PLANNING PROBLEM

High-level mission planning is begun from 5 to 10 years prior to launch. The goal of this planning is to establish a flight manifest, define the objectives, capabilities and constraints of the missions comprising the manifest, and translate those into hardware, software and flight procedures. The manifest must reflect the precedence and duration of Shuttle processing activities, constraints such as facility utilization, work shift requirements, interval between launches, maintenance requirements, and other processing ground rules, to achieve a specified flight rate. Each mission flow consists of a standard set of processes of varying durations applied to a specific Orbiter. The manifest must reflect the precedence of certain processes, the facilities required and the constraints upon Shuttle processing. Additionally, unplanned or non-standard activities must be incorporated into a specific mission's flow.

Another important objective of high-level mission planning is to explore alternative planning options. These exercises determine how the flight manifest is affected when program ground rules are changed, new facilities are constructed, launch delays are anticipated, or new vehicles are introduced. The planning options can be very diverse and speculative, involving concepts ranging from the impact of facility repairs, to crew rescue at the space station, to concepts still on the drawing board. Additionally, there is considerable time pressure to produce answers to "what if" questions quickly.

Until recently, the manifest planning process was largely manual, performed by planners with many years of experience in the domain. Because of the great importance, diversity and complexity of the high-level studies, mission planners can dramatically benefit from our automated system for manifest planning. The object-oriented approach results in a system that is comprehensive and flexible and can accommodate their changing needs.

AUTOMATED PLANNING SOLUTION

In a project funded by NASA, we developed the Automated Manifest Planner (AMP) to solve the multi-mission planning problem. AMP is a flexible, comprehensive planning tool which draws on artificial intelligence techniques from a number of different areas to meet the requirements for manifest representation, manifest design and manifest analysis. AMP is designed to capture the expertise of experienced manifest planners and provide comprehensive, interactive manifest planning assistance. The planner can choose among different planning methods for use at various levels of the scheduling process. AMP can automatically plan missions, taking into consideration resources, ground rules, constraints and planner heuristics to improve the scheduling. By making use of generic mission definitions and relevant constraints, AMP will generate a manifest from scratch or replan all or portions of an existing manifest. The resulting manifest has no resource conflicts, no broken ground rules, and all processing performed in the correct order. By utilizing planner rules of thumb, AMP allows novices to produce quality manifests.

AMP provides flexibility by allowing the planners themselves to modify ground rules, facilities and missions and interactively edit the manifest produced. AMP improves the turnaround time on planning options by orders of magnitude and dramatically reduces the time needed to modify and maintain the manifest. The tool allows timely response to both simple and complex studies, from slips in dates or modified task durations, to new facilities, Orbiters, or different types of launch vehicles.

The manifests generated by AMP are displayed immediately on-screen in bar chart format. The planner may use the mouse to graphically edit flows, activities and other aspects of the manifest in order to bend the rules or seize particular opportunities. Although automated planning will never produce manifests with resource conflicts, these problems may be introduced through the editing process. AMP will shift dates forward to accommodate delays or minor resource changes where possible, and flag remaining conflicts. The planner can then either fix these problems by hand, or more efficiently, automatically replan that portion of the manifest.

Interactive explanation capabilities are provided in the AMP tool to give the planners insight into the reasoning that produced the manifest. This includes the reasons for particular resource/facility assignments, the reason float time is present, or the reason launch dates or other processing dates were pushed back. These explanations allow the planner to identify opportunities to improve the manifest and give the planners greater confidence in the manifests produced.

Because of the diverse and dynamic demands of manifest planning, AMP was necessarily designed to be a general scheduling tool, offering planners a host of planning methods and techniques for customizing the system for a particular planning situation. For this reason, AMP has broad applicability beyond NASA manifest planning.

TECHNICAL APPROACH

AMP uses a combination of artificial intelligence techniques to allow both the automatic generation of correct manifests and the improvement of these manifests through captured planner heuristics. We employ an object-oriented representation for capturing ground rules, constraints, activities, missions and resources. The heuristics planners use in generating and analyzing manifests are represented as rules. The planning techniques combine object-oriented programming and rule inference strategies.

Representing the Manifest

In order to automate the manifest planning process and allow comprehensive manifest design and analysis, one must first establish a representation of the manifest and its components. These components include the generic flows and processing activities, scheduled flows and processing activities, ground rules, planning constraints involving task sequencing and desirable conditions, and the available resources. These resources are varied and include Orbiters, payloads, launch pads, Orbiter Processing Facilities (OPFs), Mobile Launcher Platforms (MLPs) and other facilities, and time resources, relating to time needed by certain processes and time required at certain locations and on certain equipment, and calendar time constraints.

These diverse manifest components can be captured using object-oriented techniques. A generic flow for a type of mission is an object containing a list of generic activities which are themselves objects which include slots for the types of resources needed to perform the activity, as well as associated scheduling methods. A manifest is an object which contains a list of particular flows. These particular flows are copies of the corresponding generic flows and contain a list of copies of the generic activities. These activities are linked together in a network which describes the required sequencing of operations.

The resources required by activities are organized into an object class hierarchy. The super-class is Required Facilities which has subclasses of OPFs, MLPs, and vehicles, for example. The OPFs class contains the three OPF instances - OPF1, OPF2, and OPF3 - corresponding to the three available Orbiter processing facilities. The Vehicles class has subclasses of Orbiters and HLLVs (Heavy Lift Launch Vehicles). The Orbiters class contains 4 instances representing the four Space Shuttle Orbiters.

Constraints and ground rules may be represented using a combination of objects and rules, as appropriate. For example, one special required facility is called Space and has one instance. This one instance, along with the flight activity's requirement for a Space resource, represents the constraint that only one Orbiter can be in space at a time. Typical ground rules include Orbiter Maintenance Down Period (OMDP) times and locations, the influence of payloads on durations, and special procedures.

Capturing Planner Expertise

An important aspect of many AI development efforts is the capture of the corporate knowledge of the experts. By eliciting and storing the details of a process, novices can be productive even when the experts are unavailable. The required knowledge for manifest planning can be captured in a number of ways. First, the expert's knowledge about the events and processes in a typical mission is captured in a generic flow. The generic flow represents the overall sequence of the processing activities in a mission. This flow preserves the required order of those activities and the resources required for each activity. Second, alternative planning methods are used to capture the expert's approach to planning and resource allocation for the activities in a flow and the flows in a manifest. For example, the expert planner may schedule certain flow activities in a forward direction, a backward direction, or in a priority order from certain dates or activities. Finally, rules are used to capture exceptions or additions to the standard flow. A rule is attached to the object to which it relates. Rules often add or delete activities to the specific flow. For example, a rule adds the activities of transporting the Orbiter to and from Palmdale, California if OMDP processing is required and that processing should take place in California rather than at Kennedy Space Center.

Intelligent Entities

An object-oriented approach allows the system to represent activities and activity scheduling information as objects. The objects are organized into an object hierarchy or class structure, where objects in the same class share characteristics. The object hierarchy for AMP includes objects and classes of objects to represent manifests, individual missions, processing activities, facilities, vehicles, etc. These objects are not passive data, but individual, intelligent entities that can be requested to perform actions on themselves or each other. These objects know how to schedule and unschedule themselves, and plot and erase themselves.

When the planner wants to initiate planning of a manifest, he or she in effect sends a message to the manifest object telling it to plan itself. The manifest object responds by sending scheduling messages to each of its missions. Each mission schedules itself by sending scheduling messages to each of its constituent processing activities. Each activity schedules itself by sending messages to other activities and making scheduling requests of each of its required resource classes, such as the class of OPFs or the class of MLPs. The resource classes respond to schedule requests by sending messages to each particular resource in their class. Each particular resource then checks its own availability and sends that information back to the class which makes the best resource selection. As each activity responds to scheduling requests, it checks its own local slots for rules and scheduling method choices, firing rules and executing the appropriate scheduling methods. After all these recursive planning calls have been made, the manifest object plots itself on-screen. Plotting follows the same level-by-level sequence.

The concept of intelligent entities, described above, allows the planner to mix and match different scheduling methods for different entities. It also facilitates capture of the planners' heuristic knowledge by the planners themselves. Because the scheduling problem is broken down into so many separate smaller problems, very complex scheduling is performed by relatively simple methods. These simple methods allow the easy inclusion of rules to alter planning methods in certain circumstances. Because each entity represents such a small part of the overall problem, the rules required for each entity are very simple and few in number and are tailored to each object's planning method. There is almost no interaction between the rule bases, because they are only related to the intelligent entity (such as an activity) to which they are attached. The small number and simple form of the rules makes it easier for the planners to enter these rules themselves or to have semi-automatic learning capabilities generate the rules.

Another design principle of AMP is the philosophy of permitting the planners to access all parts of the system, including the resource hierarchy, generic and specific missions and activities, plot definition files, and rules attached to each entity. This philosophy gives the planners maximal flexibility to tailor AMP to fit their changing needs without requiring programming expertise.

AMP DEVELOPMENT

The AMP project involved extensive knowledge engineering with the NASA expert planners. AMP was developed as a series of incremental releases which provided extensive planning, plotting, and editing options and methods. The Mission Planning Office is using AMP to perform Shuttle manifest planning and the more speculative alternative planning studies. AMP can plan one year of Shuttle flows in one minute on a 486 PC.

CONCLUSION

AMP substantially reduces the time required to maintain NASA's flight manifest and perform studies. This improves response time and allows planners to play a more proactive role in the studies. By allowing the planners more time to concentrate on the significant or unusual aspects of scheduling, they may be able to generate better manifests, and produce them more quickly. Additionally, by modeling planner expertise, less experienced planners can take advantage of the knowledge of planning experts and generate better manifests or work with less supervision.

The flexibility required by the mission planners dictates that the tool be so flexible as to make AMP adaptable to almost any scheduling problem, including planning for detailed Shuttle and payload processing, manufacturing scheduling, etc. We recently completed a project for Johnson Space Center in which we applied AMP techniques to the planning of the crew activity timeline for both Shuttle and space station flight planners. We expect to implement a full-scale version for their daily use.

Design and Implementation of An Experiment Scheduling System for the ACTS Satellite

N95- 23756

Mark J. Ringer
NYMA Inc.
NASA Lewis Research Center Group
Cleveland, Ohio 44142
(216) 433-8060
Ringer@mars.lerc.nasa.gov

KEY WORDS AND PHRASES

Planning and Scheduling

INTRODUCTION

The Advanced Communication Technology Satellite (ACTS) was launched on the 12th of September 1993 aboard STS-51. All events since that time have proceeded as planned with user operations commencing on December 6th, 1993. ACTS is a geosynchronous satellite designed to extend the state of the art in communication satellite design and is available to experimenters on a "time/bandwidth available" basis. The ACTS satellite requires the advance scheduling of experimental activities based upon a complex set of resource, state, and activity constraints in order to ensure smooth operations. This paper describes the software system developed to schedule experiments for ACTS.

DOMAIN DESCRIPTION

ACTS is a next generation communication satellite that incorporates three main technical gains: Demand Assigned Multiple Access - Time Division Multiple Access (DAMA-TDMA) with very small (0.3°) hopping spot beam antennas, use of Ka Band (30/20 GHz), and onboard processing. The DAMA-TDMA beam-hopping network allows multiple geographically distributed users to access the satellite virtually simultaneously with smaller aperture antennae. On-board processing allows rain-fade alleviation algorithms to be added to the communication path since the Ka band is more susceptible to attenuation by rain. Very high data rates are possible in the Ka band, these rates can approach 800 megabits per second.

The ACTS scheduling system considers a large amount of information from both experimental and operational activities during the scheduling process. This information is classified into four categories: activity, calendar, resource, and state constraints. Activity constraints encompass the requests for duration, terminal usage, bandwidth, rain-fade type, and terminal spot beam location. Calendar constraints include predetermined events such as eclipses of the satellite and planned maintenance. Resources include both the bandwidth constraints for each spot beam and the bandwidth requested by the experimenters. The processors onboard ACTS allow 31 possible configuration "states" connecting uplink beams to the processors then to the downlink beams. Each experimenter requires a subset of these states to successfully complete their experiment.

IMPLEMENTATION STRATEGY

The entire scheduling process begins with a database of user requests. Requests are then individually scheduled by the human scheduling expert with the aid of the ACTS Scheduler. The generated schedules represent a valid, conflict free set of events that satisfy experimenters' requests. These events are then output in a timeline format that details hour-by-hour events on the satellite. Information is sent through the database which adds domain specific knowledge for configuring the satellite. Configuration orders are then sent to the ACTS Master Control Center to be uplinked to the satellite. This process is shown in Figure 1.

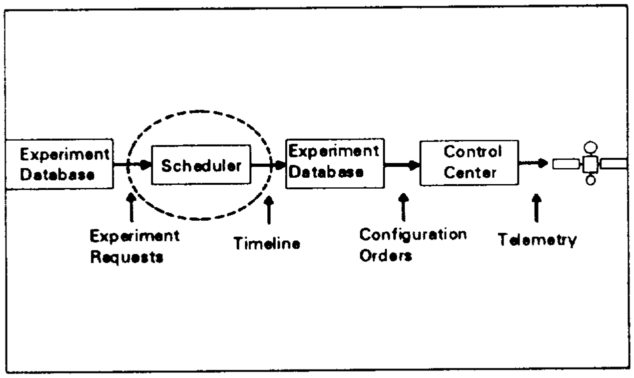


Figure 1 Complete Scheduling Process

SCHEDULING PROCESS

The ACTS Scheduler is a resource-based experiment scheduler [Biefeld 1990, Johnston 1989]. The major resource constraints are classified as capacity (non-depletable) resources which model communication bandwidth. The resource hierarchy must also include parent children relations. A value subscribed to a child resource must also be subscribed to the parent resource, and so on. Because each experiment is usually unrelated to others via temporal relations, temporal precedence constraints are not needed to model the domain of ACTS. Each experiment may request multiple runs, therefore, the ACTS Scheduler must be able to represent multiple instances of an activity. Each of these instances may also be slight variations on the original experiment to meet time and/or bandwidth constraints during the time frame of the instance.

Schedules are generated in a human-computer interactive paradigm within the confines of a constructive scheduling framework. For reasons that are to detailed to completely justify in this paper, automated scheduling 'rules' are neither necessary nor feasible for inclusion in the ACTS Scheduler. The rules needed for automated scheduling are both difficult to capture and constantly varying. For these reasons, a human-computer interactive paradigm was chosen to generate schedules. In this paradigm, the computer performs all of the computationally intensive valid interval calculations, resource updates, activity instance tracking, while the humans perform the functions that require heuristic knowledge [Fox 1992].

A constructive scheduling framework can be defined in the following manner. The initial schedule is free of constraint violations, being either empty or populated with activities that as a whole violate no constraints. Considering the initial case, the constructive method generates a schedule by 1) choosing an activity to schedule, 2) finding all possible temporal periods that the activity can be placed without violating any constraints, 3) deciding one temporal location to place the activity, and finally, 4) updating all the constraints affected by the activity. This four step process is repeated until either activities can no longer be placed on the schedule (without constraint violations) or no more unscheduled activities exist. In a fully automated scheduling system, items 1 an 3 are the functions that requires heuristic knowledge, while items 2 and 4 require a meticulous and time consuming search and data consistency effort. Items 1 and 3 are often times domain specific, while items 2 and 4 are more generic across multiple scheduling problems. The basis of the joint human computer effort is the split of items 1 and 3 to the responsibility of the human, while items 2 and 4 are the responsibility of the scheduling software.

REPRESENTATION DETAILS

Three extremely difficult representation problems exist within the ACTS scheduling project: unconventional resource hierarchies, multiple admissible state constraints, and context dependent overhead. Resource hierarchies are addressed in many commercial scheduling packages, but with a very limited scope. For example, consider a construction scheduling problem where 4 electricians, 3 plumbers, and 2 carpenters are working. In this case, a total of 9 workers are being consumed, the sum of the three specific technical areas. This concept is called conjunctive inheritance. In the ACTS scheduling project, a type of inheritance named maximal disjunctive is defined. The resource usage of the parent is defined as the value of the single largest resource user of its children. For example, if three activities were using 4,3, and 2 units of a maximally disjunctive resource (which have a common parent), only 4 units would need to be subscribed to the parent resource. These two inheritance types are described in Figure 2. A boolean inheritance is also defined. For each child that consumes a non-zero amount, a value of one (1) is subscribed to the parent. The maximal disjunctive inheritance type is used in the ACTS uplink channels when multiple communication frequencies overlap within the processing equipment onboard. The boolean inheritance is used to allocate overhead during the sharing of ground terminals.

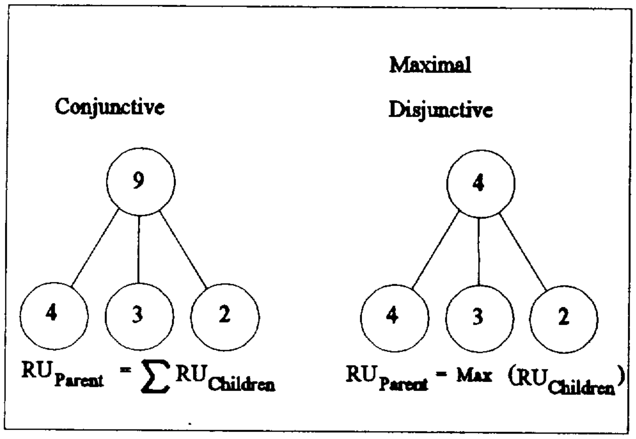


Figure 2 Resource Inheritance Types

State constraints are among the most difficult of problems within scheduling. The difficulty stems from the fact that state constrained variables have a temporal cost of transformation from one value to another. In the ACTS scheduling problem, an additional caveat is added, one that I call multiple admissible state constraints. A request for a conventional state constrained variable is in the form *Activity 'a' requests Resource 'r' to be in State 's'*. The multiple admissible state constraints in ACTS can be stated in the form *Activity 'a' requests Resource 'r' to be in one of the States* $(s_0, s_1, \ldots s_n)$. This adds a host of complications in the representation and reasoning about state resources.

