SAND96-2419C
CONF-9610197--1

Graphical Programming of Telerobotic Tasks

Daniel E. Small
Michael J. McDonald

Sandia National Laboratories
Intelligent Systems and Robotics Center
Albuquerque, NM 87185-1004

RECEIVED
NOV 06 1996
OSTI

Introduction

With a goal of producing faster, safer, and cheaper technologies for nuclear waste cleanup, Sandia is actively developing and extending intelligent systems technologies through the US Department of Energy Office of Technology Development (DOE OTD) Robotics Technology Development Program (RTDP). Graphical programming is a key technology for robotic waste cleanup that Sandia is developing for this goal.

Graphical programming uses simulation such as Telegrip "on-line" to program and control robots. Characterized by its model-based control architecture, integrated simulation, "point-and-click" graphical user interfaces, task and path planning software, and network communications, Sandia's Graphical Programming systems allow operators to focus on high-level robotic tasks rather than the low-level details. Use of scripted tasks, rather than customized programs, minimizes the necessity of recompiling supervisory control systems and enhances flexibility. Rapid world-modelling technologies allow Graphical Programming to be used in dynamic and unpredictable environments including digging and pipe-cutting.

The graphical grogramming paradigm, as developed by Sandia for application to robot system control, broke new ground in 1990 by integrating sophisticated 3-D graphics modeling technology into the real-time control of robot systems.[1] Enhanced operations-based Graphical Programming systems, developed both for targeted prototype application systems and to extend and refine the paradigm, were developed at Sandia through 1993.[2] In 1994, Graphical Programming was enhanced with task interfaces, wide area network (WAN) communications capability, teleconferencing, and new rapid world model building technologies, to produce the first ever Virtual Collaborative Environment or VCE.[3] New extensions are being tested to evaluate their effectiveness and further refine the graphical programming concept.

This paper describes Sancho, Sandia's most advanced graphical programming supervisory software. Sancho, now operational on several robot systems, incorporates all of Sandia's recent advances in supervisory control. Graphical Programming uses 3-D graphics models as intuitive operator interfaces to program and control complex robotic systems. The goal of the paper is to help the reader understand how Sandia implements graphical programming systems and which key features in Sancho have proven to be the most effective.

MASTER

The Sancho Graphical Programming System

The Sancho Graphical Programming system uses TELEGRIP as a substantial part of the operator interface for programming, controlling, and monitoring robot operations. Sancho commands, controls, and monitors systems at the

---

1. Christensen, B. K. and L. M. Desjarlais, *A Graphical Interface for Robotic Remediation of Underground Storage Tanks*, **First IEEE Conference on Visualization – Visualization '90**, San Francisco, CA, October 23-26, 1990.

2. McDonald & Palmquist, *Graphical Programming: On-Line Robot Simulation for Telerobotic Control,* **RIA Robot and Vision Conference '93**.

3. Harrigan, Davies, & McDonald, *Remote Use of Distributed Robotics Resources to Enhance Technology Development Insertion,* **ISRAM '94**

HH

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

high-level while the subsystems (robots, machine tools, sensors) use local controllers to implement individual real-time capabilities, such as robotic motion, laser mapping, and sensor-based control operations. Subsystem planning, simulation, monitoring and execution are integral functions of Sancho.

Sancho works in four major steps:

(1) **Define/Select**: Using TELEGRIP, the user defines tasks through menus and graphical picks in the interface

(2) **Plan/Simulate**: Sancho plans and tests solutions for safety while the user monitors the planning with TGRIP.

(3) **Approve/Download**: If the operator accepts a task plan, Sancho sends a network-based robot program of the approved solution to the robot controller.

(4) **Monitor/Update**: Sancho monitors the robot and updates TGRIP with data fed back from its controller.

These steps are described in better detail below.

Define/Select

There are several ways to define tasks and operations to the Robot. Early Sandia Graphical Programming systems provided the user with a set of operations that the robot could perform directly. Modern interfaces support higher level problem definition in the form of task interfaces. Sancho supports both paradigms, (described below).

A rich world model, that includes 3D geometry and underlying task-sensitive information, can be used to simplify task definition. Sancho uses direct object selection as a primary method for defining task work locations and to control task execution. A simple point-and-click on a 3D graphic object is all that is required to direct the system to query underlying model data (for example, the name and control settings for working on the part). Sancho also incorporates a Visual Targeting system as a video-based point selection option, which is discussed below.

Visual Targeting allows the operator to define work points from video images. With Stereo Visual Targeting, the user selects two similar points from two different views. These 2D selections are triangulated to derive 3D points. A second targeting method assumes an accurate model is in place, and uses one image pick to generate a vector which is correlated with the world model using a ray-casting technique. The 3D point at which the vector intersects the model can be correlated with the surface data to find the normal orientation, thus creating an approach point for the robot. Both of these methods are supported in Sancho for calibration and work-site selection.

Operations-based Interface Paradigm

In the operations-based interface paradigm, robot tasks are executed by commanding the system to individually execute a sequence of operations that accomplish the task. Operations-based interfaces can exhibit a great deal of functionality for a wide variety of uses.

