2. AUTOMATION AND CONTROL

2.1 General Considerations

A number of assumptions must be applied to the consideration of an semi-autonomous plant for producing oxygen on the moon, these include:

- the availability of a high data-rate computing system with mass storage
- the use of expert system-based monitoring and control programs
- remote as well as local control capability
- telerobotic inspection and repair capability
- the use of "smart" components (described in Section 2.4)
- the modularity and interchangeability of most components
- a system design capable of evolving toward fully autonomous operation

Expert systems will be needed because of the high degree of complexity and low level of manned presence on the lunar surface. Additionally, the high data rates required for close monitoring will also require an automatic, or possibly autonomous, system for control. There will also be the probability of remote operation from earth, LEO, or the manned base that dictates the need for an expert system to comprehend the control source, self-diagnose faults, and provide the necessary interlocks to prevent dual operational attempts from two sources.

Coupled with the need for automatic operation is the need for robotic devices that will be used for materials handling and replacement of parts, but not for repair. The state-of-the-art of robotics today is such that the possibility of complicated robotic repair is small. However, the use of robots for replacement of defective components is the obvious requirement for parts that are interchangeable and modular. Modularity allows operation at a reduced level of performance to continue while one of the modules is being replaced. Interchangeability is required to keep the number of spare parts to a minimum.

Part of this philosophy of replacement is the question of redundancy. All workshop participants felt that this was an area that required further study to determine the trade offs between replacement and redundant systems, recognizing that some of these would be "smart" components, having the ability to perform self-health evaluations and maintain a certain degree of data handling for a distributed system.

Also, in arriving at appropriate designs for automated control and communication systems it must be recognized that the lunar environment presents special difficulties because of microgravity, radiation, vacuum conditions and charge build-up caused by dust particles. Dust could be repelled and surface cleanliness maintained by application of the proper electrical potential, but this area was viewed as a research recommendation. The operation of moving parts in a vacuum at a temperature where vacuum welding could take place also raises questions about lunar tribology and suggests a need for further work in this area. Other special issues specific to lunar operations include thermal extremes, contamination of the research environment (especially seismic contamination), long diurnal cycles, and micrometerorite protection.

In the generic task for this kind of activity one or more human operators interact through communication channels with an automated plant. 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 some planetary body, and producing liquid oxygen together with a number of useful byproducts. The communication channels are characterized by their digital communications 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 in the human operator; 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 (Davis, 1988). Here a computer program is used to translate human-language commands, such as "turn heater on", into the binary machine-language sequence which causes the plant to perform the ordered operation.

The principal 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 unattractive.

Another extreme philosophy is to completely automate the plant and 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 with no intervention. The plant handles all new developments and unforseen contingencies; 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 execution of complex missions in hostile and imperfectly known environments with an acceptable probability of success.

A reasonable compromise between these 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 2.1 (Sheridan, 1988). The function of the various numbered paths and loops in this figure may be summarized as follows:

