Apr 23rd, 2:00 PM - 5:00 PM

Paper Session I-B - Autonomous Robotic Systems For SEI Tasks

Davoud Manouchehri  
Rockwell International Corporation, Downey, CA

C. M. Wu  
Rockwell International Corporation, Downey, CA

Margaret M. Clarke  
Rockwell International Corporation, Downey, CA

A. J. Mauceri  
Rockwell International Corporation, Downey, CA

Follow this and additional works at: http://commons.erau.edu/space-congress-proceedings

Scholarly Commons Citation  
ABSTRACT

This paper addresses work performed at Rockwell International's Space Systems Division to further NASA's Space Exploration Initiative (SEI) goals in the development of robotic sensors and displays. As robots perform work farther away from the Earth environment, long time delays between the work station and the robot cause instability in systems that need high refresh rates and may also pose safety risks. One answer to these problems is to design the robotic system so that it is autonomous and closes the control loop around a variety of sensors located on the robot itself. The paper describes work done at Rockwell to implement these control strategies. The first section discusses problems relating to the control of high-frequency systems when there is a significant time delay in the control loop. Next, the integration of tactile and force torque sensors on the wrist and fingers of a teleoperated robotic device is described. Third, the development of a control architecture to implement closing the control loop around the sensors is discussed. The architecture is based on the NASREM control architecture and allows hierarchical integration of sensor data into the control loop. Lastly, future technology needs in this area are described.

1. PROBLEMS ASSOCIATED WITH TIME DELAY IN THE CONTROL LOOP

A typical SEI scenario might involve a robotic device at a distant site, for example, on the moon or Mars, performing tasks like sample acquisition, for which a concept (modified from Pivirotto, 1990) is shown in Figure 1. The time delay between sending a signal from Earth to the robot and receiving signal confirmation back through the appropriate communication satellites could vary from a few seconds for lunar tasks to many minutes for Martian tasks. Time-delayed control of robotic systems falls into two categories: effects on human operators and effects on equipment. Both are discussed below.

1.1 EFFECTS ON HUMAN PERFORMANCE

Human performance often degrades when a noticeable time delay is introduced into the control loop unless specific strategies are adopted to compensate for this. For example, a typical human reaction is to become impatient and to repeat the command. This in turn may overload the remote system or cause other errors in the performance of the remote task. Various techniques have been adopted to help the operator compensate for the time delay. For example, predictive displays may show the operator an immediate simulation of the robot performing the command. Training may help. Another strategy, suggested by Sheridan (1987), is to use supervisory control, discussed in more detail below.

1.2 EFFECTS ON ROBOTIC SYSTEMS

Performance of robot control deteriorates under conditions of significant time delay when control algorithms have not been designed to accommodate such delays. The error function, which is the difference between the command and the response function, increases. This remains true as long as the time delay remains in the system. When the system's lag reaches the point that the actuators fail to output proper responses, the system becomes unstable and does not respond to commands. This raises concerns regarding the effect-
tiveness and safety of conventional control systems in such environments. The problem is compounded when the operator is added to the system and there is a time delay in the camera images and sensor data received at the operator work station.

1.3 SUPERVISORY/AUTONOMOUS CONTROL

Three predominant strategies have been developed in various robotic research centers and laboratories to solve the problems associated with time delay in the control loop. The first solution involves breaking down the input commands into discrete signals. The controller then executes one discrete command at a time, waiting for its completion before moving on to the next input signal. The drawback is that the robot slows down. This slowness can become a critical problem when the task is complex and a large number of discrete commands must be decomposed. The second solution involves the use of predictive displays. Detailed graphic kinematic models of the robot and of the task world, containing all equipment and obstructions in the work space, are developed. The two models simulate the task and verify its kinematics. Commands to the robot are input to these models to simulate and verify the task before they are downloaded to the robot. The disadvantages of this technique include difficulty in developing the world model and in using incoming sensor data to update the model in real time so that it remains faithful to the real-world situation. In addition, it is difficult to simulate the dynamics of the task and of the work site. The third solution is to employ more advanced control algorithm techniques with enough intelligence to accomplish tasks without the intervention of a human operator. This requires the presence of a sensory system that not only senses pertinent changes in sensory information but also processes them and provides the control system with the kind of intelligence necessary to react to differences in these sense data, including contingency situations. The disadvantage of this approach is its complexity and its high sensor system requirements. On the other hand, it does provide the capability to control remote robots that perform complex tasks with considerable time delay in the control loop.

