Modeling, Simulation, and High-Autonomy Control of a Martian Oxygen Production Plant

L.C. Schooley, F.E. Cellier, F.-Y. Wang, B.P. Zeigler

Department of Electrical and Computer Engineering
and
Department of Systems and Industrial Engineering
The University of Arizona

Abstract

This report documents progress on a project for the development of a high-autonomy intelligent command and control architecture for process plants used to produce oxygen from local planetary resources. A distributed command and control architecture is being developed and implemented so that an oxygen production plant, or other equipment, can be reliably commanded and controlled over an extended time period in a high-autonomy mode with high-level task-oriented teleoperation from one or several remote locations. During this reporting period, progress has been made at all levels of the architecture. At the remote site, several remote observers can now participate in monitoring the plant. At the local site, a command and control center has been introduced for increased flexibility, reliability and robustness. The local control architecture has been enhanced to control multiple tubes in parallel, and has been refined for increased robustness. The simulation model has been enhanced to full dynamics descriptions.

Introduction

A distributed command and control architecture is being developed that ultimately will provide the capability to teleoperate one or several process plants located on Mars, Luna, the asteroids, and/or other objects in space in a supervisory control mode from one or several locations on planet Earth.

The architecture must be able to guarantee reliable and robust control over an extended time period in a high-autonomy mode with high-level task-oriented teleoperation. The architecture must be resilient to temporary breakdowns in communication links, and must be able to accommodate a varying number of remote participants and local plants to be controlled. New remote observers should be able to join the operation at any time, while others may sign off. New plants can be added to the control umbrella at any point in time, while others can be removed after they have accomplished their respective goals.

These demands can be satisfied by a systematic and consistent application of the object-oriented programming paradigm. Each module within the overall command and control architecture acts as an independent intelligent agent. No bilateral links are established that will endanger the overall architecture by spreading a "disease" of one agent across the network to other agents. Each agent must be able to function autonomously, though not necessarily in an optimal manner. Agents can request information from their environment for improved operation, but they have no knowledge of the current network configuration or of who might be out there to fill their request. If the request is filled, the operation of the agent will improve, if not, it must be able to function with the resources that are at its disposal at the time. Each agent carries a model of itself that allows it to estimate its current performance level.

These are the ultimate goals of the project, not a reflection of the current state-of-the-art. However, the reporting period has brought us a good deal closer to realizing these goals in comparison with the state of affairs one year ago (Schooley et al., 1991).

In this report, we shall first explain the overall command and control architecture, then look at the individual parts of this architecture starting at the site of the remote commander and successively progressing towards the plant site.

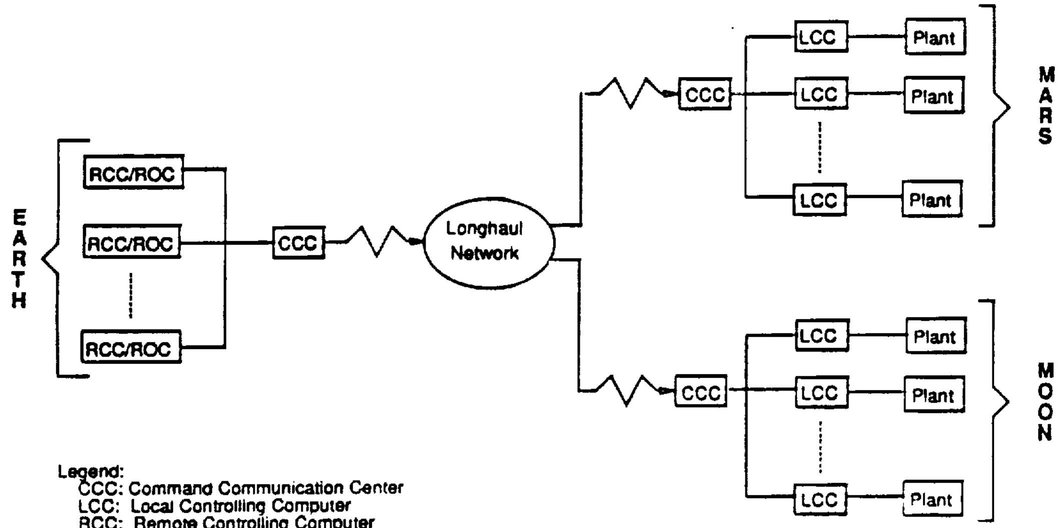
The Overall Command and Control Architecture

Figure 1 shows the overall command and control architecture. Each site is equipped with a Command Communication Center (CCC). This is the gateway to the longhaul communication network and manages the resources available at the site. The reliability of the CCCs themselves can be guaranteed by standard technology such as resource duplication. Each participating plant or operator communicates with its own CCC only through a plant (operator) interface computer.