The most unconventional of the constraints in the ACTS scheduler is the context dependent overhead. Since ACTS uses time-division multiplexing, requests for communication bandwidth are actually converted to time slots on the satellite. An activity not only needs multiples of these time-slots, but an overhead amount based upon the number, location, and type of terminals concurrently operating. The rules governing overhead dependency based upon number, location, and type of terminals concurrently operating are not straight forward. Because of the nature of these rules, it is very difficult to incrementally add the correct amount of overhead to the schedule. Therefore, two sets of resource usages are kept, conventional usage and overhead usage. When modifications are made to the schedule, the overhead is recomputed from scratch. If the overall resource usage is needed, these two numbers are simply summed. Another difficulty arises from the fact that the overhead has a temporal extent unrelated to the activity duration. In particular, the overhead allocated to an activity must have a temporal extent that spans the duration between state changeovers.

CURRENT WORK

Operations of the scheduling system started on December 6, 1993. Operations of the satellite have ramped up from checkout phase to an operational phase. During the first few months of operations, a multitude of minor modifications and additions have been completed. All of these additions have been requested by the customer in order to either make the scheduling process run more smoothly or to more correctly model the domain.

Currently, a Graphical User Interface GUI is being developed and tested. Since the ACTS scheduler was developed on such a tight timescale, only a text-based user interface was initially developed. In order to increase the information transfer to the human scheduler, a graphical representation of timelines, resource usages, and Gantt charts is in development. This will allow the human scheduler to more closely and accurately assess the state of the schedule during the scheduling process.

CONCLUSION

The ACTS scheduling project was undertaken with severe time pressures. The software was essentially written in five months with the additional assistance of previous schedulers being written by the author [Ringer 1991, Ringer 1993]. Without the scheduler to generate valid schedules and output them to generate orders for satellite configuration, operations would not have proceeded as smoothly as they have. The scheduler represents a custom designed piece of software that is unavailable in an off the shelf form. Numerous domain specific constraint types have been modeled to accurately solve the scheduling problem. Most importantly, the scheduling system significantly reduced the time necessary to generate and modify valid experiment schedules for ACTS.

REFERENCES

[Biefeld 1990] Biefeld, E., Cooper, L. *Operations Mission Planner: Final Report.*, JPL Publication 90-16, March, 1990.

[Fox 1992] Fox, B., *Chronological and Non-Chronological Scheduling*, In Proceedings AAAI-92 Workshop on Knowledge-Based Production Planning, Scheduling, and Control, July, 1992.

[Johnston 1989] Johnston, M. *Knowledge-Based Telescope Scheduling.* In Knowledge-Based Systems in Astronomy, Springer-Verlag, 1989.

[Ringer 1991] Ringer, M.J., Autonomous Power System: Integrated Scheduling. In *Proceedings NASA Johnson Space Operations, Applications, and Research Symposium*, Houston, Texas, July 1991.

[Ringer 1993] Ringer, M.J. *Time-Phased Scheduling Abstractions.* Masters Thesis, Case Western Reserve University, May 1993.

Session PS-NT

Planning and Scheduling Workshop: New Techniques

New Techniques

Telescope Loading: A Problem Reduction Approach

John L. Bresina
Recom Technologies
AI Research Branch, Mail Stop: 269-2
NASA Ames Research Center, Moffett Field, CA 94035-1000
Phone: 415.604.3365, *Fax:* 415.604.3594, *E-mail:* `bresina@ptolemy.arc.nasa.gov`

KEY WORDS AND PHRASES

Automated telescope management, periodic observation campaign, problem reduction, scheduling, telescope loading.

ABSTRACT

This paper presents a problem reduction approach to telescope loading. To study time-varying celestial behavior, astronomers submit periodic observation campaigns which involve a sequence of observations at a given sampling frequency over months or years. The loader's task is to generate an assignment of observation tasks to each night in the time window such that resource demand does not exceed resource capacity and such that the observations usefully contribute to the campaigns' scientific purposes, in a manner that is fair to all participating astronomers.

INTRODUCTION

In order to carry out a scientific campaign involving the study of a time-varying celestial behavior, an astronomer submits requests for observation time on a telescope. Satisfying such a campaign requires periodic observations of the celestial object over an extended interval of time (months or years). The number of scientific campaigns that the community of astronomers would like to pursue overwhelms existing telescopes. Enabling astronomers to effectively pursue their campaigns is an important and difficult problem. This is the problem addressed by telescope loading. Telescope loading involves assigning each observation to a particular night for execution such that underutilization of telescope time is minimized, oversubscription is eliminated fairly, and the astronomers' scientific goals are served.

In our application domain, the telescopes are land-based and fully automatic; a telescope control computer opens the observatory at twilight and collects data through the night without human assistance [4]. We are implementing an overall automated management system [2; 3] to enable participating astronomers to submit observation requests and obtain results from a remotely located telescope, *via* electronic communication networks, without the necessity of human intervention. In addition to the telescope loader described in this paper, the system also includes a night scheduler [10]. Each night, the observations assigned by the loader are given to the scheduler to determine the time each one will be executed.

Simply demonstrating, by construction, the feasibility of a solution to a telescope loading problem is not sufficient — the quality of the solution is an important consideration. A "good" loading assignment uses all available telescope time on observations that usefully contribute to the submitted campaign goals, in a manner that is fair to all participating astronomers. To evaluate a particular loading assignment, the expected quality of the schedule for each of the nights must also be taken into account. The loader uses the night scheduler and its evaluation function to determine expected schedule quality for a night's candidate loading assignment.

Telescope time is typically oversubscribed and it is usually impossible to fully satisfy every submitted observation campaign. Hence, in this application domain, the problem-solving method must be able to relax the constraints of the initial problem until it is satisfiable. Since there are many alternative ways of relaxing the problem, when relaxation is necessary, the complexity of loading is increased. The ideal objective is to find an optimal solution to a minimal relaxation of the initial problem; however, this ideal is seldom realizable. Thus, problems in this domain cannot be effectively solved (in general) using existing optimization methods from the fields of operations research or artificial intelligence.

In this paper, we describe our proposed solution method for the telescope loading problem. In order to reduce the search complexity, our solution method employs a problem reduction approach. Problem reduction is a type of "divide-and-conquer" technique that recursively decomposes a problem into conjunctions of subproblems until the subproblems are simple enough to easily solve, then all the subproblem solutions are composed to form the solution to the original problem.

LOADING PROBLEM

A telescope loading problem consists of a time window, a set of campaign goals, and a specification of resource capacity for each night in the time window. Our campaign specification language is based on a proposal by Louis Boyd, Director of Fairborn Observatory [1]. In this paper, we only describe aspects of the specification language needed

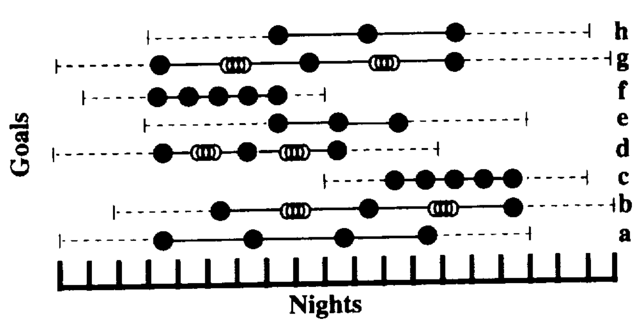


Figure 1: Schematic of an example loading problem.

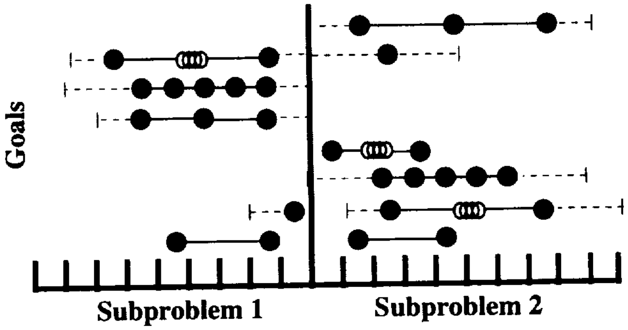


Figure 2: Schematic of an example problem reduction.

to explain the solution method.

Each campaign consists of one or more periodic observation goals. Briefly, a campaign goal specifies a number of repeated observations to execute within a given time window with a given time gap between executions. The number of repeated observations is specified with an ideal execution count and a minimum execution count; if it is not possible to obtain the minimum, then the campaign goal should not be attempted. An astronomer specifies the ideal gap (in days) between executions either with a fixed gap length or with a gap probability distribution (used, for instance, to reduce aliasing in the data or to determine the period of a recently discovered variable star). For example, a fixed gap length of one indicates that the observation should be executed every other night. An example of a gap distribution specification is the uniform probability distribution, $U(0,2)$, which indicates that gaps should be randomly selected with a uniform probability from the set {0 days, 1 day, 2 days}.

Figure 1 illustrates a small example loading problem covering 19 nights and containing eight hypothetical campaign goals (**a**–**h**). Each goal is pictured as a sequence of observations which can slide within an window of nights, indicated by the dashed lines. A probabilistic (variable) gap between observations is indicated with a mechanical spring, and a fixed gap is indicated with a solid line. The loader's objective is to place each sequence within its window such that no night is overloaded. It may not be possible to achieve this objective with respect to the ideal gaps and the ideal number of observations specified by each astronomer. Some of the observations may not be done, and some of the gaps may be longer or shorter than ideal. The transformations that the loader can apply to the desired observation sequences are: (*i*) shrinking the goal's time window, (*ii*) clipping some observations from a sequence, and (*iii*) stretching or shrinking the gaps in a sequence. The first transformation restricts the possible nights to which the observations may be assigned, and the later two are problem relaxation transformations. If the time window is reduced too much, or if the execution count is reduced too much, or if the gap length is increased too much, then the scientific purpose of the campaign goal will not be served and the data collection activity will have wasted valuable telescope time. For each campaign goal, the astronomer specifies the limits of these transformations and specifies the relative desirability of the types of relaxations.

The resource capacity profile is a specification of the projected number of hours of observation time available for each night in the time window at a particular telescope. The amount of twilight time depends on the time of the year, and how much of that observation time can be expected to be available varies due to seasonal weather patterns. After each night, the loader tries to reassign the unexecuted tasks to future nights; if the initial loading assignment fills up all available observation time, then oversubscription could continue to grow worse over time. Therefore, in addition to accounting for changing observation availability, we need to leave some margin of capacity for future revision to the initial loading assignment.

LOADING PROBLEM REDUCTION

In this section, we describe our problem reduction approach to solving telescope loading problems. Our problem reduction approach for loading is a three stage process. First, a temporal decomposition process is applied to partition the problem's time window. Second, a campaign decomposition process is applied to the campaign goals one at a time; this process splits the campaign goals among the subproblems such that the average load is balanced. Third, a relaxation process is applied to each subproblem in an attempt to reduce oversubscription of resource capacity. Problem reduction is then recursively applied until all observation tasks are assigned to specific nights. Figure 2 illustrates a single application of problem reduction to the problem in Figure 1.

Our problem reduction technique implements a refinement process that incrementally restricts the set of possible nights that each campaign goal can be assigned for execution, incrementally balances the load, and incrementally relaxes the initial problem.

Each decomposition modifies campaign goals and shifts the load within the local context of a single subproblem. For problem solving efficiency, our reduction process ensures that the subproblems created are independently solvable; *i.e.*, that all subproblem solutions can be composed (*via* concatenation) into a valid solution to the original problem. We next discuss each reduction stage.

Temporal Decomposition

The objective of the temporal decomposition process is to partition the given problem's time window based on an analysis of resource contention. For each subinterval, a subproblem is created to cover that time window. For simplification, we restrict the temporal decomposition space by partitioning into only two subintervals. An heuristic evaluation function is used to select the best two-element partition of the problem's time window.

Resource contention is computed by subtracting resource capacity from resource demand; a contention greater than zero means the telescope is oversubscribed. The resource demand for a particular night is the observation time needed to satisfy the requests. The resource demand profile depends on how the requests will be assigned during the loading process. The expected demand for a given night is the summation, over all campaign goals, of the product of the goal's demand and the probability that the goal will be assigned to that night; it is assumed that all the possible start dates for a campaign goal are equiprobable (*cf.* Muscettola and Smith [7]).

Campaign Decomposition

The campaign decomposition stage involves placing each campaign goal of the parent problem into a subproblem or splitting it between the subproblems. The primary objectives are to restrict loading assignment alternatives and help balance the load.

If a campaign goal's time window is a subset of a subproblem's time window, then that goal can only be incorporated into that one subproblem's campaign set. All the goals of this type are processed first. In our example, only goals **c** and **f** are of this type. The remaining goals are processed one at a time based on which campaign goal can make the biggest impact on balancing the load between the two subproblems. The system selects the goal with the highest "load shift potential", which is the maximum amount of its demand that can be shifted from the high contention subproblem (*i.e.*, the one with a higher average contention) to the low contention subproblem.

Goals of this type can either be split across both subproblems, or their time windows can be shrunk to fit into one of the subproblems; the later was done to goal **h** in our example. When a goal is split, two campaign subgoals are created. In order for the subproblems to be independently solvable, the possible assignments for the two subgoals must be consistent with the original goal's constraints.

The system uses a "greedy" approach to load balancing — when decomposing the selected campaign goal, the system shifts as much demand from the high contention subproblem as can be accommodated in the low contention subproblem (without going past the balance point). After decomposing the selected goal, the average contention of the two subproblems is updated and the load shift potentials of the remaining goals are updated. Then the campaign goal with the current highest load shift potential is selected next. This process repeats until all goals have been decomposed.

Problem Relaxation

After the campaign goals have been split between the subproblems, further modification may be required in order to achieve a zero average contention in each subproblem. If a subproblem's average contention is greater than zero, then the resource demand of one or more of its goals must be reduced by decreasing the execution count. In our example, one observation task was removed from goal **d**.

The determination of how much to alter execution counts and gap lengths is impacted by both hard constraints and soft constraints (*i.e.*, preferences). When decreasing a goal's demand, there are limits (specified by the astronomer) to how much the observation sampling strategy can be modified. When the contention within a subproblem can not be sufficiently reduced without violating a goal's hard constraints, then one or more of the subproblem's goals must be entirely eliminated.

An heuristic evaluation function is employed to provide guidance to the relaxation process at a given subproblem. The evaluation of a candidate campaign set measures the desirability of the relaxation with respect to each astronomer's preferences and relative priorities of the campaign goals. The heuristic evaluation combines the scores of all these local evaluations and selects the campaign set that achieves the most effective and fairest relaxation.

Recursive Application & Termination

Upon completion of the problem relaxation stage, the expected demand is computed for each subproblem and then subtracted from the capacity profile to derive the contention profile. The decomposition process can then be recursively applied to each subproblem.

When a campaign goal's time window becomes too small, candidate night assignments are generated for the goal's observations according to the gap specification and ideal execution count. The problem reduction process terminates when decomposition has terminated for all goals and each observation task has been assigned to a particular night.

From problem to subproblem, each successive modification is a fine-tuning, or specialization, of the preceding modification. The average contention derived from the first temporal decomposition covers the entire time span considered by the loader and is a very abstract characterization of the contention during that time window. Each successive subproblem produced (*via* recursive decomposition) has an average contention that covers a smaller time span and is a more accurate characterization of the contention. Hence, this combination of temporal decomposition and averaging automatically generates a hierarchy of abstraction levels.

CONCLUDING REMARKS

We have presented an approach to solving the telescope loading problem for periodic observation campaigns. The complexity of this loading problem is reduced by employing a problem reduction approach that reasons at different levels of abstraction and ensures that the subproblems created are independently solvable. The abstraction levels are automatically generated based on properties of the problem instance, namely, the contention profile. A given abstraction level not only depends on previous abstraction levels, but also on the decisions made in previous problem decompositions.

Our approach is a novel application of problem reduction to a domain in which reasoning about metric time is central, solution quality is important, and problem relaxation is necessary. The problem reduction technique implements an incremental refinement process. Each subproblem inherits the loading biases of all ancestor problems and imposes an additional bias on all descendant subproblems; the specific loading assignment produced is a result of the combination of all such biases.

The BAIT system [8] is a related automated telescope management system. The BAIT loader only considers the current night and uses a probabilistic selection technique to determine which of the active tasks to include. The assigned probability to a task is based on the desired (fixed) gap between observations and the time since the last observation.

Our loader addresses a similar problem to that addressed by the SPIKE system [5]. SPIKE solves the loading problem for the Hubble Space Telescope, at a grain size of about a week, employing a constraint satisfaction approach. Our loading approach is related to opportunistic scheduling approaches, especially those of Muscettola [6] and Sadeh [9], although neither of their systems has been applied to a problem domain with periodic requests. Though there are substantial differences between these two scheduling systems, both systems focus on bottlenecks and use variable ordering heuristics based on some type of contention analysis.

One of the primary differences from SPIKE and the opportunistic systems is that our approach incorporates problem relaxation. Another key difference from these three systems is that, in our approach, the reasoning is carried out at a much more abstract level, at least during the first few levels of problem reduction. The contention analyses, the problem solving decisions, and even the tasks assigned are initially quite abstract, in comparison to their approaches. As the problem is recursively decomposed, these aspects become more detailed. Though this research effort is in its early stages and system implementation is not yet completed, we conjecture that the combination of problem reduction and automatic, problem-specific abstraction should yield efficient problem solving and quality solutions.

Acknowledgements

Firstly, I would like to thank the rest of the telescope management project team: Mark Drummond, Keith Swanson, and Will Edgington. I would also like to thank Saul Amarel, William Borucki, Louis Boyd, David Genet, Russell Genet, Gregory Henry, Peter Friedland, Nicola Muscettola, Charles Schmidt, and Donald Smith.

REFERENCES

[1] Boyd, L., Epand, D., Bresina, J., Drummond, M., Swanson, K., Crawford, D., Genet, D., Genet, R., Henry, G., McCook, G., Neely, W., Schmidtke, P., Smith, D., and Trublood, M. 1993. Automatic Telescope Instruction Set 1993. *IAPPP Communications*, 52, T. Oswalt (ed.).

