N90-16699

# 1989 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

## JOHN F. KENNEDY SPACE CENTER
## UNIVERSITY OF CENTRAL FLORIDA

# OFF-LINE ROBOT PROGRAMMING AND GRAPHICAL VERIFICATION OF PATH PLANNING

PREPARED BY:                         Dr. Gregory L. Tonkay

ACADEMIC RANK:                       Assistant Professor

UNIVERSITY AND DEPARTMENT:           Lehigh University
                                     Department of Industrial Engineering

NASA/KSC

        DIVISION:                    Mechanical Engineering

        BRANCH:                      Special Projects (RADL)

NASA COLLEAGUE:                      Mr. V. Leon Davis

DATE:                                August 8, 1989

CONTRACT NUMBER:                     University of Central Florida
                                     NASA-NGT-60002 Supplement: 2

# ACKNOWLEDGEMENTS

# ABSTRACT

The objective of this project was to develop or specify an integrated environment for off-line programming, graphical path verification, and debugging for robotic systems. Two alternatives were compared. The first was the integration of the ASEA Off-line Programming package with ROBSIM, a robotic simulation program. The second alternative was the purchase of the commercial product IGRIP. The needs of the RADL (Robotics Applications Development Laboratory) were explored and the alternatives were evaluated based on these needs. As a result, IGRIP was proposed as the best solution to the problem.

# SUMMARY

The RADL at KSC is experiencing competition for on-line time with the robots. This is because all of the programming, development, and debugging ties up the robots. To alleviate this problem, it was proposed that an off-line programming and debugging environment be developed. This project explored two alternatives:

    1) the integration of two existing software packages, ASEA Off-line Programming and ROBSIM.

    2) the purchase of commercially available software.

The commercially available software chosen was Deneb Robotic's IGRIP. This package was evaluated because it could run on the InterGraph workstations currently at KSC.

This report exams the types of projects the RADL is involved with and determines several features which would be desirable. Next, each of the alternatives was evaluated based on these and other criteria.

The ASEA Off-line Programming package was found to be easy to use except for the wrist orientation coordinates. The user interface on the ROBSIM package was difficult to use. The potential user had to understand joint transformation matrices, Euler angles, and dynamic parameters. In addition, the current version at KSC had several bugs.

The IGRIP package was found to be extremely easy to use and performed most of the functions required by the RADL personnel. The one capability it did not possess was dynamic simulation. However, this could be supplied by interfacing one of several commercial packages. The IGRIP package was superior in all respects to the other alternative except for price. Even in this category, it was unclear how much it would cost to integrate ASEA and ROBSIM, thus making a cost comparison difficult.

The final recommendation in this project was to purchase IGRIP for the InterGraph workstations that currently exist at KSC.

# TABLE OF CONTENTS

Section                     Title

# LIST OF ABBREVIATIONS AND ACRONYMS

ARLA - Asea Robotics LAnguage

CAD - Computer Aided Design

GSL - Graphical Simulation Language

IGRIP - Interactive Graphics Robot Instruction Program

KSC - Kennedy Space Center

LaRC - Langley Research Center

MSFC - Marshall Space Flight Center

OPF - Orbiter Processing Facility

PCR - Payload Changeout Room

RADL - Robotics Applications Development Laboratory

ROBSIM - ROBotic SIMulation program

TCP - Tool Center Point

# I. INTRODUCTION

## 1.1 ROBOTICS AT KENNEDY SPACE CENTER

With the recent proposal by President Bush to establish a permanent lunar base and initiate a manned mission to Mars, there will be an increase in activity at KSC. Launches will occur more frequently and more payloads will be processed. In order to meet this goal, NASA will need to apply robots to tasks in space as well as ground preparation and servicing of spacecraft. Robots have replaced men performing dangerous or tedious tasks in the industrial and service sectors. It is only natural that space related tasks should be the next frontier for robotics. Several tasks at KSC are candidates for robot applications; for example, working with hazardous fuels and cryogenics, inspecting spacecraft and payloads, and performing last-second tasks at the launch pads.

