Monitoring and Control Systems for Automated Process Plants

Larry C Schooley & Francois E Cellier
Symposium on Space Mining and Manufacturing
UN/NASA Space Engineering Research Center
for Utilization of Local Planetary Resources
October 25, 1989

Abstract

This paper discusses a philosophy of automation and control for process plants which are located in space environments and may be some distance from the nearest human presence. One or more human operators interact through a communication channel with the plant in an effort to cause useful things to happen. Direct operator control is not feasible due to time delay in the communication channel. Complete autonomy is not feasible due to various limitations on current technology. Major variables of the problem include the nature and degree of automation of the plant, and the distribution of machine (artificial) intelligence between the operator location and the plant site.

The discussion begins with a review of Sheridan's paradigm for supervisory control of telerobots. Here the machine intelligence is distributed between the operator site and the plant site, and there are a number of local sensing and control loops as well as remote monitoring and command loops. For operation of a complex process plant, this control philosophy must be expanded and extended in a number of areas.

After description of some recent research on operation of a robot-assisted biochemical analysis laboratory planned for Space Station Freedom, an extended architecture is proposed, and several topics for continued research are outlined. These include, but are not limited to: distributed supervisory control by intelligent agents, design and function of the telerobots, use of neural networks for sensing and preprocessing smart components, advanced artificial intelligence, and communication requirements.

Introduction

The generic task for this kind of activity is shown in Figure 1. One or more human operators interact through a communication channel with an automated plant in an effort to cause useful things to happen. The plant could be as simple as a laboratory robot performing a few well-defined repetitive tasks, or as complicated as a complete system for mining raw materials on the Moon or other locations, and producing liquid oxygen and associated by-products. The communication channel is characterized by its digital communication capacity (bits/second), and by the round trip time delay. The time delay is composed of round trip travel time through the communication media and various processing delays imposed by components of the overall system. The distance between the operators and the plant may be as small as from the operations center of Space Station Freedom to an attached payload, or as large as from the Earth to Mars, or even Neptune. Other major variables include the degree of automation of the plant, and the distribution of machine (artificial) intelligence between the operator location and the plant site.

In the original paradigm for such systems, all of the intelligence resided with the human operator, and the plant reacted under closed loop or open loop control to the operator's detailed commands. An early example of this approach is the development of master-slave manipulators used to avoid radiation hazards in the nuclear energy industry. As computing and display devices became more advanced, a certain amount of very primitive machine intelligence was also installed at the operator site. An example of this would be the control of simple remote sensing satellites in low Earth orbit, such as the Solar Mesosphere Explorer. Here a computer program would be used to translate human-language commands such as "turn heater on" into the binary machine-language sequence which caused the plant to perform the ordered operation.

The obvious problem with this mode of operation is that it becomes totally impractical with even a small amount of time delay in the communication system. The closed control loops become unstable (even the ones where the human operator is part of the loop), unless the control signal bandwidth is reduced to such an extent that even the most elementary operations take an unacceptably long time to accomplish. An example of this is the control from Earth of the manipulator of the Surveyor III spacecraft in 1967. It took many hours to dig a small trench in the lunar surface and perform a few simple tests. Even when time delay is not a problem, as on the space station, the cost of crew time and the problems of providing complete crew training make the direct control approach quite unattractive.

Another extreme philosophy is to completely automate the plant and to place enough artificial intelligence at the plant site to enable autonomous operation. Now the human operator only starts the process, perhaps with the launch command, and the plant proceeds to carry out the entire mission without any intervention. The plant circumnavigates all new developments and unforeseen contingencies, and the only contact with the operator is to periodically report results. The problem with this approach, of course, is that the current state of the art does not permit autonomous execution of complex missions in hostile and imperfectly known environments with an acceptable probability of success.

The obvious compromise between two extremes is to adopt the philosophy of supervisory control of semi-autonomous plants. Here the machine intelligence is distributed between the operator site and the plant site, as shown in Figure 2 (Sheridan, 1988). The function of the various numbered paths and loops in this figure may be summarized as follows:

1. Plant is observed directly by human operator's own sense (perhaps with some time delay). An example would be a video feedback channel.