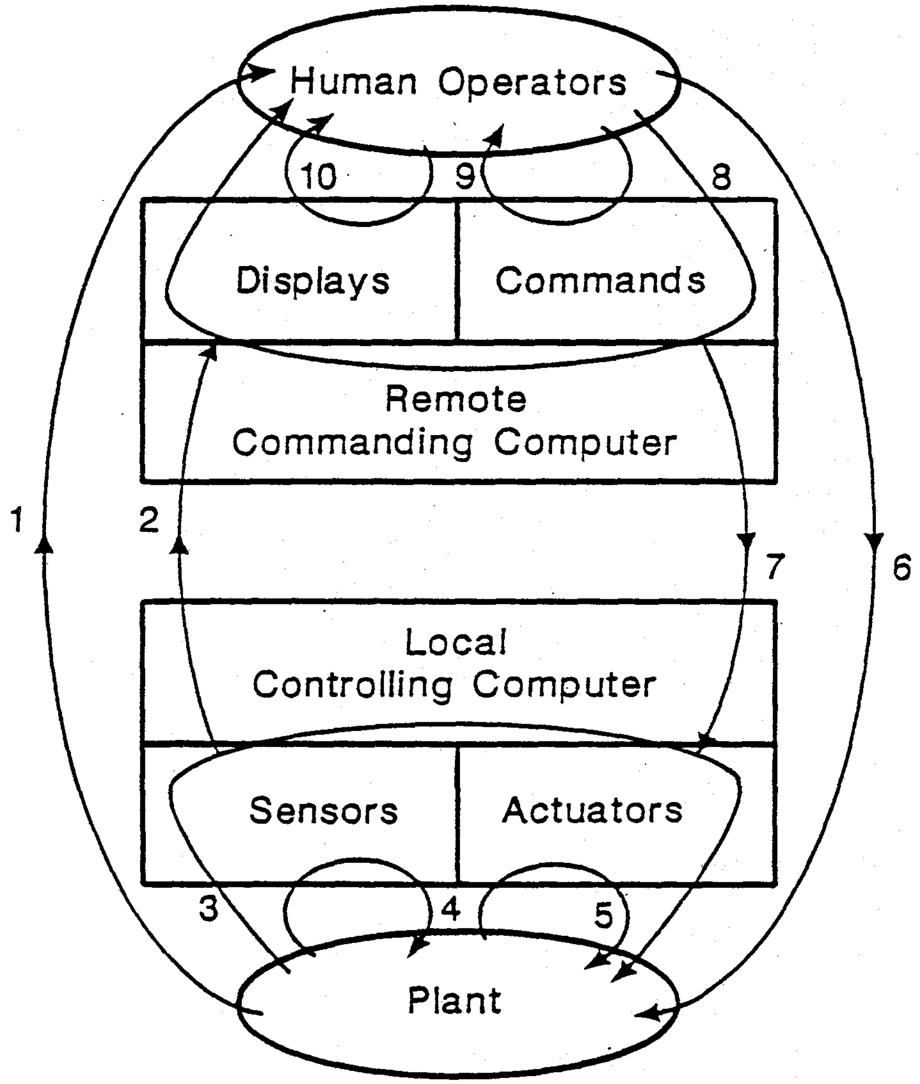
1. The plant is observed directly by the human operator's own senses (perhaps with some time delay); an example would be a video feedback channel.

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

3. The plant is controlled by local automatic mode.

4. Smart sensors interact directly with the plant.

5. Smart actuators interact directly with the plant.



SUPERVISORY CONTROL OF AUTOMATED PLANTS
FIGURE 2.1

6. The human operator directly affects the plant by manipulation.

7. The human operator affects the plant indirectly through a supervisory control interface, remote and local computers, and actuators.

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

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

10. The 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 instruction, and supply commands 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 supervisors 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 and 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 sensors and effectors and the task environment. The supervisory control loop closure is intermittent and aperiodic. Examples of missions where this mode of operation has been successfully employed include the Viking Mars lander and the recent Voyager expeditions to the outer planets. These 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 rover vehicle which is completely controlled from the Earth would be usable only 4% of the time, while a technically feasible intelligent robotic system, needing only minimum support from 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 as the local site (with 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, which appear often in the literature (Sheridan, 1988), will also be used here:

Teleoperation: Extension of a person's sensing and manipulating ability to some remote place. 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 is also intended 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. "Robot" ordinarily implies autonomy, with essentially no human interaction.

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 not necessarily imply supervisory control.

To highlight automation and control problems that may arise as the development of space production systems continues, the next section contains a brief description of some recent work 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.

2.2 Supervisory Control Experiments

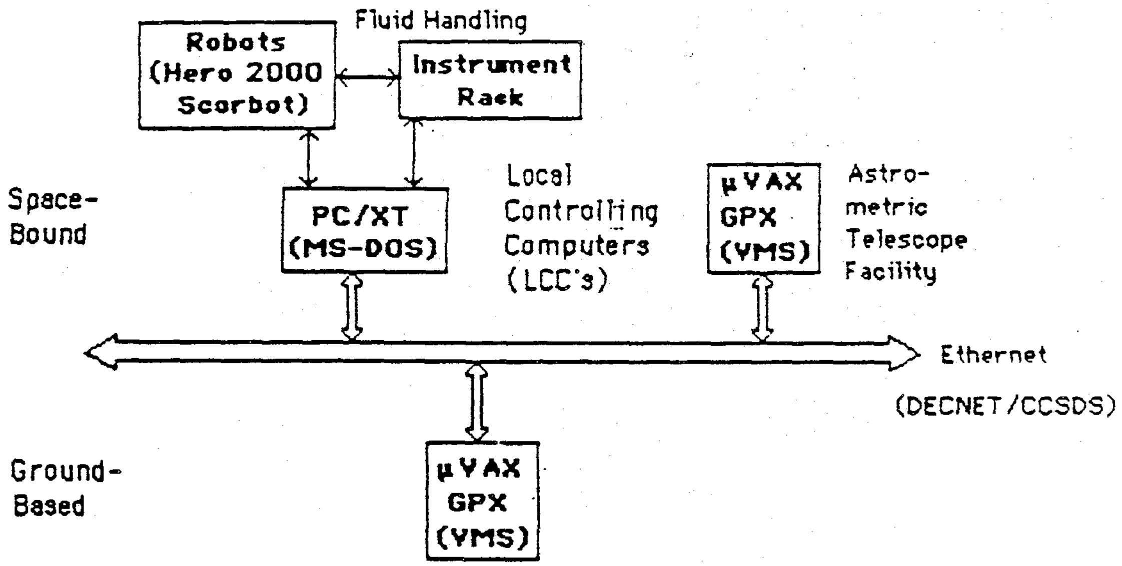
This section summarizes two recent examples of supervisory control of telerobots which were developed at the University of Arizona as part of the Telescience Testbed Pilot Program. (Figure 2.2). NASA's Office of Space Science and Applications initiated this pilot program to validate the user-oriented rapid-prototyping testbed approach in order to address a wide range of operations and information system issues. Many of these same issues arose during group discussions at the Workshop.