## 1.2 MISSION OF THE ROBOTICS APPLICATIONS DEVELOPMENT LABORATORY

The Robotics Applications Development Laboratory (RADL) was established to explore the feasibility of applying robotic principles to the shuttle/payload ground processing activities at KSC. The robotic prototype system in the laboratory provides a testbed for projects dealing with many aspects of ground preparation. Furthermore, the laboratory provides a training environment in robotics for engineers. With the expected increase in activity, the laboratory will experience increasing competition for resources, especially programming time on the robots.

## 1.3 OBJECTIVE OF THIS RESEARCH PROJECT

The objective of this research project was to advise RADL personnel of the best way to proceed in order to alleviate the problem of limited availability of programming time and application time on the robots in the RADL. Furthermore, an analysis of current and future projects has shown that several types of tasks consistently reoccur. Tools that could be applied to these tasks have been evaluated and are discussed in greater detail in this paper. A list of these tasks includes the capability to:

> program off-line which reduces the time actually spent using the robot

310

graphically view the robot moving through its environment to detect many programming errors even before the robot is operated

detect collisions between objects in the environment

place various robots (or variations of a proposed design) in a graphical model of the environment to determine optimal configurations and limits

design and locate fixtures in the environment to minimize access problems

detect singularities in a program before it is actually run on the robot

view multiple devices moving within the environment and verify the communication signals between the devices

## II. RADL FACILITIES

### 2.1 ROBOTS

In its current configuration, the RADL consists of two robots: an ASEA IRB 90/2 and a PUMA 560. Most of the development work to date has been performed on the ASEA. This robot has large reach and payload capabilities and is mounted on a 30 ft track to further increase the already large work envelope. It is an ideal candidate to work on the large pieces of equipment that exist at KSC. The ASEA robot is also equipped with an adaptive control option that allows it to dynamically alter its path planning based on outside signals.

### 2.2 PERIPHERAL EQUIPMENT

The robots in the RADL are interfaced to several other pieces of equipment which provide additional support [1]. A MicroVax II is the central computer in the laboratory. It communicates with the ASEA robot through a computer link that has the capability to upload/download programs and perform control functions. The MicroVax II is also interfaced to a DataCube vision system that performs complex vision calculations, a MasterPiece 280 PPC programmable logic controller that can control process outputs and

monitor inputs, and a MasterView graphics presentation system.


## III. APPLICATION AREAS

There are several projects at KSC, currently underway or proposed, that could be significantly enhanced by the findings of this research project. This section will briefly describe some of these projects and relate how off-line programming and graphical verification of path planning could enhance the projects.

### 3.1 ORBITER TILE INSPECTION

Each time an orbiter returns to earth, the protective heat tiles must be inspected for damage and misalignment. Of particular importance are the leading and trailing edges of the wings, the nose, and around the landing gear. Each of the tiles are individually inspected; a time consuming and tedious task that is ideally suited for a robot. Past projects in the RADL have shown that a robot can effectively inspect a mockup of a section of the orbiter. However, before a robot is used near a real orbiter, a graphical verification of the program would provide a substantially increased level of confidence.

If a decision were made to incorporate a robot in the inspection process, NASA would require specifications about the type of robot that should be purchased or designed. A state-of-the-art design environment could show the robot moving through its range of motion next to the orbiter. The number and location of positions required to inspect the orbiter could be determined without even turning the robot on, let alone moving it near the orbiter. If a robot was being designed to perform the task, the designer could experiment with various link lengths, joint limits, and joint configurations to determine the optimal configuration. Commercially available robots could be quickly and easily compared to determine the optimal robot for the inspection task.

### 3.2 INSPECTION AND PROCESSING OF ORBITER PAYLOADS

This task would employ a robot to inspect the payload of the shuttle prior to lift off. It would also involve tasks to bring experiments on-line just prior to lift off. Examples would include turning switches on, removing lens caps,

verifying that pieces are in place, and inspecting for sharp edges that could catch and tear the space suits of the astronauts.