[2] Bresina, J., Drummond, M., Swanson, K., and Edgington, W. (1994). Automated Management and Scheduling of Remote Automatic Telescopes. In *Astronomy for the Earth and Moon.* D.P. Smith (ed.).

[3] Drummond, M., Bresina, J., Swanson, K., and Edgington, W. 1994. The Associate Principal Astronomer Telescope Operations Model. In *Proc. of i-SAIRAS 94.* JPL, Pasadena, CA.

[4] Genet, R.M., and Hayes, D.S. 1989. *Robotic Observatories: A Handbook of Remote-Access Personal-Computer Astronomy.* AutoScope Corp., Ft. Collins, CO.

[5] Johnston, M.D. 1990. SPIKE: AI Scheduling for NASA's Hubble Space Telescope. In *Proc. of the 6th Conf. on AI Applications.* IEEE Comp. Society Press.

[6] Muscettola, N. 1992. *Scheduling by Iterative Partition of Bottleneck Conflicts.* Tech. rpt. RI-TR-92-05, The Robotics Institute, CMU.

[7] Muscettola, N. and Smith, S.F. 1987. A Probabilistic Framework for Resource-Constrained Multi-Agent Planning. In *Proc. of the 10th IJCAI.* Milan, Italy.

[8] Richmond, M., Treffers, R., & Filippenko, A. 1993. The Berkeley Automatic Imaging Telescope. *Publications of the Astronomical Society of the Pacific.* No. 105 (October).

[9] Sadeh, N. 1991. *Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling.* Tech. rpt. CS-91-102, School of Comp. Sci., CMU.

[10] Swanson, K., Bresina, J. and Drummond, M. 1994. Robust Telescope Scheduling. In *Proc. of i-SAIRAS 94.* JPL, Pasadena, CA.

High Performance Techniques for Space Mission Scheduling *

Stephen F. Smith
The Robotics Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213
Tel: (412) 268-8811; Fax: (412) 621-5477
E-Mail Address: sfs@cs.cmu.edu

KEY WORDS AND PHRASES

Artificial Intelligence, constraint-based reasoning, space mission scheduling.

INTRODUCTION

In this paper, we summarize current research at Carnegie Mellon University aimed at development of high performance techniques and tools for space mission scheduling. Similar to prior research in opportunistic scheduling, our approach assumes the use of dynamic analysis of problem constraints as a basis for heuristic focusing of problem solving search. This methodology, however, is grounded in representational assumptions more akin to those adopted in recent temporal planning research, and in a problem solving framework which similarly emphasizes constraint posting in an explicitly maintained solution constraint network. These more general representational assumptions are necessitated by the predominance of state-dependent constraints in space mission planning domains, and the consequent need to integrate resource allocation and plan synthesis processes.

First, we review the space mission problems we have considered to date and indicate the results obtained in these application domains. Next, we summarize recent work in constraint-posting scheduling procedures, which offer the promise of better future solutions to this class of problems.

SPACE-BASED OBSERVATORY SCHEDULING

Our research has focused specifically on space-based observatory management applications, which, like most mission planning problems, require allocation of resources to competing goal activities over time in the presence of complex state-dependent constraints. Such problems are typically categorized as scheduling problems, where observing time must be allocated so as to optimize overall performance objectives (e.g., maximizing scientific return, balancing observing priorities). Yet classical scheduling frameworks, which emphasize formulation of scheduling problems as assignment problems, prove insufficient in this case. Since the executability of a given observation also depends on conditions of the predicted spacecraft state other than resource availability (e.g., the operating state of the required viewing instrument, spacecraft power levels and pointing direction, the visibility of the target, etc.), solution feasibility can only be guaranteed by dynamically generating and synchronizing the auxiliary activities necessary to bring about and preserve enabling state conditions. In short, effective solutions to these problems must integrate resource allocation and plan synthesis capabilities.

Given the above problem characteristics, our initial research focused on the development of a modeling and problem solving infra-structure that synthesized the respective strengths of planning and scheduling frameworks. This effort led to the development of HSTS [8,9], a problem solving architecture that promotes an integrated view of scheduling and planning as an opportunistic process of constraint posting in an explicitly maintained solution constraint network. The HSTS problem solving architecture was originally developed and applied in the context of the problem of constructing short-term observation schedules for the Hubble Space Telescope (HST), motivated by the limitations of the current solution. In the HST domain, several results with the HSTS problem solving architecture have been demonstrated. The leverage provided by HSTS's emphasis on decomposable domain descriptions was demonstrated through experiments with a se-

---

* This research has been sponsored in part by the National Aeronautics and Space Administration, under contract NCC 2-531, by the Advanced Research Projects Agency under contract F30602-90-C-0119 and the CMU Robotics Institute.

quence of domain models that increasingly captured more and more of the telescope's operational constraints. The observation scheduler was shown to scale to the full problem, producing observation schedules complete with all necessary enabling activities such as instrument configuration, telescope repointing, data communication, etc. in a time frame acceptable for actual application [8]. Complementary results demonstrated the ability of "multi-perspective" scheduling techniques to produce better quality schedules, in terms of balancing conflicting mission objectives, than a variant of the short-term scheduling algorithm currently being used in HST mission operations [13].

More recently, HSTS has been used to develop of scheduler for application to a second orbiting telescope, the Small Wave Sub Millimeter Astronomy Satellite (SWAS), currently due to be launched in early 1995 [7]. The SWAS problem differs fairly significantly in character from the HST problem. Whereas scheduling in the HST domain is concerned with synchronization of well-specified programs of target observations, viewing goals in the SWAS domain are formulated as cumulative amounts of time to be spent on various targets. Thus, the SWAS scheduling task is to efficiently distribute and interleave viewing time among various targets. We have developed an initial, priority-based scheduling procedure, which operates with a domain model that ensures satisfaction of all dominant spacecraft operating constraints (e.g., slew time, target acquisition procedures, power constraints) and is designed to optimize an overall priority score defined by the SWAS mission team. The viability and potential of the scheduler was recently demonstrated using a provided set of reference targets and representative 1-week scheduling problems. In these experiments, the schedules generated show overall satellite utilization percentages of greater than 70% (over 20% higher than expectations provided a priori by the SWAS mission team), and problems are solved in 5-6 minutes on a SPARC IPX. During the coming months, plans call for integration of the scheduler into the SWAS mission planning software environment, and full-scale comparative testing against their current baseline approach.

SCHEDULING VIA CONSTRAINT POSTING

Methodologically, our research approach has been to combine incremental development of solutions to specific application problems with more basic investigations into more broadly applicable and higher performance longer term solutions. In this section, we summarize our progress toward exploiting the constraint posting scheduling framework that is promoted by HSTS.

As indicated earlier, research in constraint-based scheduling has typically formulated the problem as one of finding a consistent assignment of start times for each goal activity. The HSTS framework, in contrast, advocates a problem formulation more akin to least-commitment planning frameworks: the problem is most naturally treated as one of posting sufficient additional precedence constraints between pairs of activities contending for the same resources to ensure feasibility with respect to time and capacity constraints. Solutions generated in this way typically represent a set of feasible schedules (i.e., the sets of activity start times consistent with posted sequencing constraints), as opposed to a single assignment of start times.

While frameworks such as HSTS do not prohibit the use of "fixed time" scheduling techniques, there are several potential advantages to a solution approach that retains solution flexibility as problem constraints permit. From the standpoint of solution use, the generation of sets of feasible schedules provides a measure of robustness against executional uncertainty, allowing determination of actual start times to be delayed and minimizing the need for solution revision. From the standpoint of solution development, a constraint posting formulation of the problem can provide a more convenient search space in which to operate. During schedule generation, alternatives are not unnecessarily pruned by the need to (over) commit to specific start times. When the need for schedule revision becomes apparent, modifications can often be made much more directly and efficiently through simple adjustment of posted constraints.

Given these potential advantages, recent research has focused on development and evaluation of constraint-posting scheduling techniques. One approach, generalizing directly from the concept of bottleneck analysis used in previous work in opportunistic scheduling but without the "fixed times" assumption, has led to development of a procedure called Conflict Partition Scheduling (CPS) [10]. Experimental analysis on benchmark constraint satisfaction scheduling problems showed CPS to outperform two state of the art "fixed-times" scheduling approaches - a micro-opportunistic procedure [11] (based similarly on "contention-based" focus of attention) and a min-conflict iterative repair procedure [6].

Our more recent work has concentrated on the development of simpler, computationally cheaper alternatives to contention-based problem analysis when a constraint posting framework is assumed, leading to development of a procedure called Precedence Constraint Posting (PCP) [12]. PCP couples the use of previously developed dominance conditions for incremental pruning of the set of feasible sequencing alternatives [5] with a simple look-ahead analysis of the temporal flexibility associated with different sequencing decisions. At each step of the search, a measure of *residual temporal slack* is computed for each sequencing decision that remains to be made; the decision with the smallest residual slack is chosen as the most critical, and a precedence constraint is posted in the direction that retains the most flexibility. After posting the new constraint, dominance conditions are checked to identify other sequencing decisions that now have only a single feasible ordering; these unconditional decisions are also taken (i.e. the implied precedence constraints are also posted) before recomputing estimates of residual slack. The PCP procedure terminates when either all pairs of activities contending for the same resource have been sequenced, or an infeasible state has been reached. Experimental results with PCP on the same suite of constraint satisfaction scheduling problems have shown comparable problem solving performance to contention-based scheduling approaches with orders of magnitude reduction in computational time [12].

One of our principal current interests is applying the PCP procedure in more frequently encountered, optimization-based scheduling contexts (i.e., where the goal is not simply a feasible solution but a feasible solution that minimizes/maximizes some objective criterion). We are exploring two general approaches to adapting PCP for this purpose:

- *discrete relaxation search*, where PCP is embedded as a solution feasibility evaluator within a larger search through the space of possible constraint relaxations defined by the objective criteria, and
- *upper-bound improvement search*, where the PCP procedure itself is modified to directly incorporate the objective criteria (e.g., using estimates of "residual tardiness cost" as opposed to residual temporal slack), and a dynamically adjusted upper-bound solution provides the basis for search space pruning.

The utility of each of these approaches depends on characteristics of the specific optimization criterion that is considered. For example, the common manufacturing problem of minimizing weighted tardiness is better formulated as an improvement search, since there is no structure to support an effective search through the possible due date relaxations of all jobs. In this problem context, we have performed some initial experimentation with a configuration of PCP that utilizes a dispatch heuristic to estimate the tardy cost associated with different sequencing decisions, and decisions are used to incrementally improve an upper bound solution. This extended procedure has been shown to produce schedules 10-30% better (depending on problem constrainedness) than the combined results of the best priority heuristics known for the weighted tardiness problem on a generated set of large (1000+ activities) scheduling problems (with average solution time of 1 minute).

One criterion that is straightforwardly formulated as discrete relaxation search, however, is minimizing makespan (or overall duration) of the schedule (or equivalently maximizing resource utilization). We have developed a procedure, referred to as MULTI-PCP, which first establishes lower and upper bounds on the overall completion time of the schedule (using a critical path method and a simple dispatch heuristic respectively), and then searches for the minimum feasible "common due date" by repeatedly applying PCP to various dates within these bounds. We have contrasted the performance of this procedure with that of the shifting bottleneck family of procedures (SBP) [1] (one of the best approximation algorithms currently known for minimizing makespan) on a set of previously studied benchmark problems. In these experiments, MULTI-PCP was shown to produce competitive solutions (more often than not closer to the optimum than the solutions obtained with the shifting bottleneck procedure) in equivalent or less computation time. Moreover, on tests of larger problems (involving 1000 activities), we have shown PCP to consistently produce better results than SBP with increasingly better computational efficiency [4].

All of the benchmark problems mentioned above make assumptions of fixed activity durations and simple precedence constraints between related activities (Indeed these are the problem assumptions uniformly made in the classical scheduling literature.) In space mission scheduling domains, in contrast, goal activities may have imprecise (or adjustable) durations, and a much richer set of qualitative and quantitative temporal constraints may be imposed on goal activities. Recent experimental analysis with PCP on problems that incorporate such constraints has pointed up inadequacies in the use of simple temporal slack as a look-ahead bias; in such problem contexts, earliest (and latest) start (and end) time information provides a much less accurate estimate of temporal flexibility. To overcome this limitation, we have recently generalized our look-ahead model of temporal flexibility to instead rely on "shortest path" information [3] This extends the applicability of PCP to the full range of temporal constraints expressible in HSTS. Our short term plans are to further develop and apply this approach to the core SWAS scheduling problem of maximizing utilization under cumulative activity duration constraints.

Finally, we mention our recent application of constraint-posting scheduling techniques in a domain of some relevance in other space mission planning arenas: scheduling experiments in an automated robotic chemistry workstation (resident at CMU) to maximize parallel experimentation. This problem is dominated by the presence of finite temporal separation constraints between successive steps of individual experimental plans (e.g., a chemical reaction must be sampled by the robot 2 hours after the last sample taken). By developing a constraint-posting variant of the existing "fixed-times" scheduling procedure and introducing the capability to support flexibility in constraint specification, the utilization of the workstation was almost doubled [2]. A version of this scheduler has been operational since September, 1993.

Acknowledgments: As implied by the references, Nicola Muscettola (now at NASA Ames) has played a central role in much of the work summarized in this paper, and I am grateful for the research collaboration we have had over the past several years. Acknowledgments are due also to Casper Cheng, who has been a driving force in the development of the PCP family of scheduling procedures.

REFERENCES

[1] J. Adams, E. Balas, and D. Zawack, "The shifting bottleneck procedure for job shop scheduling", *Management Science,* 34(3):391--401, 1988.

[2] R.J. Aarts and S.F. Smith, "A high performance scheduler for an automated chemistry workstation", *Proceedings ECAI-94*, Amsterdam, Aug. 1994.

[3] C. Cheng and S.F. Smith, "Generating feasible schedules under complex metric constraints", *Proceedings AAAI-94*, Seattle, WA, August 1994.

[4] C. Cheng and S.F. Smith, "A new algorithm for job shop scheduling to minimize makespan", Working paper, The Robotics Institute, Carnegie Mellon Univ., Jan. 1994.

[5] F. Roubellat Erschler, J. and J.P. Vernhes, "Finding some essential characteristics of the feasible solutions for a scheduling problem", *Operations Research*, 24:772-782, 1976.

[6] S. Minton, M.D. Johnston, A.B. Philips, and P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems", *Artificial Intelligence*, 58:361--205, 1992.

[7] N. Muscettola and S.F. Smith. "Constraint-directed integration of planning and scheduling for space-based observatory management", *Proceedings SOAR-93,* August 1993.

[8] N. Muscettola, S.F. Smith, A. Cesta, and D. D'Aloisi, "Coordinating space telescope operations in an integrated planning and scheduling architecture", *IEEE Control Systems Magazine*, 12(1), February 1992.

[9] N.Muscettola, "HSTS: Integrating planning and scheduling", Technical Report CMU-RI-TR-93-05, The Robotics Institute, Carnegie Mellon University, March 1993.

[10] N.Muscettola, "Scheduling by iterative partition of bottleneck conflicts", *Proceedings 9th IEEE Conf. on AI Applications*, March 1993.

[11] N.Sadeh, "Look-ahead Techniques for Micro-opportunistic Job Shop Scheduling", PhD thesis, School of Computer Science, Carnegie Mellon University, March 1991.

[12] S.F. Smith and C.Cheng, "Slack-based heuristics for constraint satisfaction scheduling, *Proceedings AAAI-93*, Washington DC, July 1993.

[13] S.F Smith and D.K. Pathak, "Balancing antagonistic time and resource utilization constraints in over-subscribed scheduling problems", *Proceedings 8th IEEE Conference on AI Applications*, March 1992.

Applications of Dynamic Scheduling Technique to Space Related Problems ~ Some Case Studies

N95-23759

Shinichi Nakasuka and Tetsujiro Ninomiya
Department of Aeronautics and Astronautics, University of Tokyo
Hongo 7, Bunkyo-ku, Tokyo 113, JAPAN
TEL: +81-3-3812-2111 ext.6590 FAX: +81-3-3818-7493
e-mail: nakasuka@space.t.u-tokyo.ac.jp

KEY WORDS AND PHRASES

Artificial intelligence, machine learning, resource allocation, scheduling, space transportation system

ABSTRACT

The paper discusses the applications of "Dynamic Scheduling" technique, which has been invented for the scheduling of Flexible Manufacturing System, to two space related scheduling problems; operation scheduling of a future space transportation system, and resource allocation in a space system with limited resources such as space station or space shuttle.

DYNAMIC SCHEDULING

"Reactive Scheduling", in which the next operation to be performed is decided only when it is required, using a certain heuristic scheduling rule, has been widely utilized in manufacturing control or resource allocation of multiprocessor system. It has been pointed in many literatures that by employing sophisticated scheduling rules (called "dispatching or routing rules"), a schedule of some quality can easily be obtained with quite little computational load. This method is quite robust to the various changes and anomalies in the production lines, because the scheduling decisions are deferred until required. The weak point of this strategy is, however, that these rules only refer to local information for decision making, and so the performance of the generated schedule is sometimes much degraded in the global sense.

In the field of manufacturing control, "Dynamic Scheduling" has been proposed [1] to compensate for this shortcoming of the reactive scheduling. In this method, many scheduling rules are prepared beforehand, from which one is selected considering the instantaneous situations at each decision timing, such as machine status, buffer contents or current production requirements. Therefore, the scheduling decisions reflect more global information, which results in uniformly good scheduling performance in any line status or production requirements. For this objective, knowledge is required that predicts which rule is the best in a certain instantaneous situation, and machine learning has been utilized for acquiring such knowledge. For example, in [1], the relationships between the situation and the best rule is obtained in the form of decision trees.