The first example involved teleoperation of a forerunner of the Astrometric Telescope Facility (ATF, Figure 2.3). The second involved development of systems and software for Remote Fluid Handling (RFH, Figure 2.4) in support of the microgravity and life sciences. These seemingly quite different testbed demonstrations were selected to pursue the following goals:

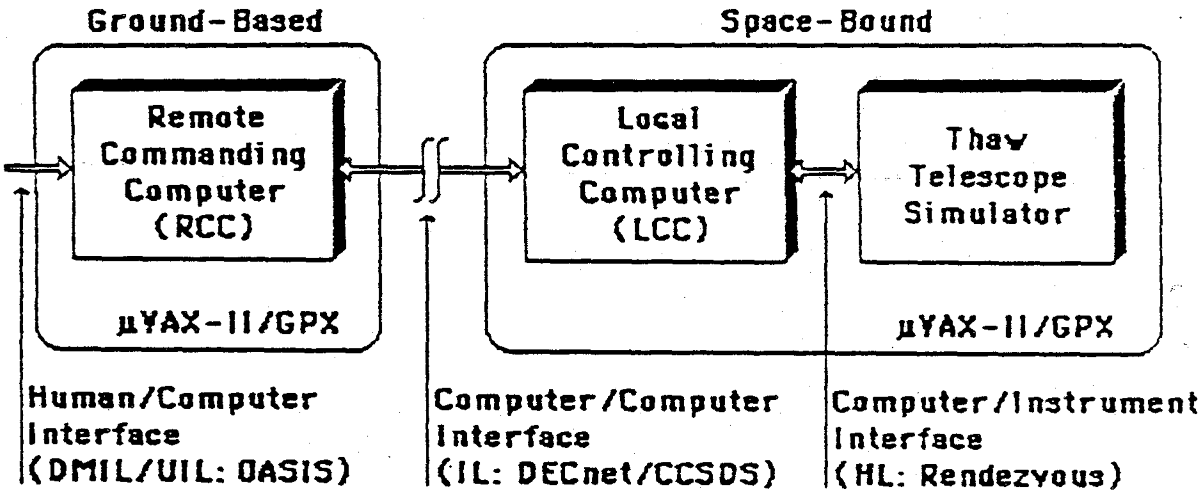
(a) To design a set of tools which would allow teleoperation of scientific experiments. These tools were to include the man/machine interface at the remote commanding computer, the machine/machine interface between the remote commanding computer and the local controlling computer, and the machine/instrument interface between the local controlling computer and the equipment (robots, telescopes, measurement instruments, analytical instruments, etc.) which could comprise any specific experiment.
(b) To ensure that these tools were generic and modular, so that they could easily be applied to a wide variety of scientific applications on the space station (or other platforms) and so that the individual modules could be revised without significant impact on the remaining parts of the software.
(c) To evaluate the technologies underlying the above developments and develop recommendations and specifications for future development.

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. Various real-time activities of the emulated telescope and representative data were then telemetered back to the first workstation for display (Schooley and Cellier, 1988).