2. Plant is observed indirectly through artificial sensors, computers, and displays.

3. Plant is controlled by local automatic mode.

4. Smart sensor's interact directly with the plant.

5. Smart actuators interact directly with the plant.

6. Human operator directly affects plant by manipulation.

7. Human operator affects plant indirectly through a supervisory controls interface, remote and local computers and actuators.

8. Human operator gets feedback from within the remote commanding computer -in editing a program, running an expert system or planning model, etc.

9. Human operator orients himself relative to supervisory control or adjusts control parameters.

10. Human operator orients himself relative to displays or adjusts display parameters.

The term "supervisory control" is derived by analogy to a supervisor directing and monitoring the activities of one or more subordinates. The supervisor communicates general directives that the subordinates will understand and translate into detailed actions. He also receives from those or other subordinates communications from which he may infer past and current states of an ongoing system or process of interest. To the extent that the subordinates are more intelligent or reliable, the human supervisor need give less specific instructions, and fill in context only to the extent that subordinates cannot act from their own memories, senses, and inference capabilities. When the subordinates can absorb complex instructions and execute more extensive tasks on their own, the supervisor can give less attention to monitoring their activities and less frequently issue new instructions or otherwise interfere (Sheridan, 1988).

In this paradigm, the operators give information to and receive information from a computer system at the operator site which communicates with a computer system at the plant site. The human indicates goals, branching conditions, identifications of objects, operations to be performed, criteria of failure, criteria of task termination, etc. to his computer, and after intermediate processing, the plant site computer closes a control loop through artificial sensors and effectors and the task environment. The supervisory control-loop closure is intermittent and periodic. Examples of missions where this mode of operation has been successfully employed include the Viking Mars lander and the recent Voyager expedition to the outer plants. There relatively crude intelligent robot systems included an appreciable capability for intelligent decision-making and action (under supervision from Earth) in response to uncertain environments and unanticipated events. It has been estimated that a Mars roving vehicle which is completely controlled from the Earth-based control stations, could operate at least 80-90% of the time (Heer and Lum, 1988).

The literature is about evenly divided on which site should be referred to as "local" and which as "remote". Human factors engineers, mission control personnel, and many scientists tend to refer to the location of the human operators as the local site and the distant plant as the remote site. Control engineers, the onboard crew (if any), and some scientists tend to refer to the plant

