SAND97-0173C
CONF-970462--6

Graphical Programming of Telerobotic Tasks*

Daniel E. Small and Michael J. McDonald
Sandia National Laboratories
Intelligent Systems and Robotics Center
Albuquerque, NM 87185-1004

Abstract

With a goal of producing faster, safer, and cheaper technologies for nuclear waste cleanup, Sandia is actively developing and extending intelligent systems technologies. Graphical Programming is a key technology for robotic waste cleanup that Sandia is developing for this goal. 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.

Sancho, developed to rapidly apply Graphical Programming on a diverse set of robot systems, uses a general set of tools to implement task and operational behavior. Sancho can be rapidly reconfigured for new tasks and operations without modifying the supervisory code. Other innovations include task-based interfaces, event-based sequencing, and sophisticated GUI design. 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.

Introduction

Graphical Programming uses 3-D graphics models as intuitive operator interfaces to program and control complex robotic systems and brings engineering simulation to the operator 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-modeling technologies allow Graphical Programming to be used in dynamic and unpredictable environments including digging and pipe-cutting.

The graphical programming 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]

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. 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.

The Sancho Graphical Programming System

The Sancho Graphical Programming system uses an engineering quality simulator (Deneb Inc.'s TELEGRIP) as a key part of the operator interface for programming, controlling, and monitoring robot operations. Sancho commands, controls, and monitors systems at the 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:** The user defines tasks through menus and graphical picks in the simulator interface

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

*This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000. Sandia is a multiprogram laboratory operated by the 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.

(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 the simulator's world model with data fed back from its controller.

Define/Select

Sancho supports a broad spectrum of methods for defining tasks. Users can select simple operations (e.g. open gripper, close gripper) for immediate execution. Tightly constrained tasks (e.g. tool exchanges) can be selected from a menu without further input. Complex tasks (e.g. locating and cutting a pipe) require sophisticated user interface components to support, for example, part calibration and 3D task workstite location.

Sancho uses a rich world model, that includes 3D geometry and underlying task-sensitive information, which is used to simplify task definition. Direct object selection is 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. 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 is then correlated with the surface data to find the normal orientation to create robot approach points.

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 which associates robot operations with individual tools. 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) Move cut pipe 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 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. If collisions are detected or joint limits are exceeded, Sancho will invoke advanced path planning algorithms to overcome these problems. Operators can 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

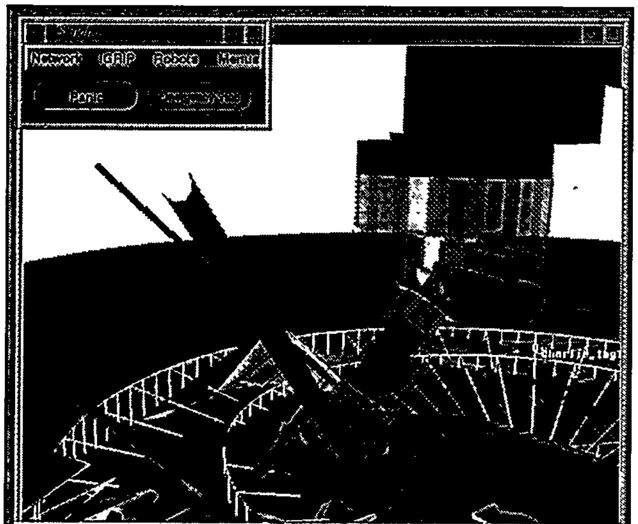

Figure 1: Sancho Graphical Programming System

this grammar include invoking goal selection interface components, commanding path plans, calibrating and generating models, commanding tool operations and querying the simulator's models for embedded data. Because the scripts are interpreted, rather than compiled, the task can be tuned and changed without modifying the core Sancho 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 and grabbing objects, 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. 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, which allows the model to be used quickly and effectively2 for successive tasks. The real-time tracking also provides a continual quality audit function from development to retirement.