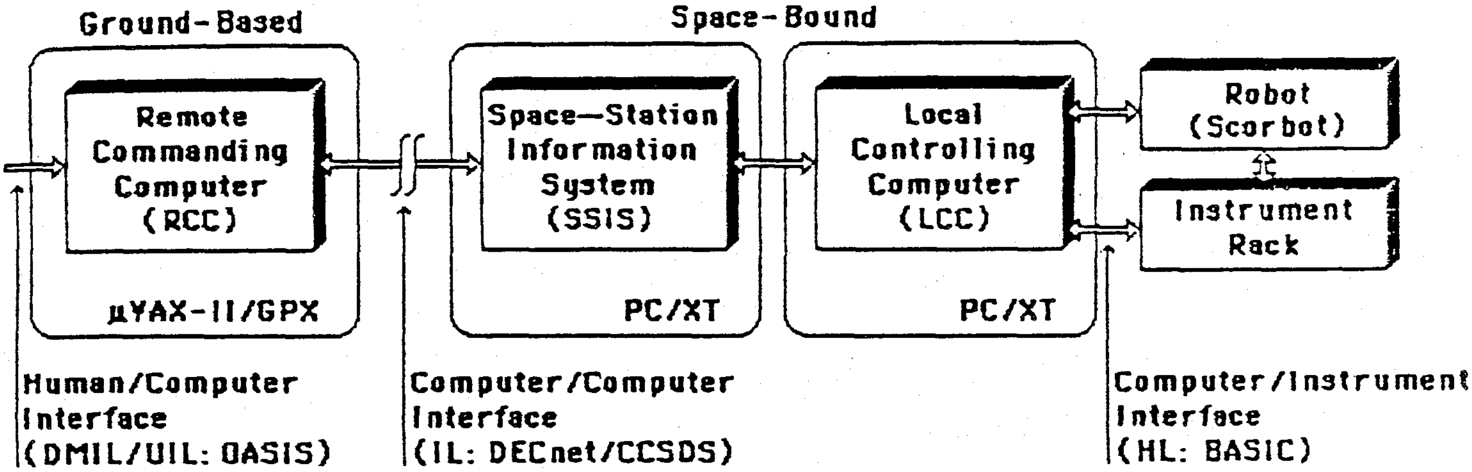
The RFH demonstration involved teleoperation of a laboratory providing automated handling and analysis of fluids, such as those which might be extracted from laboratory animals or human subjects



REMOTE COMMANDING COMPUTER (RCC)
FIGURE 2.2



ASTROMETRIC TELESCOPE FACILITY TESTBED
FIGURE 2.3



REMOTE FLUID HANDLING LABORATORY TESTBED

FIGURE 2.4

as part of the life sciences program or sent from Earth for processing as part of the microgravity sciences program. The experiments selected were: determination of the pH of a solution, and 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 communicated at 9600 baud over dialup phone lines, Sytek network, or Ethernet network with a local controlling computer (LCC). 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 (R.Davis and E.Hansen, 1988). 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 compatible PC with 640K memory, an 8087 coprocessor, a 20 Mbyte hard disk, and a LabTender multifunction board. 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 for the two demonstrations. Nearly all would be required for highly automated lunar and/or asteroidal oxygen production facilities. This emphasizes the feasibility of 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
CCDSD Depacketizer	Both
Telemetry Packet Storage	N/I
CCSDS Packetizer	Both
Ada/DECnet Interface	Both
TELECOMMAND STORAGE & RETRIEVAL	
Queue Manager	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

TELMETRY 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

This relatively simple architecture must be extended in several directions. Brief discussions of some of the more important extensions follow..

2.3 Distributed Agents

The entire testbed software is coded in Ada, and use of this language is recommended for the lunar and astroidal systems. Most of the entries in the above table are coded as separate tasks that execute in parallel. The depacketizer task accepts incoming packets, strips them of their various packet 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 from the mailbox and passes them to the local controller task, where they are scanned, parsed, and interpreted. The local controller then makes a rendezvous, with the simulator task, which performs the required action. The telemetry handler task extracts the telemetry information from the simulator, and passes this information to the packetizer task for transmission to the remote commander.

This process can be viewed as a chain of factory workers standing at a conveyor belt and performing actions on goods that pass in front of them. Each worker is responsible for a relatively simple task and knows nothing about the rest of the "world". Yet, the end product can be quite complex. The individual tasks can be executed either as separate tasks within the same program running on one CPU, or as separate programs running truly in parallel on separate CPUs.