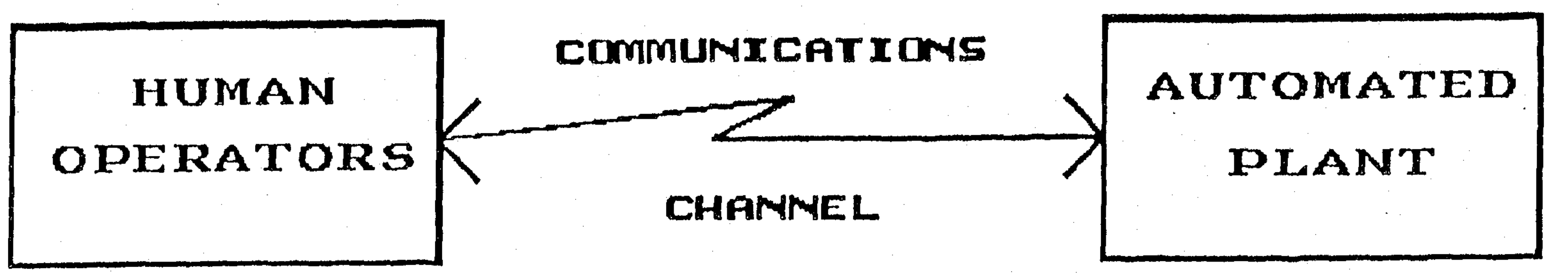
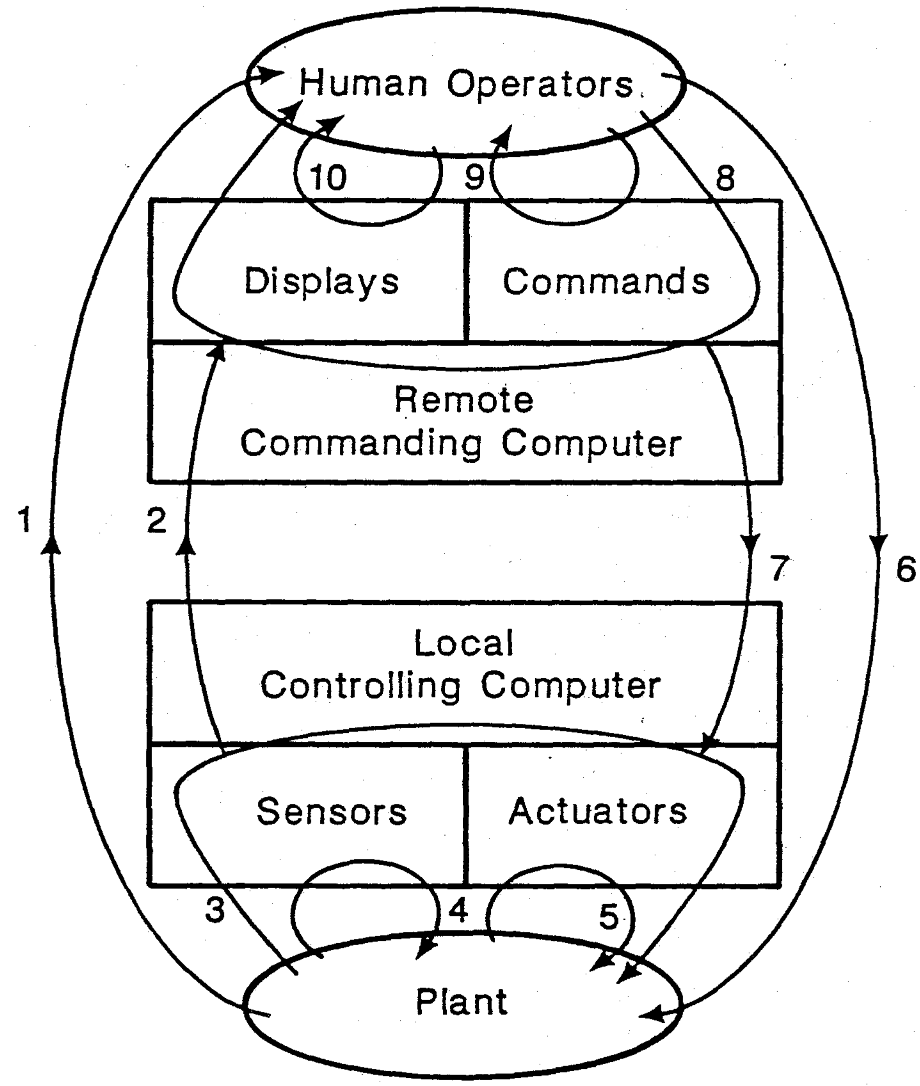


FIGURE 1.1 GENERIC TELEOPERATIONS TASK



SUPERVISORY CONTROL OF AUTOMATED PLANTS

FIGURE 2.1

as the local site (with the local control loops which operate without time delay) and the human operator site as the remote site. The latter is the terminology used in this report. To help avoid ambiguity and possible misunderstanding, the computing systems at the plant will be called the Local Controlling Computer (LCC), and the computer systems at the human operator site will be called the Remote Commanding Computer (RCC).

The following terms have become somewhat standard in the literature (Sheridan, 1988):

Teleoperation
Extension of a person's sensing and manipulating ability to a place remote from him. A teleoperator includes, at the minimum, artificial sensors, arms and hands (or other end effectors), a vehicle for carrying these, and communication channels to and from the operator. Teleoperation refers most commonly to direct and continuous human control of the teleoperator, with no automatic control loops. More recently, however, the term teleoperation has also been used to encompass telerobotic control.

Robotics
The science and art of performing, by means of an automatic apparatus or device, functions ordinarily ascribed to human beings, or operating with what appears to be almost human intelligence. "Robot" ordinarily implies autonomy, with essentially no human interaction.

Telerobotics
The science and art of enabling a human operator to command and observe the actions of a remote device, which itself, to a limited extent, can automatically perform sensing and manipulating tasks programmed into it (e.e., which acts as a robot). Thus a telerobot is a supervisory-controlled teleoperator.

Telepresence
The ideal of sensing sufficient information about a remote task, environment, and teleoperator; and communicating and displaying this information in a sufficiently realistic way that the human operator feels himself to be physically present at the remote site. A more restrictive definition requires further that the teleoperator's manipulative dexterity match that of the bare-handed operator. Telepresence is usually deemed to be desirable for direct manual teleoperation. It may also be desirable for telerobotics.