This robot would most likely be located in the PCR (Payload Changeout Room) at the launch pad. A graphical design environment could be used to model a robot in the PCR to determine the optimal configuration. Also, a model of the locations of the payloads in the orbiter cargo bay would allow off-line program generation of the path to perform the inspection tasks. Collision detection capabilities could verify that no collisions would occur.

## 3.3 ORBITER RADIATOR INSPECTION

Prior to each flight, the radiators on the orbiter must be inspected. These radiators are located on the inside of the cargo bay doors. The inspection would take place while the orbiter, with the cargo bay doors open, was in the OPF (Orbiter Processing Facility). Most likely the robot would be suspended vertically from an overhead track. This would cause minimal interference with existing hardware in the OPF.

This project would benefit from a graphical design environment by using a model of the OPF to determine the envelope requirements for the robot to operate efficiently. Collision detection and program generation would also be important in the later stages.

## IV. CONSIDERATION OF ALTERNATIVE METHODS

The current method of robot programming in the RADL utilizes a teach pendant. While this is an adequate method for repetitive tasks, such as in a manufacturing environment, it is not sufficient for highly intelligent tasks where complex decisions must be made in a constantly changing environment.

In the past, robot specification and design has been performed in a trial and error manner. While this method can provide an adequate solution, it seldom approaches the optimal; primarily because the designer does not have time to try many different alternatives. A graphical design environment can provide the designer with tools to quickly make changes in the design and view the results.

313

## 4.1  ASEA OFF-LINE PROGRAMMING PACKAGE WITH ROBSIM

The first alternative explored was to integrate several pieces of software currently in the RADL. This was proposed to minimize the total cost of the project. The first piece of software was the ASEA Off-line Programming Package which uses the language ARLA. This software runs on the MicroVax II and communicates with the robot using the ASEA Computer Link hardware. It provides the capability to program without the teach pendant. Generally, the same functions are provided in ARLA as with the teach pendant [2].

Locations in the program can be entered using the coordinate system of the robot, registers, or a special record mode using the teach pendant. The biggest problem encountered while trying to program entirely off-line was using the ASEA scheme of representation for wrist coordinates. It is very difficult to visualize the map between the real world and the robot coordinate system taking into account the current TCP (Tool Center Point) definition.

Other limitations found in ARLA are the lack of arithmetic and trigonometric functions, the lack of data processing capabilities, and the failure to incorporate the robot track as an additional robot axis. Arithmetic and trigonometric functions are important to calculate positions and orientations of objects in the environment. Data processing capabilities are required to store data in files or access databases. Finally, the robot track is considered to be an external axis by the controller. When a position is entered using the keyboard, the option is not given to enter values for the external axes. Therefore, the calculation of the coordinates of the TCP are not affected by the track position. This makes it difficult to use the track in any mode other programming with the pendant.

Since the ASEA package does not include any kind of graphics, and hence no way to debug a program except to test it on the robot, the ROBSIM package was evaluated as the graphical display tool. ROBSIM was developed by Martin Marietta for LaRC [3,4]. It was designed to be a dynamic simulator, taking into account the physical properties and constants of the links and joints. ROBSIM can provide a graphical simulation of a robot in its environment if the appropriate hardware is available (Evans and Sutherland terminal). Otherwise, it must be run without graphics. There were several problems encountered in trying to model

robots with ROBSIM. The following sections will describe some of these problems in more detail.

4.1.1 GRAPHICS TERMINAL REQUIREMENTS. ROBSIM requires an Evans and Sutherland terminal for proper graphics display. This type of terminal has a series of analog dials that can be used to change the perspective of the display. Without the capability to alter the perspective from the default side view, the user cannot determine where the robot is in three-dimensional space. Although the help files state that a VT240 terminal can be used, it only permits a two-dimensional side view. No capability exists to alter the perspective in software.