On Earth, the "remote" site in this document, each operator communicates to the CCC through a workstation running the Oasis software. The operator workstations are called Remote Commanding Computer (RCC) and Remote Observing Computer (ROC) respectively. There is no fundamental difference between the two types of workstations. They run the same software. They differ only in the privileges given to the operator of the workstation. The privileges are a resource that is maintained by the CCC. Additional privileges can be requested from the CCC at any point in time. The CCC will grant these privileges if the operator of the workstation has a sufficiently high clearance, and if granting these privileges is not in conflict with other demands. For example, no plant should be commanded by several remote commanders simultaneously in an intrusive fashion since it can be expected that, due to a lack of communication between the remote commanders, such an operation would lead to confusing situations.

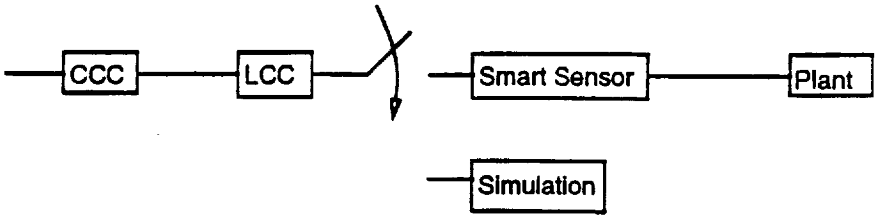
In the current implementation, only one CCC (located at the "local" site, i.e., planet Mars) manages both the resources of the plant as well as those of the remote operators (Doser, 1992). This is somewhat awkward since the different remote operators need to communicate with each other via the longhaul network (through planet Mars), but we didn't have an extra computer available to be dedicated as a second CCC. Conceptually, this problem is easy to fix. The operator privileges are symbolized in our current implementation by a privilege "key," that is maintained by the CCC. This key can be requested by remote operators to establish them as temporary remote commanders. If the key has been requested, but is currently in use by another operator, the CCC will inform the current key holder of the request. It is then the responsibility of the key holder to relinquish the key when it is no longer needed.

On Mars, the "local" site in this document, the oxygen production plant communicates with its CCC through the so-called Local Controlling Computer (LCC). As shown on *Figure 2*, even the LCC



Legend:
CCC: Command Communication Center
LCC: Local Controlling Computer
RCC: Remote Controlling Computer
ROC: Remote Observing Computer

Figure 1. Global Remote Command and Control Architecture



Legend:
CCC: Command Communication Center
LCC: Local Controlling Computer

Figure 2. Local Control Architecture

comprises a distributed control architecture (Farrenkopf and Doser, 1992).

The plant itself is interfaced with a hardware controller (the so-called functional interface) that monitors and controls the current and voltage levels of each tube, and time-multiplexes the monitored signals for consumption by the next higher-level controller, the so-called smart sensor (Taglianetti, 1992). The duties of the functional interface could be performed by the smart sensor itself, but individual control of each tube by the smart sensor would have meant buying more interface cards for the smart sensor, a solution that was deemed undesirable due to its high cost and due to the amount of space that would have been consumed by the interface. The hardware solution is much cheaper and much more compact, albeit less flexible. The smart sensor is responsible for implementing low-level control strategies. The LCC itself translates high-level task-oriented commands received from the CCC into sequences of low-level commands, downloades the corresponding low-level control programs into the smart sensor, and initiates the control action by enabling the smart sensor control. Notice that, on *Figure 1*, the term LCC denotes the overall local control architecture, whereas, on *Figure 2*, LCC is only a part of the local control architecture.

A full-dynamics simulator has been added as an additional component to the local control architecture (Farrenkopf, 1992). It can be used for three different purposes:

1. To design and test new control strategies without risking the destruction of any tubes;
2. As a "second plant" without incurring the cost of actually building a second plant, allowing to demonstrate the capability of the control architecture to deal with multiple plants; and
3. As a model for the intelligent control to estimate its own level of performance.

Since the full-dynamics model was completed only recently, it has not yet been used for purposes other than #1 above.

The Remote Site

In the reporting period, the remote site has been enhanced to accommodate several ROCs beside from the RCC. A key passing mechanism has been introduced that allows ROCs and RCC to exchange their roles (Doser, 1992). Previous reliability issues have been resolved. It is now possible to switch any ROC on and off arbitrarily during an experiment without jeopardizing the overall mission. If the RCC goes down, the CCC immediately retrieves the privilege key and informs the LCC that it is currently on its own. When the key is requested by another remote workstation, that workstation becomes the new RCC, and the LCC is instructed to again operate under remote control.