Note that the term "supervisory control" is more general than the term "telerobotics", as supervisory control can apply to control of nearby tasks as well as tasks that do not involve manipulation of physical objects. Telerobotics does imply supervisory control, but the reverse is not true. A remotely controlled vehicle that does not manipulate its physical environment would not normally be called a telerobot.

The next section contains a brief description of some recent experience in applying these principles to operation of scientific experiments on Space Station Freedom. Subsequent sections will elaborate on various components of the overall system, and identify areas where additional research is needed.

Some Recent Experience in Supervisory Control

This section summarizes two recent examples of the use of supervisory control of telerobots which were developed at the University of Arizona as part of the Telescience Testbed Pilot Program. This work was funded under NASA subcontract 800-62, dated July 1, 1987, from the Research Institute for Advanced Computer Science of the Universities Space Research Association.

NASA's Office of Space Science and Applications initiated this pilot program to validate the user-oriented rapid-prototyping testbed approach to address a range of operations and information system issues. Fifteen universities, under subcontract to NASA, conducted a variety of scientific experiments emulative of the scientific research of the space station era and aimed at resolving critical issues in space station operations concepts and information system design. The goal was to allow scientists to interact with potential space station technologies in a manner that will allow resolution of design and specification questions without having to wait until space station hardware is available.

The first example involves teleoperation of a forerunner of the Astrometric Telescope Facility (ATF) to be attached to Space Station. The second involves development of systems and software for Remote Fluid Handling (RFH) in support of the microgravity and life sciences. These seemingly quite different testbed demonstrations were selected to pursue the following goals:

1. To design a set of tools which allow teleoperation of scientific experiments. These tools include the man/machine interface at the remote commandingcomputer, the machine/machine interface between the remote commandingcomputer and the local controlling computer; and the machine/instrumentinterface between the local controlling computer and the equipment (robots,telescopes, measurement instruments, analytical instruments, etc.) which may comprise any specific experiment.

2. To ensure that these tools are generic and modular, so that they can easily be applied to a wide variety of scientific applications on Space Station (or otherplatforms), and so that the individual modules can be revised without significant impact on the remaining parts of the software.

3. To evaluate the technologies underlying the above developments and derivere commendations (specifications) for future developments.

In the ATF project, a forerunner of the astrometric telescope facility (Thaw telescope, Allegheny Observatory, University of Pittsburgh) was simulated on a MicroVAX workstation. This telescope model was remotely operated by a second MicroVAX workstation. Commands are sent from the first workstation to the telescope workstation. Various real-time activities of the emulated telescope and data are telemetered back to the first workstation from the emulated telescope for display (Schooley and Cellier, 1988).

The RFH demonstration involved teleoperation of a laboratory which provides automated handling and analysis of fluids, such as those which might be extracted from laboratory animals or human subjects as part of the life sciences program, or sent from Earth to be processed as part of the microgravity sciences program. The experiments selected were a determination of the pH of a solution, and the separation of a solution into its charged components using electrophoresis. The laboratory equipment controlled included a SCORBOT robot, special fluid handling apparatus, and the instruments and electronics required for the two experiments (Schooley and Cellier, 1988).

In both cases, the architecture consisted of a remote commanding computer (RCC) which communicates at 9600 baud over dialup phone lines, Sytek network, or Ethernet network with a local controlling computer (LCC). In both cases, the RCC consisted of a MicroVax II/GPX workstation which housed a human/computer interface consisting of an application of the Operations And Science Instrument System (OASIS) previously developed by the University of Colorado (Davis and Hansen, 1986; Jouchoux, et al, 1987, 1989). For the astrometric telescope, the LCC was a second MicroVax workstation which also ran the telescope simulation. For the fluid handling laboratory, the LCC was an IBM PC compatible with 640K memory, an

8087 coprocessor, a 20 Mbyte hard disc, and a LabTender multifunction board. A second PC represented the space resident kernel of the Space Station Information System. For both experiments, intermediate command language statements, telemetry, and scientific data were exchanged using DECnet and CCSDS packets as the communication protocols (Bienz and Hunter, 1988).