Rapid World Modeling Systems Integration[4]

Successful use of graphical programming depends on good world models. In Sancho, world models are kept correct first by monitoring the robot as it works and second by using world modeling tools to measure that which cannot be monitored. visual targeting, Contact Probing, Non-Contact Probing and Laser Mapping (LAMA) technologies are all being used for rapid world modeling. Probing has been described in previous graphical programming papers. [2] The same stereo targeting system used to define tasks can also be used to rapidly calibrate objects in the workcell. Here, the user is prompted to indicate feature points in video images as Sancho correlates these points to the CAD models. This method can place objects quickly and accurately in the model of the workcell.

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 correlating 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 mathematically 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. LAMA 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.

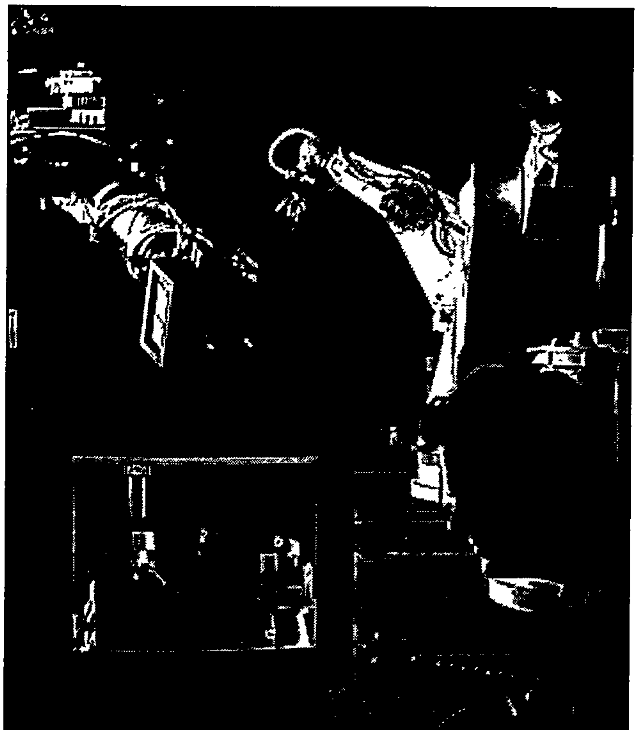

Figure 2: Experimental System

Systems Experiment

Sancho allows system researchers to study, develop, and collaborate on new system design strategies. The remainder of this paper explores the validity and ramifications of one conceptual proposal for task design and its applicability to the use of robots in high-consequence and agile intelligent systems. The experimental system used to test the proposals is described and a series of pilot experiments that demonstrate the validity of the proposal and capabilities that they leverage are discussed. In addition, several other pilot experiments are described elsewhere. [1,5]

Experimental Apparatus

The experimental system is both a production system for sharing research efforts and a testbed for Sancho development. The system includes a dual-bridge PaR XR6100 gantry robot, two Puma 700 robots, a variety of tools, workcell hardware, sensing and computing systems. The robots are network accessible. A networked multi-camera vision system supports remote camera selection, steering, and calibrated image capture. This vision system supports visual targeting, structured lighting and robot monitoring. The high-level computational environment includes Silicon Graphics Inc. (SGI) and Sun workstations (some with video capture), and Deneb Inc.'s TELEGRIP as well as other tools which are used for software development.

Pilot Experiment

For this effort, the robot system was configured to perform robotic cleanup operations in a tank containing simulated hazardous waste. A 15' diameter tank was installed in the Sandia multi-arm gantry lab and filled with 18" to 30" of simulated waste. Debris objects, including pipes, metal plates, and large fixed parts, were buried in the simulated waste.

Graphical programming was used to robotically deploy and operate a Sawzall-based cutting tool. Sancho was used to program the robot to deploy the tool, cut the pipe, and dump the cut part into a barrel. Timing studies were performed to document robotic performance and validate the results.

The all-pneumatic Sawzall tool (shown in Figure 3) uses two cylinder-driven clamps to grasp a pipe on either side of the saw and a cylinder to drive the saw through the cut. In operation, the robot grabs the tool from its stand and automatically makes all pneumatic connections. Once held, the robot moves to an approach point above a pipe and then carefully (using force-over-threshold detection techniques) docks the tool so that the stationary clamp jaw touches the pipe and the back guides are slightly offset from the pipe. Once docked, the clamps are closed to grab the pipe and the saw is actuated to saw the pipe. Because the pipe is grasped on both sides, the operator then has the option of releasing the pipe (if it was originally fixed on both ends) or holding the cut section and releasing the fixed side. In the later case, the cut end can be carried to a receptacle and deposited.