SCHEDULING OF SPACE TRANSPORTATION SYSTEM

OTV Network and Its Scheduling

"OTV Network" has been proposed [2] as the low-cost, next generation space transportation infrastructure (Fig.1). This system is based on the space fuel stations (3 in Fig.1) and reusable OTVs (2 in Fig.1). The OTVs, dropping in at fuel stations for fuel supply on their way (C), carry out various missions such as satellite delivery (D), recovery or other satellite servicing. When one mission is completed, OTV nominally returns to the Low Earth Orbit (E), gets refurbished and waits for the next mission. This concept is much alike the truck

[redacted]

4

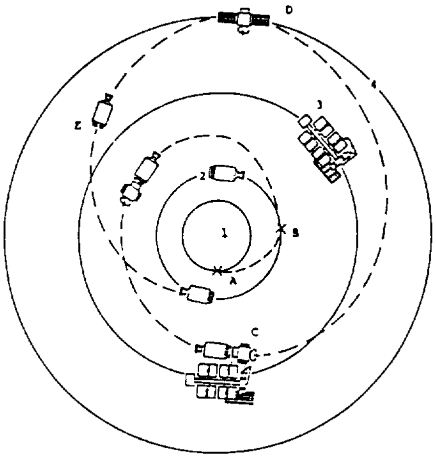


Figure 1. Transportation scenario of OTV Network

transportation system on the Earth such as trucks carry loads sometimes dropping in at gas stations.

For this OTV Network to work effectively and especially with low cost, various operations within the network must be thoughtfully scheduled so that the requirements as a transportation system (such as to deliver payloads by their due dates) can be satisfied while suppressing the total required cost as low as possible. The scheduling items include which payload to carry next with which OTV, which fuel station to use, which transfer trajectory to take, or what the OTV should do when its current mission is finished, etc. The scheduler must also deal with various anomalies such as failures of certain elements of the network system, urgent missions or mission changes.

Proposed Intelligent Scheduler

To meet these requirements, we have developed an intelligent scheduling system based on cooperation of distributed decision makers [3]. The decision making activities are assigned to several intelligent managers implemented in the computer, and they cooperatively perform decision making based on their own heuristic knowledge or the results of internal simulations where their knowledge is insufficient. Principally, the schedule is made in a so called reactive scheduling fashion, that is, the network operations are simulated according to a time line, and at each decision point one decision candidate is selected. The managers drive the lower level scheduler to predict future effects of each decision candidate, and with this predicted performance data they can pin-point one best candidate at each decision point. Reactive scheduling method is employed at the lower level for quickly simulating the optimal decision sequence to obtain such performance data.

Application of Dynamic Scheduling

The dynamic scheduling is applied to the reactive scheduling part of the proposed scheduling system, and the relationships between the situation and the best rule are obtained by machine learning using Neural Network with the back propagation algorithm. The employed Neural Network has three layers each of which has 18, 20, and 12 nodes. The data input into the input layer is a set of attributes representing the instantaneous situation of the OTV Network at the decision timing (such as the number of waiting satellites at each station), and the output layer dictates the most suitable decision rule from among 12 candidate scheduling rules (such as Minimum Slack Time Rule) which seem effective in deciding the next actions. For the back propagation learning, total 3630 data as to "18 attributes vs. the best rule" are generated by searching for the best rules exhaustively in various situation settings of the OTV Network, which have been utilized as the teaching signal for the Neural Network.

Some Simulation Results

In order to evaluate the effects of the employed scheduling architecture, the following three type schedulers are compared;

Sch.1) Single level, reactive scheduler alone

Sch.2) Proposed hierarchical scheduler without dynamic scheduling

Sch.3) Proposed hierarchical scheduler with dynamic scheduling

Table 1 summarizes the typical performance and required computation time for these schedulers. It indicates that by employing the hierarchical scheduling architecture, due date violations can be mitigated without too much additional fuel consumption. In addition, dynamic scheduling further improves the performance in both of tardiness and fuel consumption, and as a result the proposed scheduler can suppress the maximum and mean tardiness as small as one sixth of the reactive scheduler even with less fuel. The weak point of the proposed scheduler is its large computational load (about 10 times of the reactive scheduler case), but it can be said that the combinatorial explosion is suppressed to some extent.

Table 1. Summary of Scheduling Performance

Scheduler	Sch.1	Sch.2	Sch.3
Hierarchical Scheduling	off	on	on
Reactive Scheduling Rule	fixed	fixed	dyna.
Maximum Tardiness[1)]	14.0	4.4	2.6
Mean Tardiness	1.4	0.58	0.23
Number of Delayed Missions	10	9	5
Fuel Consumption[2)]	215	224	202
Computational Time[3)]	27	259	262

Note)
1) "Tardiness" means delay from due date (days).
2) Additional to the minimum requirement.
3) Measured using a computer with 300 MIPS performance (sec). (Mission density: 30 missions in 80 days)

RESOURCE ALLOCATION

Resource Allocation Problem

Resorce allocation is a very important problem especially in space, where the resource such as man power, electric power, water or tools are strictly limited and it is usually required that quite many tasks be performed within a limited time. The scheduling system must make the most of these limited recource to efficiently perform as many tasks as possible, and besides, in a case when a certain anomaly such as a malfunction of a certain tool or a degradation of power supply occurs, quickly remake the total schedule. In this case study, the dynamic scheduling technique is applied to a certain assumed resource allocation problem. Table 2 briefly describes the requirements given by the tasks and constraints in the assumed problem. Each task has its priority value, and the total scheduling performance is calculated by summing the priority values of the tasks to be completed within the fixed time.

Table 2. Requirements and Constraints for Resource Allocation Problem

Requirements:	(for each task respectively)
Starting Time	(Specified for some tasks)
Duration	Time required for the task
Man power	Number of required crews for;
- Type 1	Continuous attendance required
- Type 2	Occasional absense allowed
- Type 3	Occasional attendance required
Electric power	Power required for the task
Constraints:	
Time Limit	Total time allowed
Labor Hour	A crew's maximum labor hour/day Maximum hour of continuous work
Sleep Hour	A crew's required sleeping hour/day
Maximum power	Max. daytime power to be utilized
Battery power	Max. nighttime power to be utilized
Battery capacity	Max. energy to be loaded

Scheduling Strategies

Three type scheduling strategies are compared;

Sch.1) automatic scheduling using dynamic scheduling
Sch.2) automatic scheduling using a fixed simple scheduling rule
Sch.3) manual scheduling after some training

In both of Sch.1 and Sch.2, scheduling is performed by iterating the process of picking up one task from the pool of tasks which have not been scheduled yet then placing the task in a certain position in the scheduling table and allotting the required resources according to a certain rule. During the training of manual scheduling, it has been found that the selection of the next task to be scheduled determines the scheduling performance. Considering this, Sch.1 employs dynamic scheduling for this selection while Sch.2 utilizes a certain fixed rule. For the dynamic scheduling, 10 heuristic rules are prepared from which one is selected at each decision timing considering 22 attributes describing the situation at that timing. Examples of the rules are "select the task with the highest priority value" or "select the task with the least duration time" and so on, and the attributes include "the rate of operation of the crews" or "the maximum time window during which one, two and three crews are available", and so on. The relationships between the attributes and the best rules are acquired by the back propagation algorithm, using the data obtained by scheduling randomly generated small set of tasks with an exhaustive search strategy.

Performance Comparisons

Figure 2 and 3 describes the performance and required scheduling time of the three strategies and the exhaustive search result (i.e., optimum solution). Three levels of complexity, 1) 4 tasks in 150 minutes, 2) 6 tasks in 150 minutes and 3) 6 more complex tasks in 200 minutes are tried. Ten problems are generated randomly for each level, and maximum, averaged and minimum marks are calculated. It is observed that the fixed rule scheduling can generate schedule in quite a short time, but its performance is sometimes much degraded. On the other hand, the dynamic scheduling performs much better with slightly larger computational load, and the required time is still several orders less than the exhaustive search or manual scheduling. Moreover its computational time and performance do not get much worse even for more complex problems. These results indicate the effectiveness of the dynamic scheduling for quickly generating an acceptable schedule.

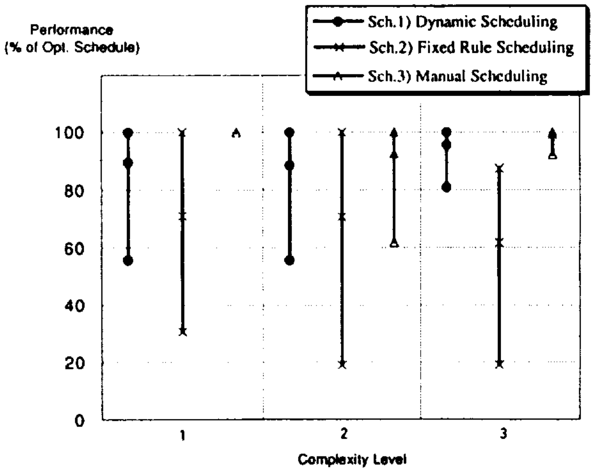


Figure 2. Performance of Three Scheduling Strategies

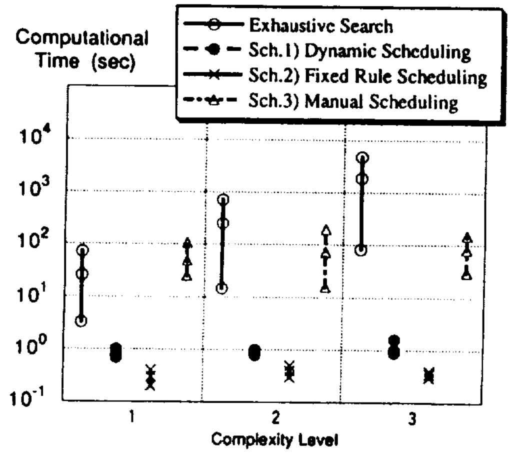


Figure 3. Required Computation Time

CONCLUSIONS

Dynamic scheduling has been applied to two space related problems; the scheduling of space transportation system and the resource allocation in a space system. The simulation results indicated that the dynamic scheduling can be effectively utilized as a sub-element of an overall scheduling system especially where the quick response is required, and that it will also provide an effective aid to an onboard rescheduling in case of some anomalies.

REFERENCES

[1] Nakasuka, S.; and Yoshida, T., 1992. Dynamic Scheduling System Utilizing Machine Learning as a Knowledge Acquisition Tool, *International Journal of Production Research*, 30: 411-431.

[2] Tanabe, T.; and Nakasuka, S., 1992. Fuel Station-based Transportation Network as a Future Space Infrastructure, *Preprints of 43rd IAF Congress*, IAA-92-0168.

[3] Nakasuka, S.; and Tanabe, T., 1992. Built-in Simulator and Intelligent Manager Architecture for Autonomous Scheduling of OTV Network Operations, In *IFAC Workshop on Spacecraft Automation and On-board Autonomous Mission Control.*

N95- 23760

Solution and Reasoning Reuse in Space Planning and Scheduling Applications

Gérard Verfaillie and **Thomas Schiex**
ONERA-CERT *
2 avenue Edouard Belin, BP 4025
31055 Toulouse Cedex, France
phone: (33) 61 25 25 25
fax: (33) 61 25 25 50
{verfail,schiex}@cert.fr

Key words and phrases

Artificial Intelligence, Dynamic Constraint Satisfaction Problems, Planning, Scheduling.

Abstract

In the space domain, as in other domains, the CSP (*Constraint Satisfaction Problems*) techniques are increasingly used to represent and solve planning and scheduling problems. But these techniques have been developed to solve CSPs which are composed of fixed sets of variables and constraints, whereas many planning and scheduling problems are *dynamic*. It is therefore important to develop methods which allow a new solution to be rapidly found, as close as possible to the previous one, when some variables or constraints are added or removed.

After presenting some existing approaches, this paper proposes a simple and efficient method, which has been developed on the basis of the *dynamic backtracking* algorithm [1]. This method allows previous solution and reasoning to be reused in the framework of a CSP which is close to the previous one. Some experimental results on general random CSPs and on operation scheduling problems for remote sensing satellites are given.

Space planning and scheduling applications and CSP

In the space domain, as in other domains, the *Constraint* based approach is increasingly used to represent and solve planning and scheduling problems. The CSP (*Constraint Satisfaction Problems*) framework offers a general formalism for constrained problems (any kind of constraint is allowed) and powerful solving methods [2]. Various *constraint programming* languages and tools have been developed these last years on this basis and are now available. Using them avoids long and useless software developments.

Let us recall that a CSP is defined by two sets: a set V of *variables* and a set C of *constraints*. Each variable has a finite set of possible values: its *domain*. Each constraint links a subset V' of the CSP variables and defines the set of the possible combinations of values for the variables in V'.

The usual problem is to find a solution, *i.e* a value for each variable such that all the constraints are satisfied. The most used methods are combinations of a *backtrack* search, using a *depth-first* strategy and some *heuristics* along with a *filtering* method (*forward-checking*, *arc-consistency*, *path-consistency* ...) which allows the search space to be pruned.

Dynamic problems: origin

A strong limitation of these techniques lies in the fact that they have been developed under the assumption that the sets of variables and constraints are given once and for all. In many real applications, and particularly in space applications, this assumption is not valid [3]. The reasons are numerous:

- before a mission, in the phase of specification and analysis, engineers may want to explore several alternatives and their implications; they may also want to derive a new specification from a previous one;
- during a mission, there is always a great difference between execution and forecast: operation results, durations, resource consumptions, possible breakdowns, ...
- according to new requirements or decisions of the people in charge of the mission, some new operations may have to be performed and others already planned and scheduled may have to be removed.

Dynamic problems: requirements

According to the computing point of view, all these situations are very similar: a previous CSP has been solved; a new one, which is close to the previous one (just some variables and constraints have been added or removed), has now to be solved. It is obviously

---

*This work was done at the University of New Hampshire (USA) and was supported the French Ministry of Defence (DGA-DRET).

C-6

possible to solve it from scratch, as it has been done for the first one, but this naive method may be very inefficient and lead to an instability of the successive solutions.

During the mission, if the time available to find a new plan or a new schedule is limited, *efficiency* can become a very important requirement. Before or during the mission, if some work (training, organization, orders ...) has been started on the basis of the previous solution, *stability* of the successive solutions can also be important.

Therefore one needs methods which, starting from the previous solution and the previous reasoning, allow a new solution to be rapidly found, as close as possible to the previous one.

Existing approaches

The existing approaches can be classified into three groups:

- *heuristic* approaches, which consist of using any previously consistent assignment (complete or not) as a heuristic in the framework of the current CSP [4];
- *local repair* approaches, which consist of starting from any previously consistent assignment (complete or not) and of repairing it, using a sequence of local modifications [5, 6, 7, 8];
- *constraint recording* approaches, which consist of recording any kind of constraint which can be deduced in the framework of a CSP and its justification, in order to reuse it in the framework of any new CSP which includes this justification[4, 9].

Dynamic backtracking

In spite of its name, the *dynamic backtracking* algorithm [1] does not deal with *dynamic* CSPs. The term *dynamic* means here that its *backtracking* mechanism allows the variables to be unassigned in an order which is different from the one which has been used to assign them. It can be described as follows:

- let *val* be a value which can not be assigned to a variable v, because of a constraint c which links v to previously assigned variables and would be unsatisfied; let V' be the set of variables linked by c; the set $V' - \{v\}$ is recorded as an *eliminating explanation* for *val*; the *conflict set* of a variable is the union of the *eliminating explanations* of all its eliminated values;
- let v be a variable whose current domain is empty, let V' be its *conflict set*; let v' be the last variable in V' according to the assignment order and *val'* be its current value; v' is unassigned; then all the *eliminating explanations* where v' is involved are removed (they are no more valid) and the set $V' - \{v'\}$ is recorded as an *eliminating explanation* for *val'*.

Termination, correctness and completeness of this algorithm have been proven.

Note the difference between such a mechanism and the usual *chronological backtracking* and *conflict directed backjumping* [10] mechanisms:

- *chronological backtracking* does not backtrack to v', but systematically to the variable which immediately precedes v according the assignment order;
- *conflict directed backjumping* backtracks (backjumps) to v', but, doing that, it unassigns all the variables which are between v' and v according to the assignment order;
- *dynamic backtracking* also backtracks to v', but it only unassigns v'.

This allows us to say that the *dynamic backtracking* mechanism is more pertinent and less destructive than both other ones.

Extended Dynamic Backtracking

Such features are very interesting in the framework of dynamic CSPs, when constraints and variables are added or removed in any order. For that, the notion of *eliminating explanation* has first to be extended in order to take into account constraints and variable domains as assumptions, as previously done with variable assignments. An extended *eliminating explanation* involves previously assigned variables (assignment constraints), variable domains (unary constraints) and usual constraints, which are together responsible for the value elimination. The previous description has just to be slightly modified to take into account this extension:

- let *val* be a value which can not be assigned to a variable v, because of a constraint c which links v to previously assigned variables and would be unsatisfied; let V' be the set of variables linked by c; the set $V' - \{v\} \cup \{c\}$ is recorded as an *eliminating explanation* for *val*; the *conflict set* of a variable is the union ...
- let v be a variable whose current domain is empty, let V' be its *conflict set* and $d(v)$ be its initial domain; let v' be the last variable in V' according to the assignment order and *val'* be its current value; v' is unassigned; then all the *eliminating explanations* where v' is involved are removed and the set $V' - \{v'\} \cup \{d(v)\}$ is recorded as an *eliminating explanation* for *val'*.

And the previous algorithm can be extended as follows to deal with dynamic CSPs:

- let c be a constraint which is added or restricted (this includes the case of restricted variable domains); if the current assignment does not violate c, there is nothing to do; else, let V' be the set of the variables linked by c; let v be the last variable involved in V' according the assignment order and val be its current value; v is unassigned; then all the *eliminating explanations* where v is involved are removed and the set $V' - \{v\} \cup \{c\}$ is recorded as an *eliminating explanation* for *val*.