The table below summarizes the software tasks which were required for the two demonstrations. This emphasizes the success in developing generic systems/software for two quite different experiments. Only the machine/instrument interface is significantly different.

Phase I Demonstration Schedule

TASKS	APPLICABILITY
Remote Commanding Computer	
OASIS Database	Both
OASIS CCSDS protocol changes	Both
Command storage	N/I
Local Controlling Computer Communications	
DECnet/Ada interface	Both
CCSDS Depacketizer	Both
Telemetry Packet Storage	N/I
CCSDS Packetizer	Both
Ada/DECnet Interface	Both
Telecommand Storage & Retrieval	
Queue Manage	Both
Mailboxes	Both
Retriever	Both
Command Scheduler	N/I
Command Processing	
Scanner	Both
Parser	Both
Interpreter	Both
Local Controlling Computer Interfaces	
Local Keyboard Interface	Both
Local CRT Interface	Both
Telemetry Handlers	
Status Data Handler	Both
Scientific Data Handler	Both
Instrumentation	
Telescope Simulator	ATF
Telescope Simulator Database	ATF
Failure Simulator	ATF
ADA/Basic Interfaces to Labtender & Robot	RFH
Application Programs in Basic	RFH
Syringe Driver Assembly & Control	RFH
Pipette Control	RFH

N/I - not implemented in phase I

This relatively simple architecture must be extended in several key directions. Subsequent sections will present brief discussions of some of the most important extensions.

Distributed Agents

The entire testbed software was coded in Ada. Most of the entries in the above table have been coded as separate tasks that execute in parallel. The depacketizer task accepts incoming packets, strips them off their various packed headers, and passes them on to the queuer task which places them in a mailbox for further processing. The command retriever task takes the commands out of the mailbox, and passes them on to the local controller task, where they are scanned, parsed, and interpreted. The local controller makes a rendezvous with the simulator task, which performs the required action. The telemetry handler task extracts the telemetry information from the stimulator, and passes this information on to the packetizer task for transmission to the remote commander.

This process can be viewed as a chain of factory workers standing at several conveyor belts and performing actions on goods that pass through in front of them. Each worker is responsible for a relatively simple and straightforward task and knows nothing about the rest of the world. Yet, the end product can be quite complex.

This is precisely what the unique features of the Ada language allowed: the overall system complexity was enhanced without enhancing the component complexity. The individual tasks can be executed as separate tasks within the same Ada program running truly in parallel on separate CPUs. The individual tasks are being referred to as agents.

At this point in time, the testbed software is not very robust. A single component faulure will bring the entire testbed to a grinding halt. To stay with the analogy: if one of the workers falls sick, the chain breaks down. In a factory, there is usually a supervisor who can assign tasks to available workers, and redistribute tasks at any time if he so wishes. In one of the current research topics, an attempt will be made to introduce a "supervisor" to the testbed. The supervisor is a so-called watchdog monitor program that operates on a "world model" (Albus, et al., 1987), a qualitative description of the overall system performance. The existing failure simulator (Lew, 1988) is being enhanced from simulating intermittent communication errors only to simulating persistent component failures as well. If the watchdog monitor (yet another agent in the system) recognizes that the system functions differently from the expectations suggested by the world model, it will become active. It can probe the system by interspersing its own commands, and observe the behavior of the plant. If it decides that a particular type of error has occurred it can either send in a repair crew, or it can terminate a task and restart it (replace the worker). Depending on the nature and context of the failure, these actions could be automated also, or human assistance could be requested.

One of the design goals is to ensure that the overall system can work with or without the watchdog monitor. The human operator must be able to start effect on overall system integrity. It must also be possible to run several watchdog monitor tasks in parallel without any negative side effects.

Smart Sensors

The previous section described how overall system complexity can be enhanced while keeping the component complexity (make the components smart), without enhancing the system complexity further. The component complexity is entirely contained within the component and is hidden from the outside.