The Command and Communication Center

The CCC is a new element in the overall Command and Control Architecture (Doser, 1992). It serves three purposes:

1. **Software-decoupling the individual computers from each other.**
   Each interface computer (be it an LCC or an RCC) needs to know only the language of its client (the plant or the operator) and the language of the CCC. Different interface computers need not know anything of each others characteristics and physical location, how many such computers are in the system, and how they operate. Each LCC can talk a different language to its client, and in principle, the same holds true for each RCC (although different RCC interfaces may not be desirable for other reasons).

2. **Managing the resource umbrella of its clients.**
   Each CCC is respondible for managing the limited resources of its clients (the RCCs and the LCCs). A restricted resource of the RCCs is their privilege level, implemented through the privilege key in the current prototype. Restricted resources of the LCCs may be the amount of energy to be used at any one time or the useable communication bandwidth between the LCC and its CCC.

3. **Managing the communication with the other CCCs.**
   Together, the CCCs manage the longhaul communication network.

Since there is up to now only one CCC in the prototype, the third function is not currently implemented. Consequently, it is still the duty of each RCC/ROC to worry about whether its commands have been received by the CCC (located at the other end of the longhaul network) or need to be resubmitted. In the future, the time delays between the RCC/ROCs and their closest CCC will be short, and a simple message acknowledging protocol can be enabled so that the RCC/ROC need not keep track any longer of multiple commands that have been submitted but not yet acknowledged. It will then be the duty of the CCCs to ensure proper transmission of commands and telemetry packets across the inherently less reliable longhaul network.

The Local Control Architecture

The Local Controller

The LCC is the heart of the autonomous control system designed for the Oxygen Production Plant four-tube unit. The control system, which is placed at the site of the plant (planet Mars), includes the LCC (Local Controlling Computer) and the Smart Sensor (Intelligent Controller). The LCC communicates with its CCC (Command Communication Center) to receive control parameters and sends telemetry data back to the RCC (Remote Commanding Computer) on Earth. Users' commands include commands to set control parameters, telemetry data requests, system start-up commands, and system shut-down commands.

The Communication Protocol

In the current reporting period, the LCC has been enhanced to communicate with other computers/controllers simultaneously with executing the local control procedures. Thus the communication process is included as part of the control program. *Figure 3* shows the overall protocol for the communication between the LCC and CCC.

The LCC, physically a PC-386, first sets up communication links to the Smart Sensor in order to initialize the sensors and actuators of the plant. Then it tests the communication link to the CCC. Upon receipt of the control parameters and the start-up command from the CCC, the LCC begins to control the plant.

The Intelligent Control Strategy

To implement autonomous control of the plant, we have developed a real-time expert system that has rule bases to govern normal control operation as well as fault detection, diagnosis and recovery. *Figure 4* shows the decision process for control of start-up and steady-state operations.

In order to operate in real-time, the inference engine of the expert system has a clock to check time constraints on control rules. When the real-time expert system receives tasks from the RCC, it schedules the appropriate control actions to execute them. During the execution, it continually monitors the state of the plant.