- let c be a constraint which is removed or relaxed (this includes the case of relaxed variable domains); all the *eliminating explanations* where c is involved are removed;

Such an algorithm has very interesting properties:

- all the possible changes to a CSP (variable and constraint addition, removal and modification) are covered;
- previous solution and reasoning (*eliminating eliminations* previously recorded) are systematically reused; just the variable assignments which are no more consistent and the *eliminating explanations* which are no more valid are removed; in that sense, this extended *dynamic backtracking* algorithm combines the advantages of the *local repair* and *constraint recording* approaches and should provide goods results in terms of both efficiency and stability;
- changes can be taken into account at any time, either after or during the search;
- in case of inconsistency, the user can be provided with an explanation: a subset of the CSP constraints and domains which are together responsible for this inconsistency;
- computing *eliminating explanations* and *conflict sets* is a very simple task (only union operations are required) and the space required to record them is polynomially bounded (it is $O(nd(n+m))$, where n is the number of variables, m the number of constraints and d the maximum domain size);

Experiments, results and analysis

This algorithm (called *ddbt* for *dynamic dynamic backtracking*) has been experimented on *dynamic* CSPs and compared with others, like *conflict directed backjumping* (*cbj* [10]), *dynamic backtracking* (*dbt* [1]), *heuristic repair* (*hrp* [6]) and *local changes* (*lc* [8]), with *backward* and *forward-checking*.

A first set of general and binary CSPs has been used for these experiments. These CSPs have been randomly generated using fixed values for the number of variables (16) and the domain size (13) and various values for the constraint tightness (from 0.1 to 0.9), the graph connectivity (from 0.2 to 0.9) and the change size (ratio between the number of added or removed constraints and the number of constraints, from 0.01 to 0.16).

The results, which have been obtained by using *forward-checking* with each algorithm, are summed up in the four following set of curves. The three first ones show efficiency results (number of constraint checks) on underconstrained, intermediate and overconstrained problems. The last one shows stability results (distance between successive solutions, *i.e.* the number of variables which are differently assigned in both solutions) on underconstrained problems:

- the first and the third sets of curves show that *ddbt* is the most efficient on underconstrained (always consistent) and overconstrained (always inconsistent) problems;
- the second one shows that *cbj* remains the most efficient on the intermediate problems (the hardest ones to be solved; sometimes consistent, sometimes not), but that *ddbt* is not far worse;
- the fourth one shows that the algorithms which reuse the previous solution such as *hrp*, *lc* and *ddbt* provide a better stability than the others do.

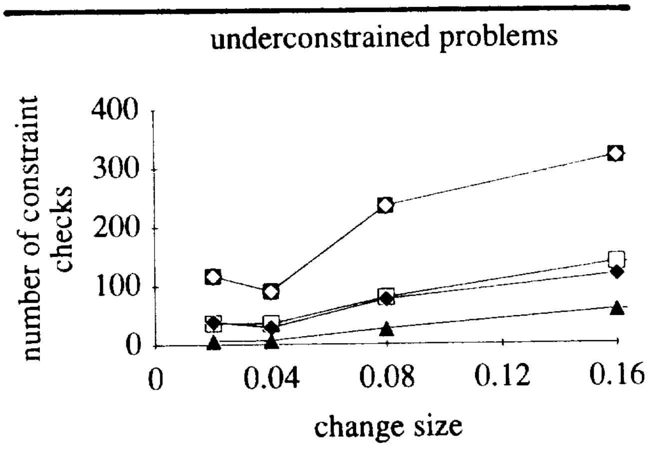
The same algorithms have been applied with the same kind of results on randomly generated operation scheduling problems for remote sensing satellites. These problems, whose definition comes from previous studies for the French Space Agency (CNES), in the framework of the SPOT program, are composed of a set of remote sensing satellites and a set of user observation requirements:

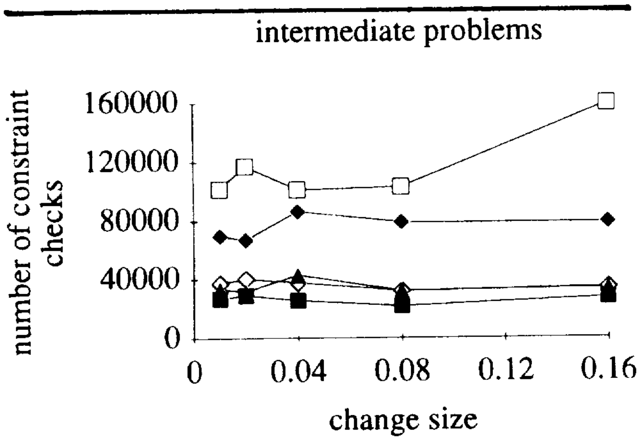
- each user requirement is defined by an area to observe and some constraints related to the mode, the quality and the period of the observation;
- each satellite is defined by its trajectory, its observation capabilities, its possible modes and minimal transition times between modes.

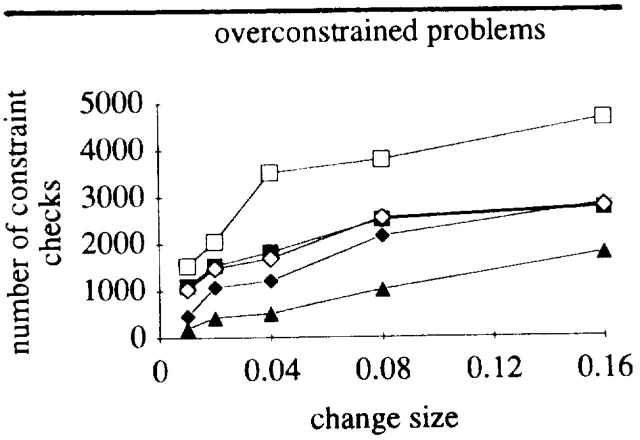
One assumes that these data allow a finite set of pairs (satellite, time slot) to be computed for each user requirement. In these conditions, the problem becomes a CSP where the only constraints are related to the minimal transition time between two time slots corresponding to the same satellite. But the lack of real data considerably limited the interest of these experiments.

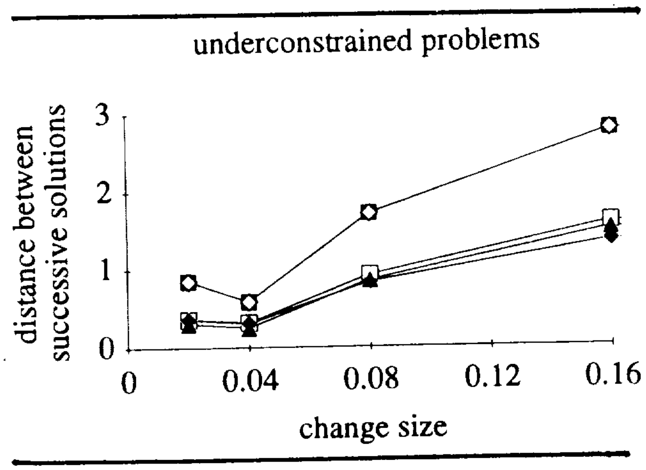
Conclusion

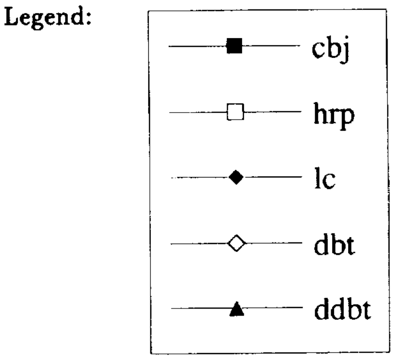
With this extension, the *dynamic backtracking* algorithm offers the opportunity to reuse previous solution and reasoning, when the problem changes, during or after the search. First experiments on small problems are promising. It should allow *dynamic* and *on-line planning and scheduling problems* to be efficiently dealt with. But these results have to be confirmed by larger experiments on various real problems.


underconstrained problems


intermediate problems


overconstrained problems


underconstrained problems



References

[1] M. Ginsberg, "Dynamic Backtracking", *Journal of Artificial Intelligence Research*, vol. 1, pp. 25–46, 1993.

[2] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press Ltd., London, 1993.

[3] C. Bastien-Thiry and G. Verfaillie, "Space Mission Plan : Robustness, Flexibility, Replanning and Explanations", *in Proc. of the ESTEC Workshop on Artificial Intelligence and Knowledge-Based Systems for Space*, 1993.

[4] P. Van Hentenryck and T. Le Provost, "Incremental Search in Constraint Logic Programming", *New Generation Computing*, vol. 9, pp. 257–275, 1991.

[5] B. Freeman-Benson, J. Maloney, and A. Borning, "An Incremental Constraint Solver", *Communications of the ACM*, vol. 33, pp. 54–63, 1990.

[6] S. Minton, M. Johnston, A. Philips, and P. Laird, "Minimizing Conflicts: a Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems", *Artificial Intelligence*, vol. 58, pp. 160–205, 1992.

[7] B. Selman, H. Levesque, and D. Mitchell, "A New Method for Solving Hard Satisfiability Problems", *in Proc. of AAAI-92*, pp. 440–446, San Jose, CA, 1992.

[8] G. Verfaillie and T. Schiex, "Solution Reuse in Dynamic Constraint Satisfaction Problems", *in Proc. of AAAI-94*, Seattle, WA, 1994.

[9] T. Schiex and G. Verfaillie, "Nogood Recording for Static and Dynamic CSP", *in Proc. of the 5^{th} IEEE International Conference on Tools with Artificial Intelligence*, Boston, MA, 1993.

[10] P. Prosser, "Hybrid Algorithms for the Constraint Satisfaction Problems", *Computational Intelligence*, vol. 9, pp. 268–299, 1993.

Empirical Results on Scheduling and Dynamic Backtracking

Mark S. Boddy and **Robert P. Goldman**
Honeywell Technology Center, MN65-2200
3660 Technology Drive
Minneapolis, MN 55418
612-951-7403 612-951-7436 (FAX)
{boddy,goldman}@src.honeywell.com

KEY WORDS AND PHRASES

Constraint Satisfaction, Scheduling, Dynamic Backtracking

INTRODUCTION

At the Honeywell Technology Center (HTC), we have been working on a scheduling problem related to commercial avionics. This application is large, complex, and hard to solve. To be a little more concrete: "large" means almost 20,000 activities, "complex" means several activity types, periodic behavior, and assorted types of temporal constraints, and "hard to solve" means that we have been unable to eliminate backtracking through the use of search heuristics. At this point, we can generate solutions, where solutions exist, or report failure and sometimes why the system failed. To the best of our knowledge, this is among the largest and most complex scheduling problems to have been solved as a constraint satisfaction problem, at least that has appeared in the published literature.

This abstract is a preliminary report on what we have done and how. In the next section, we present our approach to treating scheduling as a constraint satisfaction problem. The following sections present the application in more detail and describe how we solve scheduling problems in the application domain. The implemented system makes use of Ginsberg's Dynamic Backtracking algorithm [2], with some minor extensions to improve its utility for scheduling. We describe those extensions and the performance of the resulting system. The paper concludes with some general remarks, open questions and plans for future work.

CONSTRAINT ENVELOPE SCHEDULING

We are interested in the solution of large, complex scheduling problems. A "solution" as we use the term is not simply an implementation of an algorithm for solving a particular constraint satisfaction or constrained optimization problem. For many domains, constructing schedules is an extended, iterated process that may involve negotiation among competing agents or organizations, scheduling choices made for reasons not easily implementable in an automatic scheduler, and last-minute changes when events do not go as expected. In such an environment, the process by which a schedule is constructed must be considered in any attempt to provide a useful scheduler for a given domain.

In our approach, which we call *constraint envelope scheduling*, schedules are constructed by a process of "iterative refinement," in which scheduling decisions correspond to constraining an activity either with respect to another activity or with respect to some timeline. The schedule becomes more detailed as activities and constraints are added. Undoing a scheduling decision means removing a constraint, not removing an activity from a specified place on the timeline.

The assumptions underlying our scheduling work are as follows:

1. Explicitly modelling the constraints resulting from specific scheduling decisions makes the schedule easier to construct and modify.
2. Representing only those relationships required by the current set of constraints (the decisions made so far) provides a more useful picture of the current state of the scheduling effort.

The main consequence of this approach is that the scheduler does not manipulate totally-ordered timelines of activities and resource utilization. Instead, the evolving schedule consists of a partially ordered set of activities, becoming increasing ordered as additional constraints are added (or less so, as those decisions are rescinded). This approach is common to a number of scheduling systems, e.g., [1, 5, 4, 3]

Figure 1 depicts the process by which a partially ordered schedule is gradually refined into an executable, totally ordered schedule. Although providing increased flexibility (through delaying commitment), the explicit representation of partially-ordered activities in the time map makes reasoning about resource usage and other state changes more complicated. It is no longer possible to construct a single time-line representing (e.g.) changing resource availability over time. Instead, the system computes *bounds* on the system's behavior.

Despite the approximate nature of this reasoning, we are still ahead of the game: where the least-commitment approach to scheduling can at least provide approximate answers in support of scheduling decisions (e.g. what order activities should occur in), timeline schedulers make the same decisions arbitrarily—putting an activity on the timeline is a stronger commitment than constraining it to occur (say) between two other activities, or within a given time window.

STATIC SCHEDULING FOR AVIONICS

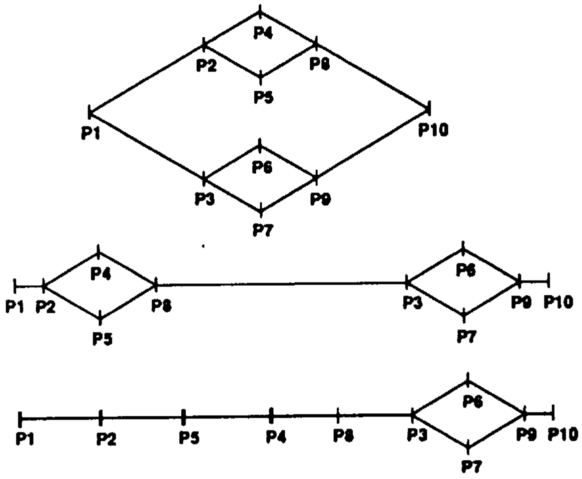


Figure 1: Gradual hardening of a partial order

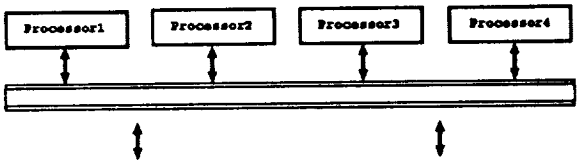


Figure 2: System architecture

One of the applications to which we have applied constraint envelope scheduling is static scheduling of processing time and bus communications in a distributed environment. This application involves safety-critical applications running on flight hardware on a commercial airplane. Figure 2 is a simple diagram of the architecture involved. The arrows at the bottom of the picture indicate that communication also occurs into and out of the cabinet in which the bus and processors reside. The schedule is static for reasons having to do with verifiability and repeatability of behavior, and ultimately with FAA certification for flight safety.

As we have already suggested, this problem is both large and complex. In a typical problem instance, there are approximately 6000 activities representing slices of processor time, and 14000 activites representing the transmission of data messages on the bus. There are six processors, which are between 80% and 90% loaded. The processes running on these processors are periodic at rates between 5 Hz and 80 Hz. This makes the problem more complicated, in that data communication is specified between processes, not between process instances. One of the decisions to be made in constructing a schedule is to determine the mapping from instances of data producers to instances of data consumers. To make matters worse, we are constructing a schedule for a 200 mS "frame" which itself runs at 5 Hz. Communication from one instance of this frame to the next is entirely legal, and so we have in some sense a circular model of time, in which constraints on activites late in the frame may affect activities early in the frame.

Processes are to a limited extent preemptible, with minimum slice times and context-dependent context-switch times (i.e., it matters who you were preempted by). Interprocess constraints include jitter (bounds on how far from perfectly periodic instances of a process may be) and latency (limits on the time between producer and consumer instances for a given data message). There are data cycles, where process A gives a message to process B gives a message to process C, which sends a message back to process A. The interaction of these cycles with latency and jitter has complex effects on schedule feasibility. In fact, much of the work that we have done on this application has been the definition and derivation of conditions under which a given set of constraints was or was not consistent.

SCHEDULING AND DYNAMIC BACKTRACKING

The scheduler we have applied to this problem uses Ginsberg's Dynamic Backtracking algorithm [2], with some minor extensions. One of these extensions was to enable the search engine to report the set of inconsistent variables involved, should it fail to find a solution. For this application, knowing what constraints are in conflict is crucial: it enables us to go back to the system designers and tell them that their requirements cannot be met.

The second extension that we made was necessitated by the nature of the scheduling problem, or at least of how we have represented it. Ginsberg's algorithm involves generating *eliminations*: explanations of why a given value for some variable is ruled out given the current partial assignment. The assumption that eliminations are available by inspection does not work for complex temporal constraints: frequently we discover that a given ordering is infeasible by trying it. Accordingly, we have extended the algorithm to handle unsuccessful attempts to assign a given value to a variable. In this case, the search engine undoes the assignment (including removing any added constraints), records an elimination explanation for that value, and reports failure back to the scheduler.

Empirically, this extended implementation of Ginsberg's algorithm has been invaluable. A typical scheduling problem involves some tens of thousands of variables representing choices on ordering, preemption or producer/consumer pairing. Given the difficulty of localizing variable interaction, sorting related variables to be close to each other is impractical or impossible. Despite considerable effort, we have not managed to find variable or value ordering heuristics that result in backtrack-free solutions (we are currently using a variant of Smith's "slack" heuristic for value ordering [6]).

For these reasons, having a search method that leaves intact that part of a partial assignment not involved in a given inconsistency is crucial. One of the ways in which we might have run into trouble using dynamic backtracking has not materialized, either: inconsistencies typically involve less than 30 variables. This means that the elimination bookkeeping is kept within bounds, as well.

There is one feature of the current algorithm which has been inconvenient, however. The requirement that it be the most recently assigned variable that is re-assigned first clashes with the fact that in scheduling applications there are frequently qualitative differences between variable types. For example, changing the ordering of an activity with respect to other activities using the same variable is in some sense a more local change to the schedule than changing the resource assigned to that activity. In the latter case, the activity must be ordered with respect to a different set of activities (those using the new resource). Any orderings remaining from the old resource assignment may now be for no purpose. For these reasons, we might like more flexible choices about variable ordering when backtracking.