To give an example: in the previous scenario, when a sensor was broken the watchdog monitor had to observe the (probably indirect) effects of that failure, reason about possible causes of the observed effects, probe the system to verify its hypotheses, and then send in an automated repair crew. In the new scenario, the sensor will be equipped with a limited degree of diagnostic capabilities of its own. The sensor can reason about its own integrity, it can test itself periodically, and it can swap the active sensory component with a backup, and finally, it is able to recognize its own limitations and call for help when needed. Thereby, the watchdog monitor is relieved of some functions, and can therefore be a little less intelligent. To stay with the factory analogy: one can either hire a very intelligent foreman who can cope with truly stupid workers, or one can hire more intelligent workers and then can hire a little less clever foreman. Overall system intelligence can be achieved in various ways, and therefore a problem of "intelligence balancing" must be solved. It is suggested that the overall system performance without failure occurrence may be better with the more centralized approach, but that the system adaptability and invulnerability to failures may be better with the decentralized approach.

Intelligent Agents

Sensors are only one type of components (agents) that can be equipped with independent intelligence, and diagnostic capabilities are only one type of intelligence that is needed in the overall system. Another type of intelligence needed is task planning under uncertainty. In the current testbed, all tasks have been hard coded as sequences of primitive operations. A command issued by the remote commander simply initiates the execution of this hard coded series of primitive operations. This is insufficient for practical purposes. In another research activity, currently funded under NASA--Ames Cooperative Agreement No. NCC 2--525 (Zeigler, et al. 1989), means of task planning are being investigated.

Tasks are decomposed into a series of unit actions. Unit actions are further decomposed into a series of primitive operations. The decomposition of unit actions into primitive operations is static; each unit action always translates into exactly the same execution plan. The decomposition of tasks into unit actions is dynamic (goal driven). Tasks are described as a system entity structure (Zeigler, et al. 1989) containing, in a hierarchical tree structure, all variants that are possible for the given task. For example, if Robby is asked to paint a car green, Robby can either use a brush or a spray. Robby can start from the front or from the rear end, and finally, Robby can drive the car to the next garage and subcontract the task. An entity structure pruner selects one among the many variants, and prunes away all branches of the tree that belong to alternate variants. It does this with the help of a goal resolver (an expert system) that, on the basis of desired task properties, decides which among the variants to choose. An action sequencer takes the pruned entity structure and generates the action plan.

At the low level end, a task optimizer checks the execution plan for possible means of optimization. If for instance the last operation of one unit action was to let Robby drop a tool, while the first operation of the next unit action was to let Robby pick up that same tool, both requests can obviously be deleted from the execution plan. This functionality is similar to code optimizers in language compiler design.

One of the next goals is to integrate the task planner as another intelligent agent into the testbed.

Effort is also underway on smart task executors. This involves intelligent hierarchical controllers and algorithms for cooperation between several robots sharing in the execution of a task (Zeigler, et al., 1980).

The overall system architecture can be viewed as an implementation of large portions of the NASREM telerobot architecture (Albus, et al., 1987), which decomposes the telerobot control system into three columns: and into six rows of increasing/decreasing complexity. The lowest level (row) is that of the primitive operation, while the highest level is that of overall mission control.

Not only will the intelligence be distributed, but there will be a considerable amount of preprocessing of the raw sensor data, particularly in the area of machine vision. It is anticipated that neural networks will be heavily used in this part of the overall system.

One of the key lessons from our telescience experiments was that it is extremely important to design the plant, the telerobots, and the system architecture simultaneously. The reason for this is that the state of the art in telerobotics is still relatively primitive, particularly in the robotics area. It is much easier to design the plant to be compatible with a set of relatively simple and robust robots than to design one complicated multiple purpose robot to deal with a general plant.

Communication Requirements

The communication system must allow the flexibility of control from more than one location, and for the possibility that several locations will observe what is transpiring. For example, it should be possible to control a lunar oxygen production plant from the plant site, from the lunar habitat, from lunar orbit, from low Earth orbit, or from the surface of the Earth, and to monitor the operation from all the other locations.

Time delay is critical, but is largely beyond control of the system designers (round trip travel time over large distances). Certainly the processing delays in the local controlling computer, the remote commanding computer, and any other computers in the communication system itself must be minimized.

Considerable communication bandwidth is also required. The exact requirements await further research, but some conclusions form the Telescience Testbed Pilot Program give an idea of the magnitude of the problem (Leiner, 1989).