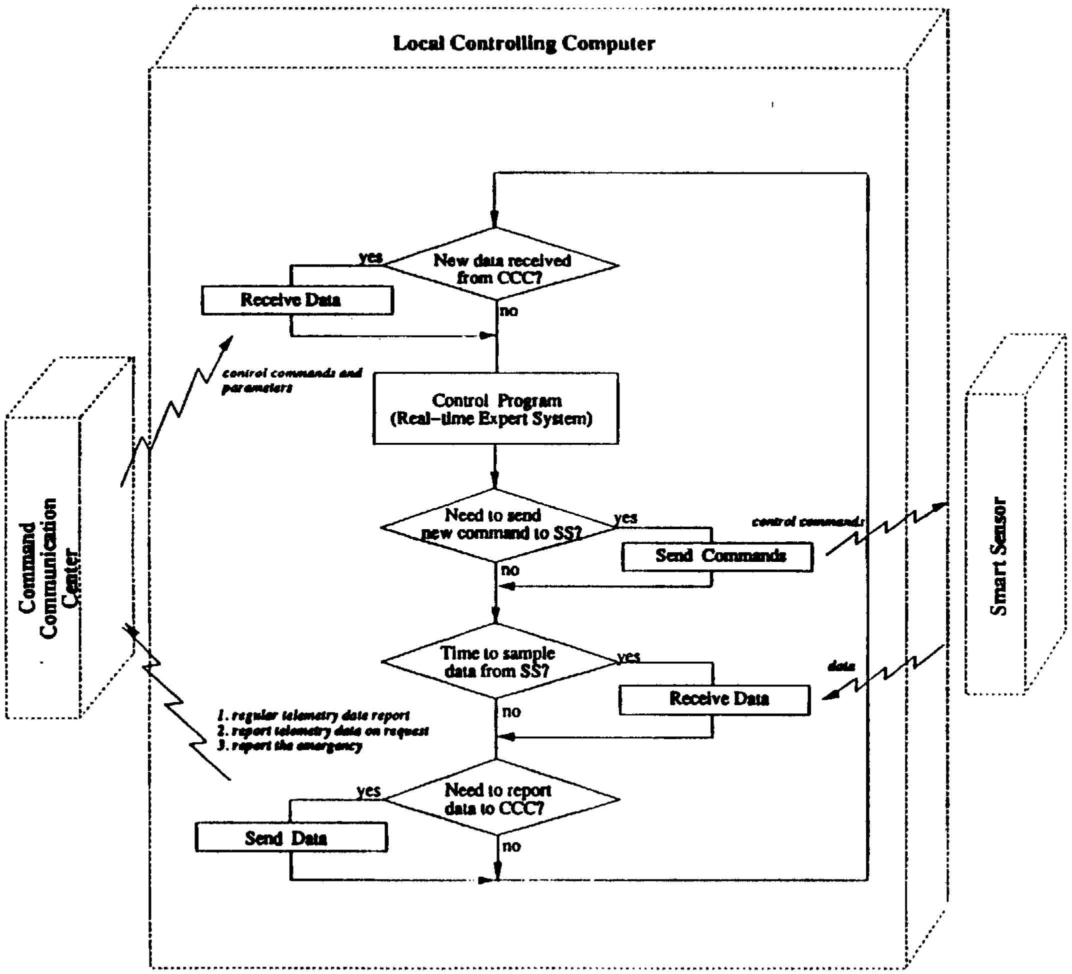

Local Controlling Computer
Command Communication Center
Smart Sensor
New data received from CCC?
yes
no
Receive Data
control commands and parameters
Control Program (Real-time Expert System)
Need to send new command to SS?
yes
no
Send Commands
control commands
Time to sample data from SS?
yes
no
Receive Data
data
1. regular telemetry data report
2. report telemetry data on request
3. report the emergency
Need to report data to CCC?
yes
no
Send Data

Figure 3. Protocol in the LCC for Communicating with the CCC and the Smart Sensor