Sancho supports a user customizable operations-based interface and organizes robot operations by tool. The operator can generate individual robot goal points, worksites and paths, plan and execute robot motions, and command the robot to execute individual robot or simulation operations or commands like "open gripper" directly. Sancho allows the applications engineer to define an unlimited set of operations for each tool that a robot can hold. These definitions include the robot commands, corresponding simulation steps, and other subsystem commands. Robot operations can be modified while Sancho is running.

Task-based Interface Paradigm

Task-based interfaces give the user the opposite perspective from operations-based interfaces. Here, the user is presented with a selection of useful goals (defined tasks) that the system knows how to plan and execute. For example, the user might be presented with a task choice such as "Cut Pipe" which can be defined as:

(1) Calibrate pipe in the world model using Visual Targeting.
(2) Get pipe cutting tool if needed
(3) Position tool above worksite
(4) Dock tool onto pipe
(5) Cut pipe
(6) Extract cut pipe piece
(7) Process cut pipe for delivery to waste repository.

As Sancho plans the task (see below), the system prompts the user to define the task parameters (e.g., to select a pipe) and approve component task plan alternatives (e.g., path plans that move the robot near tool stands and worksites). Qualified operators can adjust safety levels (i.e. size of the collision zone) for any particular task and can modify planned motions to accommodate for conditions that the automation and planning subsystems could not resolve. Because the user is guided through the task definition, tasks are always commanded correctly.

Plan/Simulate

Sancho simulates, tests, and previews robot tasks and operations for known safety concerns using TELEGRIP. If collisions are detected or joint limits are exceeded, Sancho will invoke advanced path planning algorithms to overcome these problems. Operators can also cruise or fly through the simulated world from any point in space to see intended robot motions and develop a better understanding of how the work environment is constructed.

Sancho plans linear tasks from macros defined in ASCII files. Task macros are a series of commands that adhere to a simple language grammar. Operations within this grammar include invoking goal selection interface components, commanding path plans, calibrating and generating models, commanding tool operations and querying TELEGRIP models for embedded data. Because the scripts are interpreted, rather than compiled, the task can be tuned on the fly, eliminating the need to recompile the program.

Sancho's task planning, downloading, and monitoring functions are tightly coupled. During planning, Sancho executes each operation in simulation mode while storing resultant individual robot and other subsystem commands along with simulation commands needed for model to real world synchronization.

Approve/Download

If planning successfully completes, Sancho prompts the user for approval to download. If the operator approves the task, Sancho begins to step through the stored commands and distributes them across the various subsystems, via the network. Motion elements are monitored and Sancho receives asychronous events as each motion is completed. When these events are received, Sancho sends the next command in the queue. Non-motion based simulation steps (such as attaching tools, grabbing objects, etc...) are synchronized with subsystem events to provide realistic system monitoring, as is described below.

Monitor/Update

As safe tasks are downloaded to the robot controller, Sancho slaves the simulation system to the robot's encoders and monitors the robot to verify that the task is performed as simulated, or, as with sensor-controlled tasks, to track the real-world effects of the sensors. Sancho can also command the robot subsystems to interrupt motions that excessively deviate from predicted motions or result in entry into hazardous regions. The result is elimination of predictably unsafe motions and risk management for other operations. With the real-time tracking inherent in Graphical Programming, the effect of emergency stops and other unplanned events are immediately represented in the world model and can be quickly and effectively acted on by the system operator. The real-time tracking also provides a continual quality audit function from development to retirement.

Rapid World Modelling Systems Integration

Successful execution of tasks and operations depends on accurate world models. Sancho's integration of Visual Targeting and Laser Mapping (LAMA) technologies allows the operator to rapidly calibrate and update the world model as dynamic changes are made in the environment. These are two examples of sensor subsystems which can be driven directly from Sancho to propagate changes in the real world back into the model.

Stereo Visual Targeting can be used to rapidly calibrate objects in the workcell by finding the feature points on the object in the video images, and then correllating them with the same locations on the CAD object (using the calibration methods in TELEGRIP). This method can place objects quickly and accurately in the workcell.

LAMA is a laser-based structured lighting sensor which works much in the same way as Visual Targeting; by triangulating vectors to derive 3D points. The points here are points along a laser line which is being swept across a surface. This method generates three dimensional models, 10' x 10' in about 30 seconds. When integrated with Sancho, the operator is able to use the task interface to click on a region in the model, automatically generate the scanning parameters for LAMA, execute LAMA scanning, convert the data into a polygonal model, and update the world model with the new data. The combination of Sancho, Stereo Visual Targeting and LAMA are a powerful ally to the operator, allowing rapid calibration of known parts and objects, and on-the-fly generation of surfaced parts.

Conclusions

These innovations have resulted in robot control programs and approaches that are easier and safer to use than teleoperation, off-line programming, or full automation. Sancho, developed to rapidly apply Graphical Programming on a broad set of robot systems, uses a general set of tools to implement task and operational behavior. The keys to producing this graphical programming system were automated planning and programming, model based control, interactive work point definition, task and operations based interfaces, and the integration or rapid world modelling subsystems. Graphical programming systems developed for environmental cleanup robots, use a unique approach to combining these key technologies in a single integrated system. Utilizing graphical programming to control intelligent robot systems results in faster cleanup, extreme safety, and significant overall cost savings.