4.1.2 INTEGRATION WITH INTERGRAPH. It would be difficult and time consuming to rewrite the ROBSIM I/O routines to interface with the InterGraph family of workstations which are available throughout KSC. The hooks are not readily available, and more importantly are not documented in the current version of ROBSIM.

4.1.3 LACK OF UPDATED DOCUMENTATION. The documentation is different from the current version of ROBSIM that is running on the VAX. The documentation is for the version developed by Martin Marietta. The version of ROBSIM currently running on the VAX was modified by LaRC to reside in their environment.

4.1.4 INVERSE KINEMATIC DIFFERENCES. ROBSIM uses its own internal kinematic solutions to relate joint values to the TCP position. The user must be knowledgeable about joint and link transformation matrices and Euler angles to understand how to use the program. The ROBSIM solutions and displays would only be as good as the model which the user entered for the robot. However, since ROBSIM does not provide for collision detection, the lack of accuracy would not cause a significant problem.

4.1.5 DATA EXCHANGE FORMAT. The two packages in question do not store data in the same format. Conversion programs could be written to interface the two packages, but at the expense of user-friendliness and speed.

4.1.6 PACKAGES LOCATED ON DIFFERENT SYSTEMS. Currently the ASEA Off-Line Programming package is installed on the MicroVax II in the RADL and the ROBSIM package is installed on the Engineering VAX. Since the ASEA program must remain connected to the Computer Link, ROBSIM would ideally be

ported to the same system. Unfortunately, the MicroVax II does do have enough disk space to store the ROBSIM. The two packages could be interfaced using DECNET, by sacrificing some speed and convenience.

## 4.2 IGRIP OFF-LINE ROBOT PROGRAMMING AND SIMULATION SYSTEM

IGRIP is a commercially available software package that combines many of the features required in the RADL. The software was written by Deneb Robotics, Inc. and has been on the market for several years. It is considered by many to be one of the best in its class. Since InterGraph has taken the IGRIP software and ported it to their hardware and since there are many InterGraph workstations already located at KSC, a cost effective solution exists: the purchase and installation of IGRIP on an existing system.

### 4.2.1 FEATURES OF IGRIP.

Although a complete description of IGRIP is beyond the scope of this report, some of the highlights are mentioned in this section so that the various options can be compared. IGRIP integrates a CAD system with a simulation/animation system to provide high quality, shaded surface images of the environment. Multiple robots with unlimited degrees of freedom can move through the environment, manipulate objects, and communicate with other devices. Collision detection and near miss situations can be detected between any group of objects in the environment. The simulation can be recorded and played back at a later time.

The inverse kinematic solutions can be generated by generic algorithms or user written in the language C. Complex devices can be constructed which have joint limit dependencies. The path the robot is to traverse is defined using tag points. Unreachable points on the path can be easily detected. A special mode automatically places a robot so that a group of points can be accessed. This mode would be especially useful in the tile inspection task.

Using GSL, the user can construct descriptions of how a device will operate and communicate with other devices in the environment. Over 40 commercially available robots are predefined in IGRIP. The capability also exists, via supplied translators, to upload/download native robot code for 8 major robot manufacturers (including the ASEA and PUMA robots located in the RADL).

316

4.2.2 DISADVANTAGES OF IGRIP. There are few disadvantages to the choice of this alternative. The first disadvantage is the cost of the software, approximately $60,000. The same software, written to run on a different workstation, could probably be purchased directly from Deneb Robotics at a lower cost. However, a the workstation would also have to be purchased.

The second disadvantage is that currently there is no integrated dynamic modeling package. For certain applications, this may be critical. However, dynamic simulation packages can be used in conjunction with this package to provide dynamic simulations of the environment.

## 4.3 OTHER ALTERNATIVES

There are other software packages on the market which have features similar to IGRIP. However, none have been ported to use InterGraph hardware and CAD files. Since these systems would require the purchase of an additional workstation, these packages were not explored in greater detail.

## V. COMPARISON OF ALTERNATIVES

This section will attempt to compare the two alternatives using criteria which are important to the RADL. A summary of the results of this section are listed in Table 1.