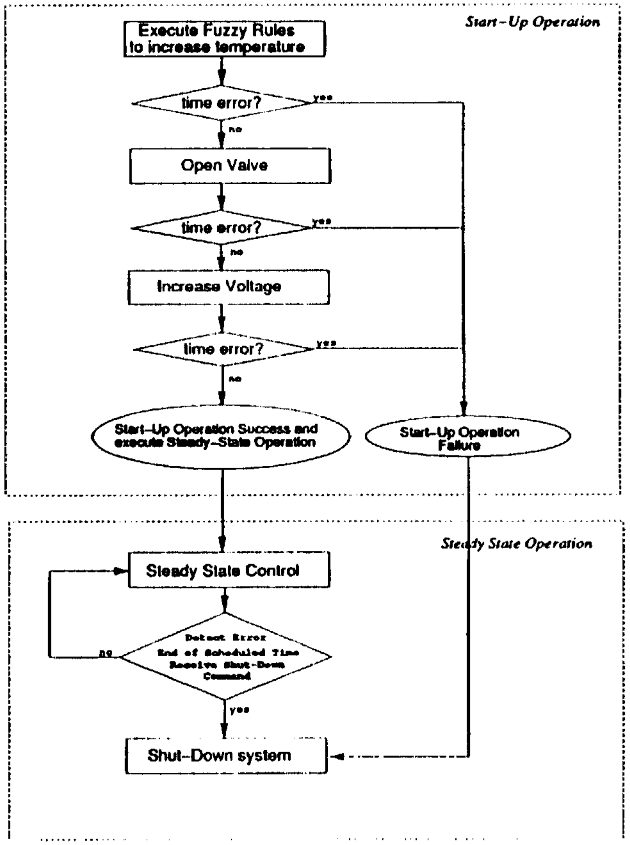


Figure 4. Flow Diagram of Start-Up and Steady-State Control Scheme

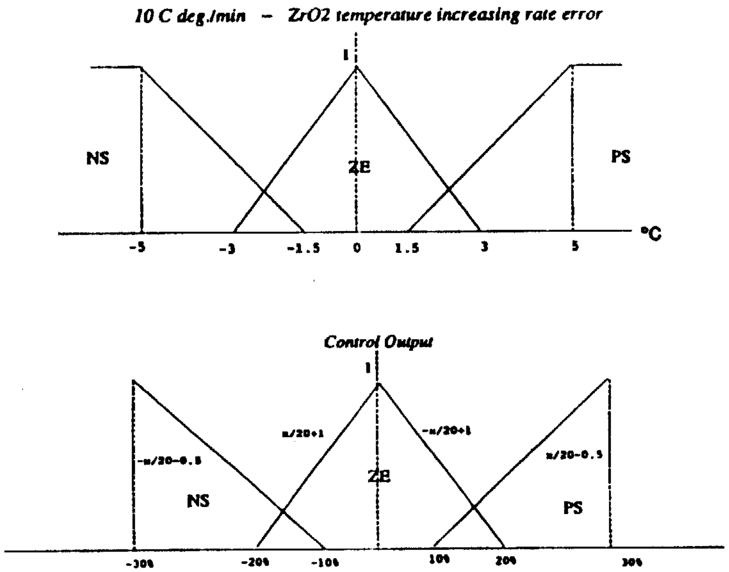
The LCC needs to control three kinds of state variable of the plant, viz, temperatures of the ZrO2 tubes, flow rate of the CO2 gas and voltages across the ZrO2 tubes. Since the unit must operate at high temperature, a control algorithm is used to increase the temperature as fast as possible while still preventing the seal between metal pipe and ZrO2 input tubes from breaking. An increase of 10 C deg/min was set as the goal rate for the control algorithm.

We have integrated fuzzy logic (Cellier and Mugica; Lee, 1990) with event-based control (Zeigler, 1989; Zeigler and Kim) to design the temperature controller. When the LCC receives a goal temperature from the RCC, the expert system sets a number of check point temperatures between the initial temperature and goal temperature intended to assure that the desired rate of temperature increase is maintained. A pair of successive check points can be considered as a small segment that must be successfully controlled to reach the final goal temperature. Event-based control is employed for each segment, i.e., after reaching one check point and issuing a control output to reach the next, the latter should be reached within a pre-estimated time window. If the next check point temperature is reached too early or too late, this represents a fault situation that the system needs to diagnose. At each check point, the fuzzy logic controller computes an appropriate control command, which is realized as a pair of on/off heater durations (for example <2,3> might mean heater on for two seconds, then off for three seconds.) High thermal mass in the plant necessitates this kind of operating regime (that is, the ZrO2 temperature tends to lag and overshoot the nominal value). A fuzzy logic approach to this problem is useful in view of the difficulty of developing exact models for the thermal behavior. Although, a full-dynamics simulation model has been recently developed, it has not yet been fully validated against the real plant. The integration of event-based and fuzzy logic control paradigms provides a powerful new approach to ensuring and testing for "hard" real-time control subgoals, (offered by event-based control) while providing for flexible behavior-based "soft" control responses (offered by fuzzy control).

To design the fuzzy control algorithm, we have employed fuzzy membership functions (e.g. Negative Small, Negative Zero, Positive Zero, Positive Small) for the state variable, rate of ZrO2 temperature increase. Calibration of the fuzzy logic membership parameters, as shown in *Figure 5*, was obtained from experiments with a mock-up of the real system (representing the proper thermal mass of the system).

Figure 6 shows a typical temperature control experiment using the integrated fuzzy

NS : Negative Small
ZE : Zero
PS : Positive Small



	[on/off] time							
Fuzzy Output	0 ~ 50	50 ~ 300	300 ~ 500	500 ~ 600	600 ~ 680	680 ~ 740	740 ~ 800	°C
	9/1	4/1	1/1	2/1	1/1	7/1	14/1	
25	1/2	2/3	2/1	3/1	2/1	6/1	14/2	
14	1/3	1/2	2/2	3/2	3/1	5/1	12/1	
0	1/4	1/3	2/3	3/3	3/2	5/2	11/2	
-14	1/5	1/4	2/4	3/4	3/3	5/3	11/3	
-25	1/9	1/5	2/5	3/5	3/4	5/4	11/4	