CONCLUSIONS

The bottom line for this project is that we have had a successful impact on the solution of a hard problem that is a critical part of a multi-billion dollar investment. In the process of solving that problem, we have provided some empirical evidence that dynamic backtracking, suitably modified, is useful for nontrivial scheduling problems. We have also gained some useful experience in how to exploit the structure of the problem: heuristics are still critical to generating solutions or finding failures in a reasonable amount of time. "Reasonable" for this application currently means a small number of hours. Minutes would be better, days would be unworkable.

There is a lot of work yet to be done on this problem. For example, the problem is currently being solved in phases, with processor schedules being generated before the bus schedule. There are indications that heuristic repair techniques as in [7] might be useful for data scheduling.

One of the things we are hoping to arrange in the next few months is to release an instance or instances of this scheduling problem to the research community. Generation or accumulation of standard scheduling problems has been difficult. This problem has the advantages of being fairly challenging in both scale and complexity, and of having its roots in a real application.

References

[1] Fox, M.S. and Smith, S.F., ISIS: A Knowledge-Based System for Factory Scheduling, *Expert Systems*, 1(1) (1984) 25–49.

[2] Ginsberg, Matthew L., Interpreting Probabilistic Reasoning, *Proceedings of the 1985 AAAI/IEEE Sponsored Workshop on Uncertainty and Probability in Artificial Intelligence*, 1985.

[3] Muscettola, N., *HSTS: Integrating Planning and Scheduling*, Technical Report CMU-RI-TR-93-05, The Robotics Institute, Carnegie Mellon University, 1993.

[4] Sadeh, N. and Fox, M.S., Variable and Value Ordering Heuristics for Activity-based Job-shop Scheduling, *Proceedings of the Fourth International Conference on Expert Systems in Production and Operations Management, Hilton Head Island, S.C.*, 1990.

[5] Smith, S.F., Ow, P.S., Potvin, J.Y., Muscettola, N., , and Matthys, D., An Integrated Framework for Generating and Revising Factory Schedules, *Journal of the Operational Research Society*, 41(6) (1990) 539–552.

[6] Smith, Stephen F. and Cheng, Cheng-Chung, Slack-Based Heuristics for Constraint Satisfaction Scheduling, *Proceedings AAAI-93, Washington, DC*, AAAI, 1993, 139–144.

[7] Zweben, M., Deale, M., and Gargan, R., Anytime Rescheduling, *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control, San Diego*, DARPA, 1990.

N95-23762

Scheduling With Genetic Algorithms*

Theron R. Fennel, A. J. Underbrink, Jr., and George P. W. Williams, Jr.
The Boeing Company, P.O. Box 240002, M/S JN-55, Huntsville, Alabama USA 35824-6402
1+ 205.461.5133 (voice) 1+ 205.461.3201 (fax)
{*randy, al, george*}`@hsvaic.hv.boeing.com`

INTRODUCTION

In many domains, scheduling a sequence of jobs is an important function contributing to the overall efficiency of the operation. At Boeing, we develop schedules for many different domains, including assembly of military and commercial aircraft, weapons systems, and space vehicles. Boeing is under contract to develop scheduling systems for the Space Station Payload Planning System (PPS) and Payload Operations and Integration Center (POIC). These applications require that we respect certain sequencing restrictions among the jobs to be scheduled while at the same time assigning resources to the jobs. We call this general problem *scheduling and resource allocation.*

Genetic algorithms (GAs) offer a search method that uses a *population* of solutions and benefits from *intrinsic parallelism* to search the problem space rapidly, producing near-optimal solutions [10, 7]. Good intermediate solutions are probabalistically recombined to produce better offspring (based upon some application specific measure of solution fitness, *e.g.*, minimum flowtime, or schedule completeness). Also, at any point in the search, any intermediate solution can be accepted as a final solution; allowing the search to proceed longer usually produces a better solution while terminating the search at virtually any time may yield an acceptable solution.

Many processess are constrained by restrictions of sequence among the individual jobs. For a specific job, other jobs must be completed beforehand. While there are obviously many other constraints on processes, it is these on which we focussed for this research: how to allocate crews to jobs while satisfying job precedence requirements and Personnel, tooling and fixture (or, more generally, *resource*) requirements.

WHY A GENETIC ALGORITHM MAKES SENSE

There are a number of reasons why we wanted to explore using genetic algorithms for this scheduling work. While some existing approaches may suffice for basic scheduling, we were also interested in the possibility of global scheduling for complex processes and large assemblies. For example, Space Station experiment payloads that must be scheduled in a 90 day increment may number in the thousands; we cannot truly optimize an increment schedule by restricting our scope to a day or week. Therefore, a solution to our application requires the following characteristics:

- Evidence of scalability: There is considerable evidence that GAs have better scalability characteristics compared to other techniques commonly used for similar problems [14].
- Ease of parallelization: GAs broken into sub-populations with limited communication between them often exhibit super-linear speedup. This effect also has been shown in loosely coupled computers, communicating asynchronously over a network [18].
- Multi-objective optimization: we wanted to combine measures of schedule duration and completeness with resource utilization and task priorities.

OUR APPROACH

We developed a genetic algorithm which satisifies temporal constraints to produce near-optimal schedules with resources assigned to jobs. Our scheduler pre-processes the temporal constraints to eliminate implied or redundant constraints (*e.g.*, transitive constraints that may be specified explicitly) and evolves a population of schedules until termination criteria are met.

Constraint Preprocessing

There are two pre-processing steps before the GA-based scheduler is run:

1. First, we simplify the temporal and precedence constraints by removing redundancy and resolving obvious conflicts
2. Then we derive a partial ordering of the jobs similar to finding a critical path. This partial ordering is used for chromosome repair (see below) and can also establish a lower bound on the duration of the schedule.

Problem Encoding

The chromosomal encoding of schedules is a two-chromosome scheme [12]: one chromosome for the job sequence and one chromosome for the resource allocation. They are described as follows:

chromosome χ_0: An ordered sequence of jobs, coded as J job numbers.

chromosome χ_1: A set of binary coded fields, each of which represents the specific resource which will be used on the job associated with the field.

This encoding scheme effectively allows us to treat the job sequence and resource assignment as two subproblems. Each can be manipulated separately but optimized together.

The Genetic Plan

The term 'genetic plan' identifies the overall approach used for evolving populations of (genetically encoded) schedules. Our basic approach enlists a 'classical' Holland-style generational GA. We employ optional elitism, which is only engaged when the score of the best-ever schedule is not matched in the current generation.

We found ranking selection to be superior to the other techniques we tried with the most fit individual receiving ~1.2 copies in the next generation. This rather low selection pressure was necessary to prevent premature convergence on some of the more difficult problems.

Our approach to genetic operator application treated reproduction, mutation, and recombination each as independent foreground operators, rather than making mutation a background operator which could potentially mutate the product of recombination and reproduction.

Genetic operations at the chromosome level were also kept independent. Once a decision was made to perform recombination or mutation, a second decision was then necessary to determine which chromosome (χ_0 or χ_1) should be manipulated. This decision was biased by the relative sizes of the chromosomes, *i.e.*, the longer chromosome was assigned a proportionally greater probability.

Genetic Operators

Since the genetic representation is distributed between two chromosomes with fundamentally different characteristics, different genetic operators were required for each chromosome. For the job-sequence chromosome (χ_0), the best recombination operator we found was the Partially Mapped Crossover (PMX) [8], though we also tried Random Respectful Recombination (R^3) [16], and Linear Order Crossover (LOX) [4]. For the resource-allocation chromosome (χ_1), the best recombination operator we found was Uniform Crossover (UX) [19], though we also tried conventional one- and two-point crossover. UX is generally considered to be quite disruptive, but since the ordering of fields in the resource chromosome does not attempt to group related fields (assuming this were even feasible), there is little locality to be preserved.

For the job-sequence chromosome, mutation swaps the alleles from two loci in the chromosome, where the first locus is the current locus and the second is either the next (adjacent) locus (50%) or another locus chosen randomly (50%). For the resource-allocation chromosome, mutation selects a random allele value, which effectively halves the mutation rate when compared to bit-flipping mutation.

The Schedule Builder

The schedule builder is responsible for decoding the chromosomes and converting them into a feasible schedule. The basic GA had a very difficult time finding *any* feasible solutions for highly-constrained scheduling problems. We therefore *enforced* feasibility in our schedules by minimally reordering jobs to accommodate precedence constraints.

Chromosome Repair

The basic idea behind chromosome repair is to use heuristic or algorithmic techniques to modify individual solutions and then to probabalistically modify the genetic information to incorporate these changes [4, 15]. In some respects, repair might be viewed as an intelligent mutation operator. Sometimes the repair corrects an illegal chromosome to make it legal (as in our schedule builder), while other times it simply improves a previous legal schedule. Our system implements both kinds of repair with variable degrees of probability, generally 5–10%. This has the effect of enriching the population with good partial solutions which can then be combined via crossover.

Since our implementation has two chromosomes (χ_0 and χ_1), we have at least two opportunities to implement chromosome repair.

Repairing the Resource Allocation Chromosome

This repair strategy ignores the previous genetic information from the resource allocation chromosome and determines a resource allocation from scratch. This is done using a greedy approach to incrementally allocate the best resources for each job, backtracking when there are conflicts preventing all the demands for the job from being satisfied.

Repairing the Job Sequence Chromosome

There are two 'levels' of repair for the job sequence chromosome. The first level repairs the chromosome to reflect the results of the schedule builder. The second level of repair is only invoked some fraction of the times the first level is invoked and causes the job sequence to be modified *before* the schedule builder is invoked. The second level repair is heuristic and simplifies the task of constructing a feasible schedule for the schedule builder (first level repair).

The nature of the second level repair is based on the partial order on the jobs from precedence and temporal constraints. This partial ordering specifies a start time for each job, which would produce a feasible schedule if adequate resources were available to satisfy any resource request. This assumption of (essentially) infinite resources has led us to call this partial ordering an 'infinite resource model' (IRM) of the schedule. When there are many precedence or temporal constraints, this IRM *may* contain a great deal of useful information, especially since highly constrained schedules are the most difficult ones for the GA to solve. Similarly, if there are few (or no) such constraints, the IRM doesn't help very much. But what help it does provide is exactly where the GA needs help, *i.e.*, in repositioning constrained jobs in the job sequence where they can be (feasibly) scheduled.

Schedule Evaluation

We explored a fairly large variety of composite evaluation functions. We defined several different evaluation criteria and finally settled on a particular combination which seems to work reasonably well for the problems we have tried. The individual criteria are separate, independently computable functions and their resulting values are combined by a higher level function which supports adjusting the weights of the individual criteria. The set of criteria in our final evaluation function are:

- Schedule Duration: The number of time units (*e.g.*, hours or minutes) scheduled to complete the jobs.
- Resource Utilization: The ratio of resource time scheduled to the schedule duration.
- Schedule Completeness: The ratio of jobs scheduled to the total number of jobs (*i.e.*, a legal schedule may not include all jobs).
- Priority: A weight score accumulating higher values for higher priority jobs.

FUTURE WORK

Considerable work remains before we can determine the true value of this approach to scheduling. A primary requirement for a better understanding would have to be more detailed comparisons against other algorithms, including a more elaborate set of benchmark tests. We would also like to implement this approach on a parallel architecture and test this implementation on some very large problems.

We would also like to explore the use of Pareto optimal selection strategies to better support multi-objective optimization. These are based on non-dominance of solutions and appear to better support multi-objective optimization. [5, 13]. Finally, we would like to compare our multiple-chromosome approach to a single chromosome implementation and determine the value (if any) of multiple chromosomes *per se*.

CONCLUSIONS

We developed a genetic algorithm for scheduling and resource allocation. We employed several interesting GA features, including a multiple-chromosome schedule encoding, multiple repair strategies, and several order-preserving operators.

A significant consequence of chromosome repair was that we found post-GA hill-climbing unnecessary. Since any improvements made via chromosome repair are then available to the GA, which can potentially improve upon them further, we opted to include these heuristic techniques in the chromosome repair strategies. Use of repair at higher probablilities leads to premature convergence of the population to relatively poor solutions, providing evidence that good solutions are not solely the result of repair.

In our tests, the scheduling algorithm creates schedules which are as good as or better than the results from a critical-path scheduler currently in use within the company. Additionally, the scheduler is able to schedule general resources more efficiently than the critical path scheduler.

Our limited test results encourage us to continue developing the genetic algorithm scheduler to include more schedule evaluation criteria. We also hope to explore the possibility of large-scale scheduling for manufacturing processes.

REFERENCES

[1] Richard K. Belew and Lashon B. Booker, editors. *Proccedings of the Fourth International Conference on Genetic Algorithms*, San Mateo, California, 1991. Morgan Kaufmann.

[2] D. E. Brown and C. C. White, editors. *Operations Research/Artifical Intelligence: The Integration of Problem Solving Strategies.* Kluwer Academic Publishers, Boston, 1990.

[3] Lawrence Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[4] E. Falkenauer and S. Bouffouix. A genetic algorithm for job shop. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* [11], pages 824-829. Sacramento, California.

[5] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In Forrest [6], pages 416–423.

[6] Stephanie Forrest, editor. *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, 1993. Morgan Kaufmann.

[7] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, New York, 1989.

[8] David E. Goldberg and Robert Lingle, Jr. Alleles, loci, and the traveling salesman problem. In Grefenstette [9], pages 154–159.

[9] John J. Grefenstette, editor. *Proceedings of the First International Conference on Genetic Algorithms and their Applications.* Lawrence Erlbaum Associates, 1985.

[10] John H. Holland. *Adaptation in Natural and Artificial Systems.* The MIT Press, Cambridge, 1991. Revised from the 1975 edition.

[11] IEEE. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, April 1991. Sacramento, California.

[12] Kate Juliff. A multi-chromosome genetic algorithm for pallet loading. In Forrest [6], pages 467–473.

[13] Sushil J. Louis and Gregory J. E. Rawlins. Pareto optimality, ga-easiness and deception. In Forrest [6], pages 118–123.

[14] H. Mülenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. In Belew and Booker [1], pages 271-278.

[15] David Orvosh and Lawrence Davis. Shall we repair? genetic algorithms, combinatorial optimization, and feasibility constraints. In Forrest [6], page 650.

[16] Nicholas J. Radcliffe. Forma analysis and random respectful recombination. In Belew and Booker [1], pages 222–229.

[17] J. David Schaffer, editor. *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California, 1989. Morgan Kaufmann.

[18] R. Shonkwiler. Parallel genetic algorithms. In Forrest [6], pages 199–205.

[19] Gilbert Syswerda. Uniform crossover in genetic algorithms. In Schaffer [17], pages 2-9.

N95- 23763

Integrating Planning and Reactive Control

David E. Wilkins Karen L. Myers
SRI International Artificial Intelligence Center
333 Ravenswood Ave. Menlo Park, CA 94025
phone: 415-859-2057 fax: 415-859-3735
{myers, wilkins}@ai.sri.com

KEY WORDS AND PHRASES

planning, reactive control, intelligent agents, replanning, uncertain reasoning

INTRODUCTION

Our research is developing persistent agents that can achieve complex tasks in dynamic and uncertain environments. We refer to such agents as *taskable, reactive agents.* An agent of this type requires a number of capabilities. The ability to execute complex tasks necessitates the use of strategic plans for accomplishing tasks; hence, the agent must be able to synthesize new plans at run time. The dynamic nature of the environment requires that the agent be able to deal with unpredictable changes in its world. As such, agents must be able to react to unanticipated events by taking appropriate actions in a timely manner, while continuing activities that support current goals. The unpredictability of the world could lead to failure of plans generated for individual tasks. Agents must have the ability to recover from failures by adapting their activities to the new situation, or replanning if the world changes sufficiently. Finally, the agent should be able to perform in the face of uncertainty.

Many domains of interest require problem-solving agents with the capabilities described above. Military and space operations provide good examples. Certainly one would not engage in an undertaking such as Desert Storm or repairing the Hubble Space Telescope without first formulating a strategic mission plan. Reactive response and failure recovery are necessary because unexpected equipment failures, weather conditions, enemy actions, and other events may require changes to the overall strategic plan.

The Cypress system, described here, provides a framework for creating taskable, reactive agents. Several features distinguish our approach: (1) the generation and execution of complex plans with parallel actions, (2) the integration of goal-driven and event-driven activities during execution, (3) the use of evidential reasoning for dealing with uncertainty, and (4) the use of replanning to handle run-time execution problems.

Our model for a taskable, reactive agent has two main intelligent components, an *executor* and a *planner.* The two components share a library of possible actions that the system can take. The library encompasses a full range of action representations, including *plans*, *planning operators*, and *executable procedures* such as predefined standard operating procedures (SOPs). These three classes of actions span multiple levels of abstraction.

The executor is always active, constantly monitoring the world for goals to be achieved or events that require immediate action. In accord with its current beliefs and goals, the executor takes actions in response to these goals and events. Appropriate responses include applying SOPs stored in the action library, invoking the planner to produce a new plan for achieving a goal, or requesting that the planner modify a previous plan during execution. The planner should be capable of synthesizing sophisticated action sequences that include parallel actions, conditional actions, and resource assignments. The planner plans only to a certain level of detail, with the executor taking that plan and expanding it at run time by applying appropriate library actions at lower levels of abstraction.

CYPRESS

Cypress constitutes a framework in which to define taskable, reactive agents based on the above model. The architecture of Cypress is depicted in Figure 1.

The motivation for Cypress was to build a heuristically adequate system for use in practical applications. To this end, Cypress relies on mature, powerful planning and execution technologies, namely the SIPE–2 generative planner [5] and the PRS-CL reactive execution system [5]. We have applied Cypress to a number of demanding problems, including real-time tracking, fault diagnosis on the Space Shuttle, production-line scheduling, and military operations [5].

PRS-CL is a framework for constructing persistent, real-time controllers that perform complex tasks in dynamic environments while responding in timely fashion to unexpected events. It has been used to monitor the Reaction Control System (RCS) of the Space Shuttle [5]. This application illustrates the use of multiple agents, and has been used to detect and recover from most of the possible malfunctions of the RCS, including sensor faults, leaking components, and regulator and jet failures. The system demonstrated guaranteed response, support for asynchronous inputs, interrupt handling, continuous operation, and handling of noisy data.