## 5.1 USER-FRIENDLINESS

This is probably the most important criterion in comparing the usefulness of the packages. If a system is difficult to use, no one will take the time to learn it or use it once they have learned it. IGRIP is by far the best choice in this category. It is a mature product that has a proven interface using sophisticated graphics and a mouse/menu system. It is easy to learn and provides many useful analyses. The ASEA/ROBSIM package is at the other end of the spectrum. While the user interface of the ASEA package is acceptable, the ROBSIM package is slow and tedious to use. The documentation does not agree with the code and several bugs exist which frustrate the user.

317

Table 1.  Comparison of Alternatives

| Feature | ASEA/ROBSIM | IGRIP |
|---|---|---|
| Ease of Use | Cumbersome, Difficult | Easy, Mouse-driven Menus |
| Cost | $ For Integration Man-Hours | $ 60,000 |
| Training | None | 5 Free People |
| Compatibility | No Other Centers Known | MSFC (Could Be Others) |
| Flexibility | Redo Integration For Each Robot | 40 Commercial Robots Defined  8 Translators to Native Code |
| Support | None | Yes |
| Collision Detection | None | Exact Intersection of Surfaces |
| Graphics | WireFrame | Wireframe, Shaded Surface |
| Kinematic Solutions | Internal | Internal or User-defined C Program |
| Availability | 2-3 Man-Months | Immediately |
| Hardware | Evans & Sutherland For Full Graphics, VT240 For Limited Graphics | Intergraph Workstation |

## 5.2 COST

The ASEA/ROBSIM package is the least expensive alternative because both packages are already at KSC. However, there would be a cost associated with interfacing the two packages and defining a model of the ASEA robot in the ROBSIM package. For optimal use of graphics, an Evans and Sutherland terminal would be required at an additional cost. Furthermore, the current version of ROBSIM has several bugs which would need to be removed. A rough estimate of time/cost required to define the model and build the system would be 2 to 3 man-months. The IGRIP package, on the other hand, has a higher initial cost ($60,000), but this includes the cost to install the software and train the operators.

## 5.3 TRAINING

IGRIP has superior training because of the availability of vendor-supplied courses. According to the InterGraph representative, the cost of the software includes training for 5 people. To further reduce the training cost, it might be possible to negotiate for this training to take place at KSC rather than the Deneb school. ASEA/ROBSIM training would be totally self-directed. Other than the resident expert who performs the integration of the two packages, no one would be available to answer questions pertaining to the working environment.

## 5.4 COMPATIBILITY WITH OTHER NASA CENTERS

This is a difficult category to award because there are no official packages at other NASA centers. While it is doubtful that anyone will use the combination of ASEA/ROBSIM, some centers may be using ROBSIM to perform dynamic modeling. MSFC is currently using the IGRIP package and highly recommends it. Choosing this option would ensure compatibility between KSC and MSFC.

## 5.5 FLEXIBILITY

Flexibility is defined here as the ease to add new robots and/or alter existing models. In this category, IGRIP is far superior to ASEA/ROBSIM. IGRIP has over 40 commercial robots predefined, including the PUMA located in the RADL. This feature provides the user with a unique capability. Given that the application environment is already defined, the user can insert several different types of robots to determine which one is best-suited for the task. An

estimate of the cycle time can also be determined. Eight of the most common off-line programming language translators are also included to allow the user to generate downloadable programs for the robot. Both the ASEA (ARLA) and PUMA (VAL II) translators are included.

With the ASEA/ROBSIM packages, each new robot would have to be kinematically modelled. Also a separate off-line programming package would have to be purchased and integrated with ROBSIM. This would be a labor intensive operation repeated each time a robot is purchased.

## 5.6 SUPPORT/UPDATES

IGRIP has the best support of the two alternatives. Support is available from InterGraph and Deneb Robotics. Updates are free for some specified time period (1 to 2 years).