Figure 5. Fuzzy Membership Function and Defuzzified Control Output

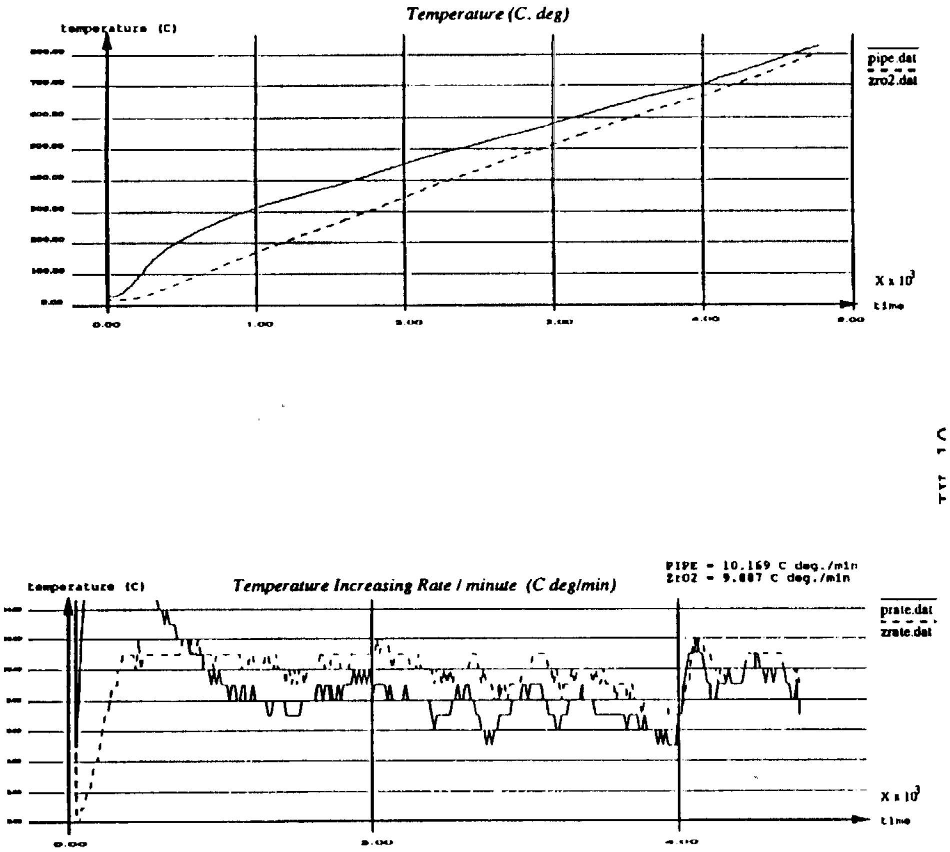


Figure 6. Temperature Control Experiment Using Fuzzy Logic and Event Based Control

logic/event-based controller.

As indicated before, the option of replacing a real plant with a dynamic simulation also allows us to perform calibration experiments in an off-line manner.

When steady-state operation has been reached, the behavior of each state variable is controlled by a dedicated watchdog monitor (Cellier et al, 1992). Each watchdog monitor has upper/lower limits for its state variable. Violation of these limits causes the watchdog monitor to activate diagnosis rules to execute appropriate recovery actions. If recovery is not possible, the expert system automatically executes a benign shut-down procedure to protect the plant from further damage. The LCC and SS (Smart Sensor) together form a two level hierarchical control architecture. The reasoning and high-level logic just described are realized in the LCC in an expert system shell written in C. On the other hand, the Smart Sensor contains simple control programs in memory and executes low-level control tasks in accordance with the parameters received from the LCC. This bi-level partition of functionality enables fast local control under the guidance of slower, more global, intelligent control.

Fault Management

As shown in *Figure 7*, interspersed with the control tasks just described, the LCC reads sensory data from the SS and reports telemetry data to the CCC to update the RCC's data base.

If the LCC detects a faulty situation, it reports the fault detection to the RCC and starts its diagnosis inferencer. During the diagnosis process, the LCC sends data more frequently to the RCC, which represent the error state transition behavior of the plant. This alerts the user at the RCC to the deviant behavior of the plant. On such an occasion, or indeed, at any time, the RCC can route new commands or parameter values through the CCC to the LCC to re-schedule control tasks and/or to establish new control set points. Also, the RCC can send shut-down commands to interrupt the system. This can happen even while the system is in start-up mode.