A generalized Sancho feature enabled the operator to indicate the perpendicular approach to the pipe. With it, the operator clicks once on the pipe model where the cut should occur and a second time toward the base of the pipe to indicate the tool orientation perpendicular to the pipe. High-level tasks were developed for the experiment to position the tool above the pipe, dock the tool on the pipe (sensor-based), perform the cutting operation, carefully back the saw and cut pipe piece away from the pipe, and dump the cut pipe into a barrel. A teleoperation step was occasionally used to adjust the position of the tool prior to docking.

Experimental Results

The first several front-to-back trials were performed. The average time to complete the task was 12.5 minutes with a best time of 10 minutes, 13 seconds while a worst-case (first ever) test took 27 minutes (which included much reprogramming during the test).

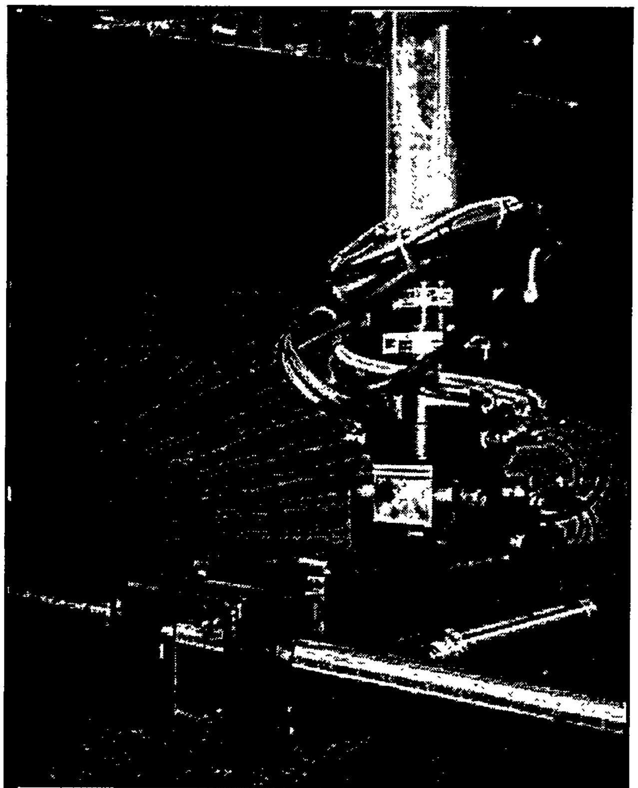

Figure 3: Pipe cutting tool. The Sawzall-based cutting tool has been docked on a pipe section

These tests showed that Sancho is an effective system for performing operations like pipe docking on coarsely modeled systems. Sancho was able to mix teleoperation, telerobotic, and pure robot operations seamlessly within a single set of tasks to complete a typical remediation task. The tests showed that operation times could be minimized by maximizing use of automatic task generation and that good (in this case +/- .5 inch accurate) models can eliminate the need for teleoperation (A small amount of teleoperation is required when the model accuracies fall below an acceptable range).

Conclusions

Graphical Programming is providing the next major leap for robotic safety and efficiency. 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 of rapid world modeling 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, enhanced safety, and significant overall cost savings.

Acknowledgments

A majority of this work was supported by the Department of Energy's Office of Science and Technology through the Robotics Technology Development Program on Technical Task Plan AL2-5-C1-61. The work described in this paper would not be possible without the combined efforts of many people in the ISRC. We especially thank James Tauscher for building the robot laboratory, and Kevin Jones for Sancho operations.

1. Christiansen, 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

4. Little & Wilson, Rapid World Modeling for Robotics, IEEE International Conference on Robotics and Automation '97

5. McDonald, Small, Cannon & Graves, Virtual Collaborative Control to Improve Intelligent Robotic System Efficiency and Quality, IEEE ICRA '97 (pending acceptance)