On the other hand, the ASEA/ROBSIM combination offers little support. While ASEA will continue to support the ARLA language, ROBSIM is not currently supported and the likelihood of updates being released is low. Each time an update is received, the two packages must be combined again and the interface code rewritten.

## 5.7 COLLISION DETECTION

Since no collision detection is available in ROBSIM, IGRIP is superior in this category. IGRIP provides collision detection using an exact, surface to surface intersection calculation. Checking can be limited to any number of objects. A near miss mode and nearest distance between two objects mode are also available with the tradeoff of a reduction in processing speed.

## 5.8 TYPE OF GRAPHICS

IGRIP is also superior in this category. Images can be depicted using wireframe, shaded surface, or transparent modes. Calculations and screen updates are performed quickly, depending on the number of elements in the environment.

ROBSIM provides only wireframe images. These images are adequate when using the suggested Evans and Sutherland terminal (which is not available at KSC). With a VT240 terminal, only two-dimensional images are available. Furthermore, the point of reference cannot be changed.

Without the Evans and Sutherland terminal, the graphical analysis capabilities are severally limited.

## 5.9 KINEMATIC SOLUTIONS

IGRIP is the best choice in this category. The inverse kinematic solutions are implemented for all of the commercial robots included in the package. Furthermore, the user can write programs to calculate the kinematic solutions for any type of device. Thus, dynamic effects can be incorporated in the calculations.

The user has no control over the kinematic solutions used in the ROBSIM package. The program would have to be altered to add this feature, if it was required.

## 5.10 AVAILABILITY

IGRIP is available immediately. The ASEA/ROBSIM package would require several man-months for a useable version to be completed.

## VI. RESULTS AND DISCUSSION

In comparing the two alternatives discussed in the previous section, it becomes obvious that in every aspect other than initial cost, IGRIP is better suited than the ASEA/ROBSIM combination for the needs of the RADL. The difference in cost is extremely small when compared to the additional capabilities that can be performed by IGRIP users.

IGRIP offers an additional capability not mentioned as a requirement by the RADL personnel: being able to create application scenarios quickly and easily to sell projects to upper levels of management and other funding bodies. It is true that a picture is worth a thousand words. If you can show the potential funding agency a video of a proposed robot, gripper, or fixture in operation, they will have more confidence and will be more likely to supply the funding.

## CONCLUDING REMARKS

In this project two alternatives were compared to find the one which was best-suited for use in the RADL at KSC. It was desired that an integrated environment for off-line programming, debugging, and graphical verification of path planning be developed.

The first alternative, combining the ASEA Off-line Programming package with ROBSIM, had several disadvantages. It was awkward to learn and use, it did not provide collision detection, and it did not provide many of the extra features found in the second alternative. ROBSIM, in its current form, would not run on the Engineering VAX. Extensive modifications would be required to interface it with the ASEA package.

IGRIP, on the other, was found to be user-friendly. It performed all of the required functions except dynamic simulation. This feature could be achieved by purchasing additional software to analyze the dynamics. IGRIP provided better graphics, a modelling environment, and over 40 commercial robots already defined. In addition, translators were available for both the robots in the RADL.

With the additional features provided by IGRIP, it was easier to justify the additional cost. Since, workstations are available, the only additional cost would be that of the software. Therefore, in conclusion, it is recommended that the RADL purchase IGRIP for use as an integrated environment.

# REFERENCES

[1]  Davis, Virgil Leon, "Systems Integration for the Kennedy Space Center Robotics Applications Development Laboratory," MS87-482 SME Technical Report, 1987.

[2]  Off-line Programming: DEC VAX/VMS User's Guide, CK 09-1422E, ASEA Robotics, 1988.

[3]  Evaluation of Automated Decisionmaking Methodologies and Development of an Integrated Robotic System Simulation: Appendix A - ROBSIM User's Guide, NASA Contractor Report - 178051, March 1986.

[4]  Evaluation of Automated Decisionmaking Methodologies and Development of an Integrated Robotic System Simulation: Appendix A - ROBSIM Programmers Guide, NASA Contractor Report - 178052, March 1986.