At this point in time, the testbed software is not sufficiently robust. A single component faiure will bring the entire testbed to a halt. In one current project, however, a "supervisor" is being developed for the testbed. This is a monitor program that contains a world model and incorporates general ideas about how the overall system should work. Also, the failure simulator is currently being enhanced--from simulating intermittent communication errors only, to simulating persistent component failures as well. If the supervisor recognizes the system to be functioning differently from the expectations of the world model, it will become active, probe the system by interspersing its own commands, and observe the resultant behavior. If a particular type of error has occurred, it can then either repair the error that was produced by the failure simulator, or terminate and restart the task.

2.4 Smart Sensors

The previous section describes how overall system complexity can be enhanced while keeping the component complexity constant. It is also possible to enhance component complexity without enhancing the system complexity further by making the components "smart", as recognied by both working groups in the Workshop.

For example, in the previous case when a sensor failed, the monitor was required to observe the effects of that failure, reason about possible causes, probe the system for verification, then perform an automated repair. A smart sensor, on the other hand, would be equipped with a limited degree of diagnostic capabilities of its own. The sensor would reason about its own integrity, test itself periodically, replace the active sensory component with a backup on its own initiative, and inform the supervisor to ensure that the backup would eventually be replaced. Finally, it would be able to recognize its own limitations and call for help when needed.

Since overall system intelligence can be achieved in various ways, a problem of "intelligence balancing" must also be solved. System performance, with fewer failure occurrences, may be best with the more centralized approach, but system adaptability and invulnerability to failures may be better with the decentralized approach. The proper mix can only be determined through experimentation and by gaining actual operating experience.

2.5 Intelligent Agents

Sensors are only one type of tasking 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 required is task planning under uncertainty. In the testbed applications, all tasks have been hard-coded as sequences of primitive operations. A command issued by the remote commander directly initiates the execution of a particular sequence. More generally, task planning must be implemented. Specific tasks can be decomposed into series of unit actions, constituting an overall plan. Unit actions can be further decomposed into series of primitive operations, or an execution plan. The decomposition of unit actions into primitive operations is static (i.e., each unit action always translates into exactly the same execution plan), while the decomposition of tasks into unit actions is dynamic (i.e., goal driven).

Tasks are described as a system entity structure containing all variants that are possible for the given task in a branching "tree". A pruner selects one among the many variants and eliminates all that belong to alternate variants. It does this with the help of an expert system goal-resolver that, on the basis of desired task properties, decides which variant to choose. An action sequencer then generates the execution plan using the pruned entity structure, and a task optimizer checks this plan for possible means of optimization, a functionality similar to code optimizers in language compiler design.

It may also be necessary to implement smart task executors (i.e., intelligent hierarchical controllers) and algorithms for cooperation between several robots sharing in the execution of a task (Zeigler, et al., 1989). The overall system architecture can be viewed as an implementation of large portions of the NASREM telerobot architecture (Albus, et al., 1987), which decomposes such a control system into three columns: sensory, control, and central world-model;. and into six rows of increasing/decreasing complexity. The lowest row is that of primitive operations while the highest level features overall mission control.

2.6 Communication Requirements

The communication system must allow for flexibility of observation and control from more than one location, as well as for the possibility that several such locations may be active simultaneously. 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. Thus, some method of conflict resolution must also be implemented.

Time delay is critical but largely beyond the control of system designers, at least over large distances. Nevertheless, techniques of telepresence (to assess advance environment) and teleoperation (featuring virtual motion) can be applied, and processing delays in the local controlling computer, the remote commanding computer, and any computers in the communication system itself, can be minimized.

Communication bandwidth requirements will be quite demanding. The following quotation from the conlusions of the Telescience Testbed Pilot Program give some idea of the magnitude of the problem (Leiner, 1989):

> For those functions which involve time critical monitoring and control 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/telementry 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 (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.

2.7 Special Environments

As noted earlier and emphasized later in this report, system design must begin and end with the special environmental conditions of the particular plant sites: microgravity, charge buildup, dust, vacuum, temperature extremes, radiation, micrometeorite showers, and a number of other unique aspects of lunar, asteroidal or Martian environments. These factors, along with the bandwith requirements and high data rates needed, imply that optical fiber communication technology should be employed at the plant site, and either optical or extremely high frequency radio transmission technology should be utilized between the local controlling computer and the remote commanding computer locations.

In any event it will be apparent that such considerations must constitute an integral part of the design of automated/automous space production facilities. Computer-mediated communication systems tailored to their unique and evolving requirements are essential, and must be developed in parallel.