Sensor Fusion and Process Fault Diagnosis Using Neural Networks

The nature of extraterrestrial environments makes the automatic process fault diagnosis an essential prerequisite for the further development of supervisory control system of the Martian Oxygen Production Plant. This fault diagnoser must be very reliable. Since it is impossible to foresee all faults that might occur, it is desirable to build some redundancy into the fault monitoring and diagnosis procedures. For this reason, a second process fault diagnosis system using multiple sensors for data acquisition and neural networks for information processing has been developed (Wang; Wang and Wu, 1992). The execution of this system has been divided into three stages:

1. **Fault Detection:**
   At this stage a possible process fault is detected by checking if particular measurable or estimable variables are within a certain tolerance of the normal values. For example, the measured temperature of the Zirconia tube should be above 790K but below 815K under normal situation. If this check is not passed, it leads to a fault message that activates the next stage of fault diagnosis.

2. **Fault Diagnosis:**
   The fault is located and the cause of it is established at this stage using a multilayer back-propagation neural network that fuses the data from several sensors. For example, output O2 flow rate is too low because the input CO2 valve is only partially opened. Eight representative fault situations have been selected in a simulation study.

3. **Fault Evaluation:**
   An assessment is made of how the fault identified in the second stage will affect the production process. The faults have been classified into different hazard classes according to a simple fault tree analysis. After the effect of the fault is determined, a decision on the actions to be taken will be made. If the fault is found to be tolerable, the production process may continue. If it is conditionally tolerable, a change of operation has to be performed by either modifying the local control algorithm or sending a request to the higher level of control for guidance. However, if the fault is intolerable, the process will be shut down immediately and an emergent request will be made to the higher level to eliminate the fault. For example, if a malfunction in the heater has been determined to be the cause of high temperature, the operation will be stopped immediately and a request for changing the heater will be made.

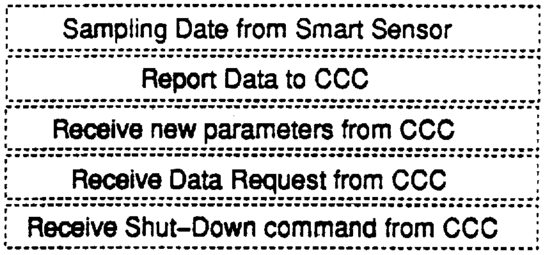
A simulation study has been performed based on a computer model of the Martian Oxygen Production Unit. Ten sensor readings were used for sensor fusion in a neural network. One thermocouple transducer is located inside the Zirconia tube. On each of the CO2, O2, and CO2/CO pipes, one thermocouple, one pressure, and one flow rate transducers are located. All readings are scaled to a range from -1.0 to +1.0. The scaling makes the sensor fusion easier to perform because the original measurement data contains both small and large values. Eight representative fault situations have been chosen in the simulation:

1. CO2 valve partially opened;
2. O2 valve partially opened;
3. CO2/CO valve partially opened;
4. Thermocouple transducers broken;
5. Leak flow in tube;
6. Malfunction in heater;
7. CO2 flow rate too high; and
8. CO2 flow rate too low.

A multilayer feedforward neural network with one hidden layer has been used. The input layer has ten nodes representing the ten sensor readings, and the output layer has eight nodes - one for each of the selected fault situations. The hidden layer has six nodes. The standard sigmoid has been used as the activation function for the output neurons, while the hyperbolic tangent has been used for hidden neurons in order to speed up the learning process. The training data consist of 450 measurement patterns (each pattern contains 10 sensor readings), 50 patterns for the normal production operation and for each of the fault situations. For the normal operation, the network should produce a value near zero (< 0.1) at all the output nodes, whereas for the situation of fault number one, the network should produce a value near one (> 0.9) at the first output node and a value near zero (< 0.1) at the rest of the output nodes, and so on. The learning is conducted using the conjugate back-propagation algorithm.

The simulation programs used are PlanNet and Matlab. The network has been trained 10,000 times to learn the specified fault situations correctly. For 100 additional measurement patterns that are not in the training data, 83 patterns have been classified correctly. *Figure 8* illustrates an example of diagnosis of the fault situation number 5.

It is planned to integrate this fault diagnoser with the actual plant, but this has not been done yet.