"For those functions which involve time critical monitoring and control, the results indicated that the required services included audio, command/control/telemetry, and video. The audio channels each require 64 kilobits per second (kb/s) for nominal pulse code modulation of a 4 kHz analog channel. For speech channels, this can be reduced to 16 kb/s by using a linear predictive coding compression algorithm. It was demonstrated that 32-64 kb/s was sufficient for the command/control/telemetry channels. Data compression is not recommended here for the sake of robustness. Arizona found that their video information for control of lifescience experiments could be transmitted at 50-400 kb/s (depending on required quality and rates of motion) by using commercially available compression techniques developed for video teleconferencing. If video comprises scientific data or critical process control data, much higher rates may be required. RPI used 512x512x8 bits per frame at 30 frames per second (7.8 Mb/s) for their microgravity experiment. Future use of high-definition television (HDTV) would require even higher rates. It should be emphasized that these rates are per subchannel. The experiments in the pilot program each used 1 or 2 audio subchannels, 1 command/control/telemetry subchannel and 1 to 5 video subchannels."

Communication system design must always take into account the special environmental aspects of the particular plant site. This could include, for example, contamination avoidance, charge buildup, dust, high vacuum, intermediate gravity, temperature extremes, long diurnal cycle, radiation, and micrometeorite showers. These factors, along with the high data rates required, imply that the communication technology employed will probably be optical fiber at the plant site, and either optical or extremely high frequency radio transmission between the local controlling computer and the remote commending computer locations.

Summary and Conclusions

It seems clear that an extension of the supervisory control paradigm is the proper approach to monitoring and control of automated process plants in remote space locations. Careful attention must be paid to simultaneous design of the plant, the telerobots, the distribution of intelligence, and the communication/control architecture. Current technology is sufficient to support all of the individual components of such a plant, but considerable additional research is necessary to develop implementation details and to integrate the parts into a robust total system.

References

1. Albus, J.S., H.G. McCain and R. Lumia, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", in NBS Technical Note 1235, Robot Systems Division, Center for Manufacturing Engineering, National Technical Information Service, Gaithersburg, MD, 1987.

2. Bienz, R.A. and J.J. Hunter, "Communication Software Design for Telescience Demonstrations", in Technical Report TSL-019/88, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ, December 1988.

3. Davis, R. and E. Hansen, "OASIS Teleoperations Package Makes its Debut", Information Systems Newsletter, March 1986.

4. Heer, E. and H. Lum, "Toward Intelligent Robot Systems in Aerospace", in Machine Intelligence and Autonomy for Aerospace Systems (E. Heer and H.Lum, Eds.), American Institute of Aeronautics and Astronautics, Inc., 1988.

5. Jouchoux, A., J. Cowley, R. Davis, E. Hansen, M.Klemp, S. Lasater, D. Mullens, T. Sparn, and G. Tate, "Developing a Spacecraft Monitor and Control System in Ada", Proceedings of the Joint Ada conference, Fifth National Conference on Ada Technology and Washington Ada Symposium, 1987, pp. 244-248.

6. Jouchoux, A., E. Hillis, G. Tate, "Porting a Spacecraft Monitor and Control System Written in Ada", Proceedings of the sixth Washington Ada Symposium, 1989, pp. 163-168.

7. Laboratory for Atmospheric and Space Physics, "OASIS Operations and Science Instrument Support, System Managers Guide", University of Colorado, Boulder, CO, 1987.

8. Leiner, B.M. Telescience Testbed Pilot Program, Final Report, Technical Report TR-89.8, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffitt Field CA, February 1989.

9. Lew, A., "Astrometric Telescope Simulator for the Design and Development of Telescope Teleoperation", MS Thesis, Electrical and Computer Engineering Department, University of Arizona, Tucson AZ, 1988.

10. Schooley, L.C. and F.E. Cellier, "Telescience Testbed Pilot Program Final Report, Technical Report TSL-021/88", Electrical and Computer Engineering Department, University of Arizona, Tucson AZ, December 1988.

11. Sheridan, T.B., "Supervisory Control of Telerobots", in Machine Intelligence and Autonomy for Aerospace Systems (E. Heer and H. Lum, Eds.), American Institute of Aeronautics and Astronautics, Inc., 1988.

12. Zeigler, B.P., F.E. Cellier and J.W. Rozenblit, "Design of a Simulation Environment for Laboratory Management by Robot Organizations", in Journal of Intelligent and Robotic Systems, 1, 1989, pp. 299-309.