1.4 TYPICAL SCENARIO FOR A TIME-DELAYED ROBOTIC OPERATION

A typical scenario might entail the acquisition of samples on the Martian surface, as described by Pivirotto and Dias, 1990, and NASA, 1989. Considering the concept shown in Figure 1, the vehicle would first plan and execute the motions needed to place the manipulator arm within the sampling site's preselected work space and, second, would extract and containerize the sample. The manipulator arms might be 6-degree-of-freedom, with appropriate position, velocity, force,
vibration, and thermal sensors. Since time delay is involved, the vehicle would need an on-board intelligent system able to plan and execute commands and an efficient sensor-based safety system to retract the arm when it hits rock too hard to sample. Hierarchical control architectures for such a system would reflect task decomposition from high-level sample acquisition tasks to low-level actuation signals. At each architecture level, a corresponding task level would be planned and executed. This execution would be either an instant reflex action, for example withdrawal of the arm from the hard rock, or, if time allows, the beginning of a lower level task.

2. SENSOR INTEGRATION

The sensor integration work was performed in the Rockwell Space Systems Division's Robotics Laboratory in Downey, California. The goal was to provide a test bed for the development of algorithms for the supervisory/autonomous control of a robotic device by closing the control loop around sensors on board a robot slave. Figure 2 shows a block diagram of the main features of this facility. The slave area contains a Remotec RM-10A teleoperator manipulator, seen in Figure 3, with associated transporter system, sensors, viewing and lighting systems, and task boards. A Lord tactile sensor, LTS-200 series, mounted on each end effector (finger) measures forces, torques, and deflections applied to the touch surface in vector and array format. A JR3 Inc. force/torque sensor mounted on each wrist measures forces and torques in three axial directions. The JR3 houses foil strain gauges whose electrical resistances change as loads are applied to the wrist. The change is then transduced into force and torque data in x, y, and z directions. The JR3 can sense forces up to 100 lb and torques up to 100 in.lb. The electrical interface of the JR3 is through an RS-232 connected to the JR3 intelligent support system (ISS), which allows for command inputs and data transmission. The algorithms developed during this sensor integration work are now being modularized for use in the appropriate NASREM levels, as described in Section 3.4.
3. DEVELOPMENT OF CONTROL ALGORITHMS

3.1 REQUIREMENTS

Few requirements were identified for the controller. Since it would be used in a laboratory environment, it should be modular to allow easy diagnosis, replacement, and upgrade. It should accommodate the integration and evaluation of different control algorithms with minimal change. Changes in algorithms should be transparent to low-level control loops. Lastly, the controller should accommodate higher levels of intelligence and reasoning as technology matures.

Consideration of these requirements caused the NASA/NBS Standard Reference Model for Telerobotic Control System Architecture (NASREM), Albus, 1987,
to be selected as the architecture that would best meet the above requirements. NASREM is described below.

3.2 NASREM

NASREM is a control architecture for autonomous/intelligent mechanical systems, including robotics, that offers standardization and flexibility. It was developed by NIST to try to standardize control architecture. NASA and some commercial industries have had good results with its use. NASA’s Goddard Space Flight Center has baselined it as the control architecture for the flight telerobotic servicer (FTS).

NASREM is based on a six-level hierarchy. As one moves from lower to higher levels, intelligence and autonomy increase, as shown in Figure 4. Using global memory, the operator can interface with the various levels and directly command the lower levels. NIST has further developed this architecture, defining the characteristics and interfaces at each level, thus allowing the porting of control algorithms from one robot to another with little modification.

The Rockwell study focuses on the four lower NASREM levels: joint coordinate transformation, computation of inertial dynamics, stringing of trajectory point commands, and specification of elementary moves. Standard software interfaces and modular design allow further upgrade and modification as more sensors are integrated into the system. Furthermore, each module is being developed independently while adhering to interface standards. This strategy allows for both horizontal and hierarchical communication. The flow of command and status feedback is hierarchical, and the sharing of data between modules at each level is horizontal. All input and output variables from all levels are accessed through global memory. This allows the execution of a variety of control algorithms, including control architecture implementation. It also allows the execution of modern control theories such as adaptive control, dynamic optimization, and model reference systems and the use of artificial intelligence and expert systems (AI/ES) in levels 5 and 6 for task decomposition and planning.