SIPE–2 is a partial-order AI planning system that supports planning at multiple levels of abstraction. It has the properties required by our agent model, including the ability to generate plans that include parallel actions, conditional actions, resource assignments, and the ability to modify previously generated plans. In contrast to most AI planning research, heuristic adequacy has been a primary design goal of SIPE–2.

PRS-CL and SIPE–2 employ their own internal representations for plans and actions for efficiency. For this reason, Cypress supports the use of an *interlingua* called the ACT formalism [5] that enables these two systems to share information. ACT provides a language for specifying actions and plans for both planners and executors. Cypress includes translators that can automatically map Acts onto SIPE–2 and PRS-CL structures, and one that can map SIPE–2 operators and plans into Acts. Using the ACT interlingua, PRS-CL can execute plans produced by SIPE–2 and can invoke SIPE–2 in situations where run-time replanning is required. The ACT-Editor subsystem supports the graphical creation and display of Acts. Gister-CL [5] implements a suite of evidential reasoning techniques that can be used to analyze uncertain information about the world and possible actions. For example, Gister-CL can be used to reason about uncertain information in order to choose among candidate Acts in either the planner or executor.

In contrast to many other agent architectures, planning and execution operate asynchronously in Cypress, in loosely coupled fashion. This approach makes it possible for the two systems to run in parallel, even on different machines, without interfering with the actions of each other. In particular, PRS-CL remains responsive to its environment during plan synthesis. While the subsystems of Cypress can function independently, Cypress is used most advantageously as an integrated framework that supports a wide range of planning and execution activities.

APPLICATIONS

An example from military operations planning [4] is currently the only implemented application that illustrates the use of all sub-

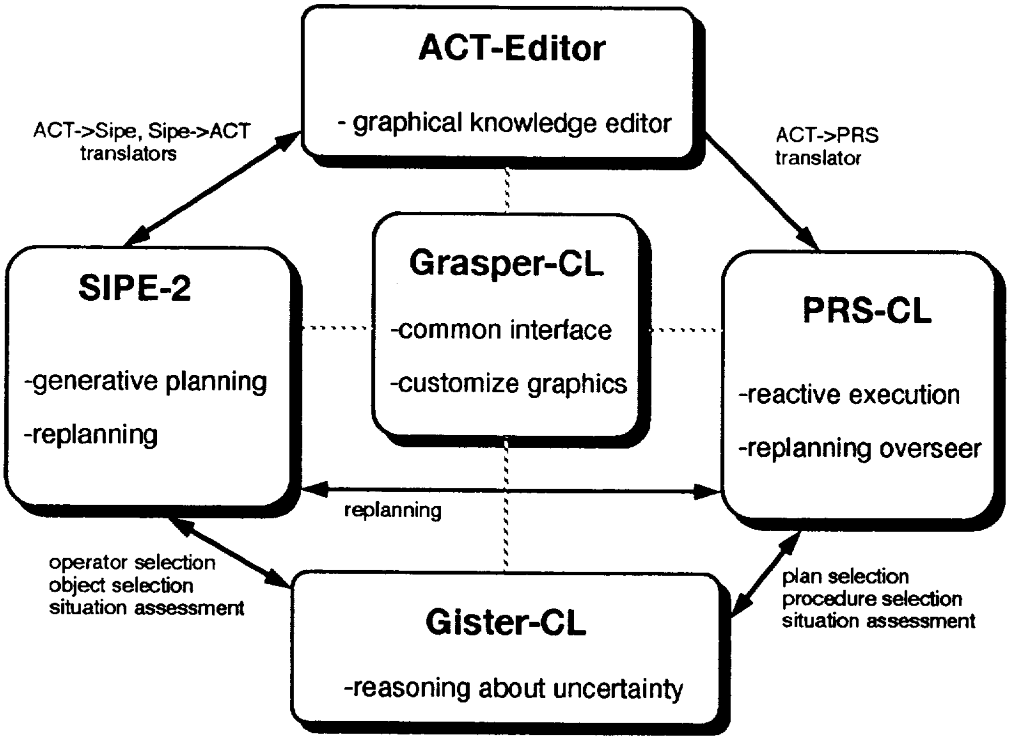


Figure 1: The Architecture of Cypress

systems of Cypress, but it is similar to a space mission. The most advantageous use of Cypress in space applications will most likely be in situations that do not directly involve humans. A planetary rover will certainly need the combination of plan-directed behavior with reactive response to the environment provided by Cypress, and can build directly on our use of Cypress modules to control an indoor mobile robot. Other appropriate space applications include control of a satellite or probe, controlling experiments on the shuttle or space station, and providing an assistant to astronauts to handle routine malfunctions and alert them of important events that affect the overall mission plan.

The military application domain knowledge includes approximately 100 plan operators, 500 objects with 15 to 20 properties per object, and 2200 initial predicate instances. Plans range in size from several dozen to 200 actions, including many that are to be executed in parallel [4].

The scenario begins with a goal request for deterring several military threats. SIPE–2 uses a set of Acts previously input to the system to generate a plan with many threads of parallel activities. During the planning process, Gister-CL assists SIPE–2 in choosing appropriate military forces for particular missions, by analyzing uncertain information about the situation. Throughout the planning process, PRS-CL monitors the world for additional goals and events that might require immediate action. PRS-CL executes the plan by applying appropriate Acts to refine the plan to lower levels of abstraction, eventually bottoming out in actions that are executable in the world.

PRS-CL responds to many unexpected events by applying Acts representing SOPs. Sometimes an event causes an execution failure that cannot be repaired by any defined Acts (e.g., if transit approval is rescinded for air space that is being used). PRS-CL then invokes a second PRS-CL agent to issue a replanning request to SIPE–2. Meanwhile, the first agent continues execution of parallel threads of the plan not affected by the failure. The planner modifies the plan by eliminating actions that use the air space in question and replacing them with an alternative mobilization. The actions in the new plan are selected so as not to interfere with the continuing execution of other actions in the original plan. The new plan is sent to to the first agent, which integrates the new plan with its current activities and continues.

In a similar fashion, a Cypress agent controlling a planetary rover would have the executor handle unexpected obstacles in its path, and call the planner to modify the plan when progress can no longer be made in the desired direction. On a satellite, the executor could continue to monitor spacecraft systems while requesting the planner to modify the plan for transmitting pictures back to earth after a failure in one of the transmitters.

CONCLUSION

Cypress is a powerful framework in which to define agents that must accomplish complex goals in dynamic and unpredictable environments. The application of Cypress to the military domain and to the Space Shuttle's RCS (only the PRS-CL subsystem is used) attests to the system's usefulness.

The asynchronous replanning facility constitutes one important technological advance, providing flexible plan execution that can adapt to significant unexpected changes in the world. An interesting technical problem that had to be solved was the design of ACT as a common representation for both executors and planners. PRS-CL had to be extended in numerous ways to support the execution of plans employing constructs not found in the domain procedures defined for previous PRS-CL applications.

Several characteristics distinguish Cypress from other systems that provide both planning and reactive execution. Many systems do not use general-purpose planning and so cannot generate plans of sufficient complexity for interesting applications. Previous work in run-time replanning has either been limited to synchronous approaches [2] or focuses on local, adaptive modifications to rule sets, rather than employing the full look-ahead reasoning of a planner [3, 1]. The ability to modify a complex, parallel plan at run time and adapt execution activity to the new plan is, to our knowledge, a new accomplishment.

ACKNOWLEDGMENTS

This research was supported by the ARPA and Rome Laboratory Planning Initiative (ARPI) under Contract F30602-90-C-0086. The military application domain is an extended version of the one used for the second IFD in ARPI. The RCS application was supported by NASA Ames Research Center and was done by a team led by Michael Georgeff. SIPE–2, PRS-CL, Gister-CL, Grasper-CL, and Cypress are trademarks of SRI International.

References

[1] R. J. Firby. An investigation into reactive planning in complex domains. In *Proceedings of the 1987 AAAI*, pages 202–206, AAAI, Menlo Park, CA, 1987.

[2] J.E. Laird. Integrating planning and execution in Soar. In *Proceedings of the AAAI Spring Symposium*, 1990.

[3] D. M. Lyons and A. J. Hendricks. A practical approach to integrating reaction and deliberation. In *First International Conference on AI Planning Systems*, pages 153–162, College Park, Maryland, 1992.

[4] David E. Wilkins and Roberto V. Desimone. Applying an AI planner to military operations planning. In M. Fox and M. Zweben, editors, *Intelligent Scheduling.* Morgan Kaufmann, San Mateo, CA, 1994.

[5] David E. Wilkins, Karen L. Myers, John D. Lowrance, and Leonard P. Wesley. Planning and reacting in uncertain and dynamic environments. *Journal of Experimental and Theoretical AI*, to appear.

N95- 23764

Automatic Generation of Efficient Orderings of Events for Scheduling Applications

Robert A. Morris
Florida Institute of Technology
Computer Science Program
150 W. University Blvd.
Melbourne, FL 32901
phone: (407) 768-8000, ext. 7290 *fax*: (407) 984-8461
email:morris@cs.fit.edu

KEY WORDS AND PHRASES

Scheduling, similarity-based clustering

PROBLEM STATEMENT

In scheduling a set of tasks, it is often not known with certainty how long a given event will take. We call this *duration uncertainty.* For example, as part of the task of making a telescope observation, the telescope must be accurately centered on a star. The time required to perform this subtask cannot be accurately predicted, since it depends on factors which vary from execution to execution (e.g., the position of the telescope at the start of the execution of this task).

Duration uncertainty is a primary obstacle to the successful completion of a schedule. If a duration of one task is longer than expected, the remaining tasks are delayed. The delay may result in the abandonment of the schedule itself, a phenomenon known as schedule *breakage.* One response to schedule breakage is on-line, *dynamic rescheduling.* A more recent alternative is called *proactive* rescheduling [2]. This method uses statistical data about the durations of events in order to anticipate the locations in the schedule where breakage is likely *prior* to the execution of the schedule. It generates alternative schedules at such sensitive points, which can be then applied by the scheduler at execution time, without the delay incurred by dynamic rescheduling.

This paper proposes a technique for making proactive error management more effective. The technique is based on applying a *similarity-based method of clustering* to the problem of identifying similar events in a set of events. The remainder of this paper consists of a discussion of the following:

1. The intuitions underlying the technique;
2. The way in which clustering techniques from the AI literature can be applied to the problem of managing duration uncertainty in scheduling;
3. The requisite assumptions about the domain for applying the technique; and
4. An implementation strategy.

INTUITIONS

The set of events under consideration have occurrences which need to be scheduled. The goal is to find an ordering of these occurrences which minimizes the amount of expected duration uncertainty associated with each. The knowledge used to find the ordering comes from observations of repeated past occurrences of the same events. Figure 1 represents a repeated occurrence of an event E. E recurs 4 times over a stretch of time. Duration uncertainty is depicted visually as the difference in the

Figure 1: A repeating event

Figure 2: Pairing an event with a similar event

lengths of each line representing a single occurrence. We assume that the events under consideration all have the tendency to exhibit duration uncertainty.

The heuristic being formalized here is that duration uncertainty can often be reduced by assigning an event in such a way that it is in temporal proximity to a *similar event.* A mundane example will illustrate. Suppose I am scheduling my daily household chores. I find that I must complete three tasks: clean the kitchen (K), clean the bathroom (B) and work in the garden (G). I can do these in any order; my main constraint is to finish all three within a certain time frame. One is clearly led to a plan to perform K and B together, either before or after G. Why? The tasks are similar, either in that they are both cleaning tasks, or perhaps also because they are indoor tasks.

How does the act of scheduling similar events in close temporal proximity lead to a reduction of duration uncertainty? Intuitively, actions are sometimes similar because they share a number of *stages.* For example, any cleaning room action consists of a *preparation stage* consisting of getting the mop or broom, getting floor cleaner, water, bucket, etc. If I perform the cleaning room actions together, say $K \rightarrow B$ (clean the kitchen followed by clean the bathroom), the preparation stage of B will not be required (or be simplified). Since the duration of any action is the sum of the durations of its stages, the duration uncertainty of the whole will be a similar function of the duration uncertainty of the different stages. It follows that I should be able to more accurately predict how long the bathroom cleaning will take *when preceded by the kitchen cleaning action* than I could predict its duration in isolation, or when preceded by a dissimilar event. This conclusion is justified by noting that the preparation stage, in such a situation, does not exist; hence, trivially, there is no uncertainty associated with it, which reduces the uncertainty of the whole event. Graphically, this can be represented as in Figure 2. This figure represents the expected durations of kitchen events when paired with the similar, bathroom cleaning event. On the other hand, if paired with a dissimilar event (e.g. gardening), one would expect K to behave as in Figure 1.

In ordering mundane events, we implicitly bring to bear the ability to apply concepts which cluster events into similarity classes. This paper addresses the same problem when such *a priori* conceptual knowledge about a domain is lacking. For example, in the telescope scheduling domain, it may be difficult or impossible to classify *a priori* whether two tasks to be scheduled are similar or not. The main contribution of this paper is to suggest that there is *a posteriori* knowledge (knowledge gained from experience) that can be used to infer the similarity of events.

COMPUTATIONAL MODEL

The computational problem to be solved can be stated as follows: given a set E of k events, find an ordering $E_1 \rightarrow E_2 \rightarrow \ldots \rightarrow E_k$ of all the elements in E which minimizes the expected duration uncertainty over all members of E. The previous section justified the intuition that some orderings of events will exhibit less duration uncertainty than others. In this section, a technique for finding these preferred orderings will be presented.

Similarity Based On Relative Durations of Events

Based on observations in the previous section, the notion of similarity between two events e and e' can be induced from observations of the durations of each event when they are placed in close temporal proximity.

Definition 1 *The* relative duration *of* e *with respect to* e' *(*$rd(e, e')$*) is the duration of* e *when* e *immediately follows* e'*. The* relative average duration *of an event* e *with respect to an event* e' *is the average duration of* e *when immediately followed by* e'*, over a set of occurrences of* e *and* e'.

$rd(e, e')$ can be viewed as a discrete random variable, associating a duration with the outcome of pairing the two events. Let $\sigma_{rd(e,e')}$ denote the standard deviation of $rd(e, e')$. It is then possible to define the notion of relative similarity between triples of events e_1, e_2, e_3:

Definition 2 e_1 *is* at least as similar *to* e_2 *as to* e_3 *if* $\sigma_{rd(e_1,e_2)} \leq \sigma_{rd(e_1,e_3)}$.

An absolute concept of similarity can be defined when a similarity *threshold* is postulated. Let θ be such a threshold. Then:

Definition 3 *Let* e *and* e' *be events. Then* e is similar to e' *if* $\sigma_{rd(e,e')} \leq \theta$.

Any similarity relation is reflexive, symmetric, and intransitive. The claim here is that comparing the value of $\sigma_{rd(e_1,e_2)}$ to a threshold can be viewed as applying a similarity relation. Clearly, reflexivity and intransitivity are satisfied. By definition, symmetry implies that if $\sigma_{rd(e,e')} \leq \theta$, then $\sigma_{rd(e',e)} \leq \theta$. Reflections from intuition should make this assumption plausible. Recall that the postulated reason for reduction of duration uncertainty when events are paired to similar events is that they share a stage, which is eliminated or simplified when the events are paired together, Clearly, the ordering of the pairing is irrelevant. For example, whether $K \rightarrow B$ or $B \rightarrow K$, the duration uncertainty of the later event will be reduced. Hence, it is reasonable to assume that similarity, defined in the previous definition, is symmetric.

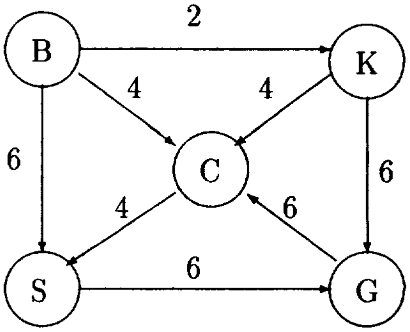


Figure 3: A σ-Graph For Five Events

Relation to Clustering Methods

In order to reduce duration uncertainty in an error management system for scheduling, events should be ordered in a way that similar events are clustered. The *similarity-based* clustering method [3] is a weak AI method which can be employed to generate efficient orderings. The computational problem of interest here can be viewed as an instance of *one-dimensional clustering.* For such a problem, the goal is to reduce the number of distinct values of a set of variables by identifying near-equivalence classes of values based on similarity. To briefly illustrate the technique of clustering, we introduce a data structure called a σ-graph:

Definition 4 *A* σ*-graph is a weighted directed graph with the following characteristics. Each vertex is labeled by one of the elements in a set* E*. Each directed edge* (e_i, e_j) *between source* e_i *and target node* e_j *is labeled with a value representing the degree of similarity between* e_i *and* e_j.

To illustrate, consider a slightly more complex mundane example. Now there are five events, including K, B and G, as before, but also including the tasks *wash car* (C) and *go to store* (S). An incomplete σ-graph for this set of events is found in Figure 3. Here, the lower the value on an arc, the greater the degree of similarity between the two events.

Clustering techniques are traditionally used for automating concept formation. One clustering method (called *agglomeration*), fuses entities to form groupings based on the threshold of minimum similarity. The fusion process stops when all values exceed the threshold. For example, if the threshold is assumed to be 2, the result of the agglomerative process applied to the example would fuse B and K into a cluster. For our purposes, however, clustering is a means to an end, viz., to generate an ordering of events which reduces the amount of duration uncertainty with which a proactive scheduling error manager needs to contend. The following section describes how similarity-based clustering can be implemented for this purpose.

Implementation and Intended Use

The procedure for generating efficient orderings of events based on relative durations is intended to be used as a preprocessing stage in a proactive error management system for scheduling. The stage can be viewed as one that deletes from the set of possible orderings those which exhibit the most duration uncertainty.

Assume as input a set E of k events. The set E has been executed up to m times in some or all of the $k!$ permutations of the orderings of the events in E. Assume an ordering of these permutations and executions. Let $rd(E_i, E_j)[p, q]$ represent the duration of E_i when immediately followed by E_j on the p^{th} occurrence of the q^{th} permutation of E; thus $1 \leq p \leq m$ and $1 \leq q \leq k!$. This yields a set of $O(k![m(k-1)])$ values of $rd(E_i, E_j)[p, q]$ for each pair $E_i, E_j \in E$. From this data, an ordering of a set E of events which minimizes duration uncertainty is based on the following steps:

1. For each E_i in E, compute the mean of the set $\{rd(E_i, E_j)[p, q] : 1 \leq p \leq m, 1 \leq q \leq k!\}$, and $\sigma_{rd(E_i,E_j)}$, for each pairing of E_i with other $E_j \in E$;
2. Form a σ-graph with E the set of vertices and for each pair E_i, E_j in E, there is an arc labeled with the value of $\sigma_{rd(E_i,E_j)}$; and
3. Apply an all-pairs shortest-path algorithm [1], such as Floyd-Warshall, to generate an ordering of the events.

For example, assume that Figure 3 represents the result of completing step 2 in the procedure. Thus, the labels on the arcs represent the standard deviations of the relative durations of the event occurrences connected by the arc. If the claims made in this paper are plausible, then such values would be the kind expected, since they reflect the intuitive degree of similarity among the events. Then, the result of applying step three would yield

$$B \rightarrow K \rightarrow C \rightarrow S \rightarrow G$$

as well as other orderings which are minimal with respect to duration uncertainty.

An example of a proactive scheduling system which might benefit from the account presented here is the *Just-In-Case* (*JIC*) error management technique described in [2]. This technique analyzes a schedule of telescope observations for possible execution breaks. For the break point with the highest probability of occurrence, the system forms a contingent alternative schedule. *JIC* utilizes duration uncertainty measures to calculate the possible schedule break points. As a preprocessing stage to the error management procedure, the three stage method presented in this section could be applied to discriminate among different orderings of the events, selecting the ones which minimize duration uncertainty. This would reduce the amount of anticipated break points with which the error manager has to contend.

ASSUMPTIONS AND LIMITATIONS

To be of optimal benefit for its intended use, the events to be analyzed by the method should possess the following properties:

1. The events in E should be causally independent; this means at least that:
    - No occurrence E_i in E prohibits the execution of any other E_j; and
    - No occurrence E_i presupposes the execution of some other E_j;

   and
2. Each of the events in E has the tendency to exhibit duration uncertainty; this means that, considered in isolation, the standard deviation of the duration of each event is high.

Even with these minimum assumptions, $\sigma_{rd(E_i,E_j)}$ is a coarse measure of event similarity. For example, assume E_i consists of the stages A, B and C, and E_j consists of A,E, and F. Assume that the duration uncertainty of E_j is caused completely by stage F. Then, the approach proposed here would fail to recognize that the two events are similar (in the sense of sharing a common stage A), since E_j would not demonstrate a reduction of duration uncertainty when paired with E_i. In such a case, it would be useful to view the *absolute* reduction in mean duration as evidence for its similarity to E_i. That is, since E_j shares a stage with E_i, its pairing with E_i should result in a reduction of the time it takes to execute. Hence, it may be the case that both mean duration and standard deviation should be viewed as the measure of similarity. This could be easily added to the implementation by including mean duration as part of the labels on the arcs of the σ-graph. The addition would imply a two dimensional description space for the events, and a similarity concept based on a *vector* of attributes.

There may be other forms of causal interaction which would make the ordering produced by this procedure less preferred than others. [1] Consider for example events E_i and E_j again. Perhaps the pairing $E_i \rightarrow E_j$ would result in a reduction of the standard deviation of the duration of E_j, and hence be preferred by the proposed model. However, it is possible that this pairing would increase the *absolute* duration of E_j.

CONCLUSION

This paper has offered an approach for aiding proactive error management techniques for scheduling. The idea is to use statistical temporal information about event occurrences to induce similarities among these occurrences, when conceptual information about the same events is unavailable. Pairing similar events in close temporal proximity can often reduce the uncertainty in the expected duration of the events. This leads to the potential for a reduction in the amount of rescheduling required by the proactive error manager.

References

[1] Cormen, T.; Leiserson, C.; and Rivest, R., 1991. *Algorithms*. Cambridge, Mass.: MIT Press.

[2] Drummond, M.; Bresina, J.; Swanson, K., 1994. Just-In-Case Scheduling. In *Proceedings of AAAI-94*, forthcoming.

[3] Michalski, R. S.; and Stepp, R. E., 1987. Clustering. In *Encyclopedia of Artificial Intelligence, S. Shapiro, ed. Cambridge, Mass: MIT Press.*

---

[1] This point was made by one of the reviewers.

N95- 23765

Terrain Modelling and Motion Planning for an Autonomous Exploration Rover

F.Richard[+], S. Benoliel[*], O. Faugeras [**], P. Grandjean[***], M. Hayard[+], T.Siméon[++]

[+] Alcatel Espace 26 ave. J.F. Champollion BP 1187 - 31037 Toulouse - FRANCE
Phone : 61 19 52 13 - E-mail : frdk8bq8@ibmmail.COM
[++] LAAS 7 ave. Cl. Roche - 31077 Toulouse - FRANCE
[*] Alcatel Alsthom Recherche Route de Nozay - 91460 Marcoussis - FRANCE
[**] INRIA 2004 route des Lucioles 06902 Sophia Antipolis FRANCE
[***] CNES 18 ave. E. Belin - 31055 Toulouse - FRANCE

KEY WORDS AND PHRASES

Map generation, Motion planning, planetary exploration, robotics.

INTRODUCTION

To assess the feasibility of planetary exploration missions, using rovers, the French national agency CNES, with a consortium of European laboratories and industrial concerns, has initiated the Eureka project[1] "Illustration of an Autonomous Robot for the Exploration of Space" (IARES). IARES is a demonstrator composed of a rover and a ground station, linked by telemetry and telecommand. It is aimed at verifying, on earth, robotic concepts developed by the RISP group of French laboratories (LAAS, INRIA, CERT, LETI) to perform scientific missions such as autonomous terrain sample collecting over large areas. To cope with the actual needs of planet exploration, IARES suitability is assessed through constraints on limited bandwidth, time delay and on-board resources. This autonomy relies heavily on robust on-board trajectory generation capabilities.

A large amount of work exists today in the field of autonomous navigation, but most of it is related to indoor navigation in a rather structured environment. In natural terrain there are a limited number of experiments : see for example J.P.L.'s Robby. [1], crossing a dry river bed, or C.M.U.'s Ambler, [1], using legged locomotion and Eden developed at L.A.A.S. [2]. The IARES approach proposed in this paper follows partly from the work done in the Eden experiment.

This paper presents the main functions of the IARES navigation sub-system and shows how they are combined to allow movement in Mars like environments. Section 2 gives an overall description of the IARES system. Section 3 details the functions of the Navigation sub-system, and finally section 4 illustrates with a simple example the use of these functions.

GENERAL DESCRIPTION

The rover (see figure 1) is a 6 wheeled vehicle derived from the Marsokhod concept [3]. It has high cross-country abilities and offers two modes of locomotion : the wheel mode and the wheel walking mode (peristaltic motion) for sandy terrains. The equipment includes an inertial reference system, and a solar sensor for localisation, stereo cameras and a 3D laser imager for terrain perception. The maximum speed of about 0.25 m/s should allow daily displacements of about 1 km.

The on-board control system of the rover is organized around an Embarked central Decisional Structure (EDS), which is

---

[1] The work described in this paper was carried out by Alcatel Espace, Alcatel Alsthom Recherche, LAAS and INRIA under contract with Centre National d'Etudes Spatiales within the framework of the NO 969 EUREKA IARES project. The partners are : Alcatel Espace, Cybernetix, Ikerlan, KFKI-RMKI, Matra Marconi Space, RISP, Sagem, Vniitransmash.

responsible for interpreting and executing mission orders [2] from the operation station. To execute the mission, this structure drives a number of functional sub-systems via flexible command scripts.

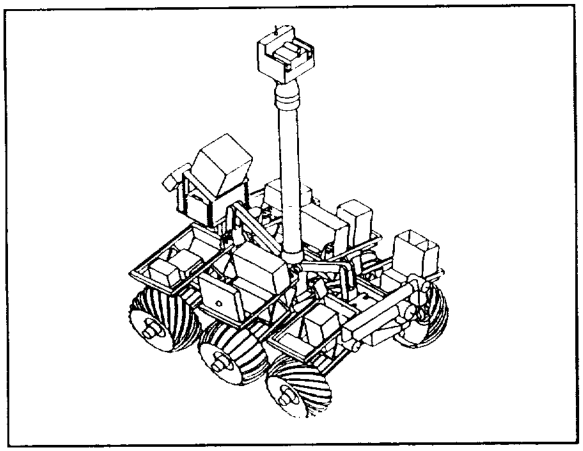


Figure 1. IARES.

Among these subsystems a major one with respect to autonomy is the Autonomous Navigation Subsystem (ANS), whose main role is to compute safe trajectories for the vehicle on unknown and uneven terrains. The terrain is in fact partially known. Indeed, as envisaged for Mars exploration, the ground station sends, for each mission, a map of the terrain consisting of possible itineraries (large navigation corridors computed from a low resolution (10m) orbiter map) .

AUTONOMOUS NAVIGATION

The displacement modes

In order to maximize the average speed of the rover, the ANS provides different modes of displacement adapted to the nature of the terrain which may be classified as flat, uneven but crossable, uncrossable, and unknown. Indeed, the computational resources are limited and it is well known that for instance trajectory planning on uneven terrain requires more complex calculations than 2D planning. The modes of displacement are the following :

- full planning mode adapted to crossable terrain,
- simple planning mode adapted to flat terrain with few obstacles,
- reactive mode for very easy terrains, i.e. flat terrain with rare obstacles.

The full planning mode requires the use of complex 2D and 3D planners. The simple planning mode uses directly a range image to determine very quickly the best direction. The reactive mode simply sets the direction of motion towards the goal.

The choice of the displacement mode is an EDS decision. To each mode corresponds a specific perception, decision, action cycle monitored by the EDS. For instance in the full planning mode this cycle is the following :

1. acquisition of the environment (10-20 meters range), robot at rest,
2. modelling of the terrain and trajectory planning, robot at rest,
3. execution of the trajectory with possible obstacle avoidance.

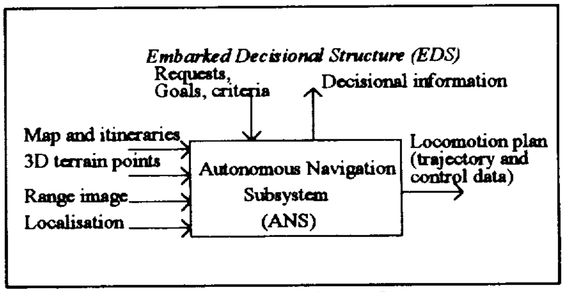


Figure 2. ANS interfaces.

Functions

ANS groups 3 functions that will be detailed below. Figure 2 shows the main interfaces of the subsystem. These functions are terrain modelling, navigation and execution. In the preceding cycle example ANS is concerned with steps 2 and 3.

As already stated the ANS may be viewed as a server which provides the EDS with a set of several services as listed below and which represent the different functions of the ANS :

- terrain modelling : to build and update a navigation map from 3D perception data,
- navigation : to evaluate topological paths towards a goal,and to plan trajectories,

- trajectory monitoring : to check in real time the feasability of the planned trajectory and correct it for avoiding obstacles.

Terrain modelling

During each one day mission the rover will cover about 1 km. The size of the area implies a hierarchical and multi-criteria description of the terrain model.

There are three different models of the terrain corresponding to three different scales (see figure 3) :

- the loaded map with itineraries,
- the regional map [2],
- the topographic map.

The so-called navigation map is the union of the topographic map and the region map.

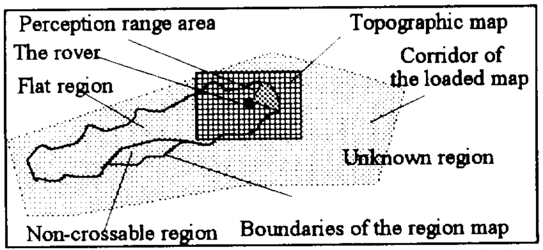


Figure 3. Simplified representation of the different terrain models.

The topographic map is built from range images (stereo [4, 5] and 3D laser imager data). It is a local bitmap around the rover which includes, in a global reference frame, elevation information, ground nature (sand, etc), and type of terrain (flat, crossable, uncrossable, unknown). Building this map consists of classifying the terrain, fusing the data and, as a corollary, refining 3D localisation through map registering. This map is used for the trajectory planning.

The regional map is built from the successive topographic maps and the loaded map. It contains a description of the memorised terrain during the trip in regions of homogeneous type. The regions sketched in figure 3 have been intentionally enlarged. A topological graph is associated with the map, to allow the computation of the feasible paths towards the goal.

Full planning mode

Path evaluation

In the full planning mode, trajectory planning is preceded by path planning and evaluation. The path, and its associated subgoals, is determined, using the region map, through the minimization of some criteria (energy, difficulty, time, etc). The goal (a point or a zone) and these criteria are selected by the EDS depending on the mission status.

Trajectory planning

Once the path is determined, the trajectory planner generates a succession of circular or straight segments to be followed by the rover. Depending on the type of the region, the 2D or the 3D planner is selected. Indeed, on flat terrain, quick 2D trajectory planning algorithms can be used, and on very rough terrain trajectory planning requires complex and computationally expensive 3D geometrical algorithms [6]. These two planners take into account the non-holonomic constraints of the rover. Finally locomotion control data is precomputed (margins, ground slope), a recommended mode of locomotion is selected (wheeled or step-wheeled) and the requirements for the next perception task are determined. The trajectory and its associated control data form what we call a locomotion plan.

Trajectory monitoring

The locomotion plan is sent to the locomotion sub-system. Nevertheless the executed trajectory is checked for obstacles. If an obstacle is detected the rover may be stopped or authorised to avoid the obstacle. On flat terrain a simple avoidance trajectory is computed, using the obstacle shape and the robot model.

Simple planning for flat terrains

The principle is to find the angular sector which is wide enough for the rover, free of obstacles and close to the direction towards the goal. This principle avoids terrain modelling and enables a fast and direct treatment of the range image as each column corresponds to one direction.

A NAVIGATION EXAMPLE

We have described above the functions offered by the ANS sub-system. We will show now how these functions combine to form the navigation process. We will base our explanation on a simple example using the full planning mode. The terrain is illustrated on figure 4. The displacement task is to go from A to B.

Reaching point B requires three displacement cycles (perception, trajectory generation, and execution), corresponding to the sub-goals P1, P2 and B. P1 and P2 are computed by ANS. Each trajectory generation requires the following steps:

- terrain modelling,
- navigation i.e. path evaluation,
- trajectory planning.

Sub-goals P1 and P2 are computed during the navigation step which uses the region map. Along the path computed by the navigation function the sub-goals are chosen on regions boundaries within a given margin. Here P1 corresponds to the boundary between flat and crossable terrain. For the first and third cycle the 2D planner is used because the crossed region is flat. For the second cycle the 3D planner is used because the region is only crossable.

In this example the navigation strategy is simple and does not require a regional map. Indeed the rover is short-sighted and it finds crossable terrains towards the goal. It only has to decide between crossing the rough terrain, or exploring terrain on the left to find flat terrains. More complex situations will require the exploration of the surroundings to find a way through large non crossable zones or even getting out of dead-end configurations. In these cases the regional map becomes necessary. It will allow backtracking to branches of paths not already explored and decide if there is no possible path towards the goal in the current itinerary corridor.

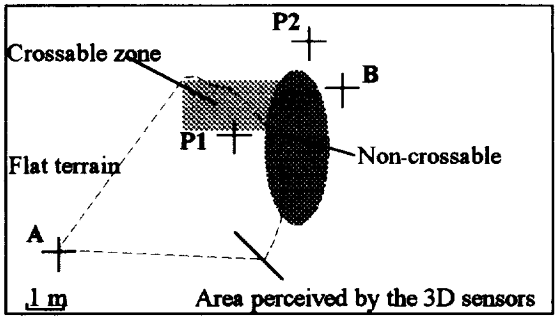


Figure 4. Example of a navigation scenario.

CONCLUSION

The IARES project aims at demonstrating the feasibility of planet exploration by a mobile robot. It requires a robust Autonomous Navigation Sub-System. We are currently developing a complete set of original methods for modelling the environment, planning trajectories and providing meaningful informations to the mission controller.

REFERENCES

[1] Weisbin, C.R.; Montemerlo, M.; and Whittaker,W., 1993. Evolving Directions in NASA's Planetary Rover Requirements and Technology. *Robotics and Autonomous Systems* 11 (1993) 3-11

[2] Lacroix, S.; Chatila, R.; Fleury, F.;Herrb, M.; and Siméon, T., 1994. Autonomous Navigation in Outdoor Environment : adaptative approach and experiment. In *IEEE International Conference on Robotics and Automation.* San Diego. May 94.

[3] Kemerdjian, A.; Gromov, V.; Mishkinyuk, V.; Kucherenko, V.; and Sologub, P., 1992. Small Marsokhod Configuration. In *IEEE International Conference on Robotics and Automation.* Nice, FRANCE May 1992.

[4] Hotz, B.; Zhang, Z.; and Fua, P., 1993. Incremental construction of local DEM for an Autonomous Planetary Rover. In *Workshop on Computer Vision for Space Applications.* Antibes FRANCE September 93.

[5] Faugeras, O.; and al., 1993. *Real time correlation based stereo: algorithm implementations and applications.* Research report, 2013, INRIA Sophia Antipolis, France.To be published.

[6] Siméon, T.; and Dacre Wright, B., 1993. A practical motion planner for all terrain mobile robot. In *IEEE International Conference on Robots and Systems.* Yokohama JAPAN July 93.