3.3 CONTROL ARCHITECTURE IMPLEMENTATION

Figure 5 shows the overall plan for implementing the NASREM control architecture in levels 1 through 4. To accommodate the new architecture, algorithms previously developed during the sensor integration work described in Section 2 were modularized for use in the appropriate NASREM levels. For example, level 4 tasks, such as module replacement, are being de-
composed into their subtasks using a series of prerecorded task sequences. In level 2, each subtask execution is verified against the manipulator singularity envelope and against surrounding obstacles. Each subtask is used to calculate joint command positions. In level 1, these joint command positions are converted to appropriate voltages and sent to each joint motor. In addition, raw sensor data are filtered and processed and then integrated into the appropriate format for use with control algorithms at level 2. Features extracted from vision data are used at level 3 to recognize a task object and determine its position and orientation.

The implementation of the control architecture began with the laying out of the overall architecture. The development of each module followed, starting at level 1 and moving to higher levels. As each module is developed, the robot becomes more autonomous. When possible, existing algorithms and software are used to avoid duplication and to take advantage of the architecture's flexibility.

An additional focus of this project has been the development of the hardware architecture depicted in Figure 6. A VME-based system was chosen to implement the control architecture because of its flexibility and central processing expansion capability and the availability of third-party hardware. The overall hardware architecture has four subsystems: operator interface, data processing, mass storage, and the robot itself. A Sun computer provides the operator interface with sensor data viewing and is the primary platform for software development.
3.4 ALGORITHM TESTING

The algorithm architecture is continuously tested for flexibility and upgrade capability. Specific control algorithms are tested for their affectivity, response characteristics, controllability, and stability. In addition, a wide variety of tasks is being selected for the study of system characteristics, for example, the ability to execute large imprecise and small precise motions, to handle small and large loads, and to track objects.

3.5 APPLICATION TO SEI

The work performed in this and similar projects is essential for the development of robotic system autonomy. Our aim is to integrate presently available technology into an autonomous framework so that the human operator can be removed from lower control levels. The success of SEI requires this technology. The lack of manpower for construction, assembly, payload handling, tending the oxygen plant, experiment handling, and performing daily maintenance and servicing tasks will require autonomous, highly intelligent machines. Some will be unique to one or two tasks, such as payload handling or tending the oxygen plant. But all will need autonomous control and intelligence. The NASREM architecture will facilitate the transfer and adaptation of intelligence/autonomy from one machine to another and will provide the necessary hooks for the high-level executive which will control all machines involved in a complex operation, such as running the lunar base.
4. FUTURE TECHNOLOGY NEEDS

4.1 SENSORS

A wide variety of sensors with self diagnosis (health monitoring) capability will be needed, including skin (tactile), position (linear and angular), proximity, and stability sensors. Built-in processing capabilities will monitor and process the sensor data.

4.2 VIEWING SYSTEMS

Viewing systems must be able to sense, analyze, and "understand" visual data that has rapid variations and discontinuities in chromaticity, intensity, depth, and motion. The goal is to develop systems that can achieve this to provide robust three-dimensional surface descriptions from which autonomous task decisions can be made (Ruoff, 1988).

4.3 ON-BOARD INTELLIGENCE

Integrated hardware and software systems that provide flexible environments for real-time processing of large amounts of data (for example, vision systems) will be required, and high levels of intelligence and decision-making authority must be introduced.

4.4 MOBILITY

The platform for future robotic devices must have mobility over the variety of terrain in its task environment, which may include slopes, soft ground, or irregular ground. It may have to right itself, avoid collisions, plan paths, and know its position in relationship to other vehicles and to the terrain.

4.5 CONTROLS

Efficient calculation of arm joint parameters that correspond to the desired position and orientation of a robot manipulator in space presents a major challenge. At present, neural networks are being developed to address this problem and feature multilayer neural networks and neural net-based fine motion control. In addition, technology must also provide suitable hardware devices to allow adjustable synapses in the networks.

4.6 SYSTEM INTEGRATION

Highly modularized designs must be developed where each module is self-contained and can monitor its health and diagnose and repair itself. Standard software and hardware interfaces must be developed to accommodate both continuous upgrades and routine servicing and maintenance. For example, quick changeout of modules should be possible with minimum tooling. In addition, standardized components and subsystems will minimize logistic requirements.

5. BIBLIOGRAPHY