Sampling Date from Smart Sensor

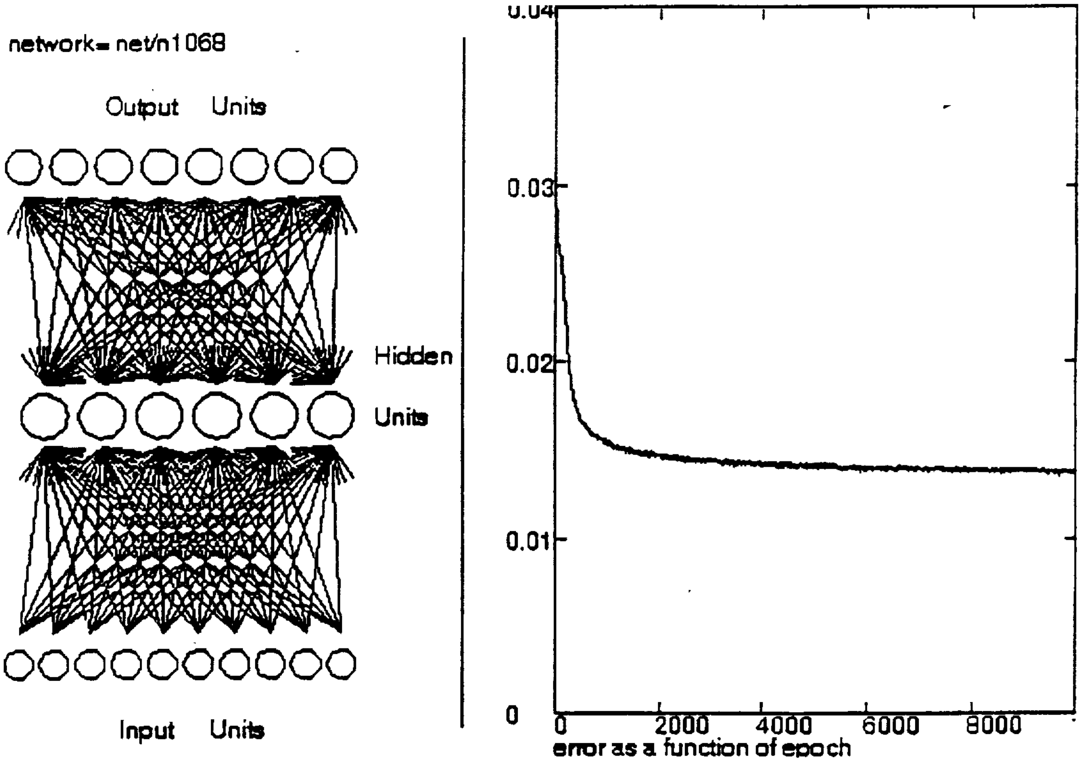
Report Data to CCC

Receive new parameters from CCC

Receive Data Request from CCC

Receive Shut-Down command from CCC

Figure 7. Demon Processes for the Control System



input	0.264	0.032	-0.397	0.135	0.721	-0.127	-0.569	0.173	0.357	-0.523
output	0.092	0.012	0.187	0.061	0.801	0.009	0.011	0.014		
desired	0.005	0.005	0.005	0.005	0.950	0.005	0.005	0.005		

Figure 8. Network Architecture, RMS Output Error During 10,000 Training Epoches, and Example of Diagnosis of Fault #5

Full-Dynamics Model of the Martian Oxygen Production Plant

Modeling Methodology

A full-dynamic simulation model of the four-tube oxygen production system has been completed. This model is based on a modeling and design methodology involving the use of Bond Graphs (Cellier, 1991). This method is advantageous because the power that moves through the system is topologically as well as mathematically represented in the model. In addition, the power flow in the thermal dissociation of carbon dioxide into oxygen and carbon monoxide can also be modelled using Bond Graphs, and this model can be conveniently connected to the thermal model of the overall system to obtain a full-dynamic system model that can be used to simulate both steady-state operation (flow equilibrium) as well as start-up and shut-down phases correctly.

System Description

The actual system topology consists of concentric cylinders of different materials that perform different tasks. The system topology is shown on *Figure 9*. A cylindrical ceramic heater surrounds a metal pipe, but is itself surrounded by a cylinder of insulation. Carbon dioxide gas enters the system through an alumina pipe and then passes over the surface of the surrounding Zirconia tube.

These cylinders can be described in terms of thermal resistance and capacitance elements. Thermal resistance elements are dependent on the shape, size, direction of entropy flow, and thermal conductivity of the material comprising the cylinder, and thermal capacitance elements are dependent on the density, specific heat, temperature, and volume of the cylinder (Hollman, 1986). Since the power flow through the system is modeled, the resistance to entropy flow is used instead of the resistance to heat flow because the product of entropy flow and temperature is of type power. The temperature dependence of the resistance and capacity values are accounted for. The temperature dependent relations for the specific heat of the different materials are available in the literature (Perry and Green, 1991) and were used in this model. Some of the cylinders consist of gas, which means heat flow due to convection as well as conduction must be accounted for. The radiation heat transfer from the pipe in contact with the heater to the Zirconia tubes is also modeled.

A lumped-heat-capacity type model has thus been constructed where the temperature of the individual elements is assumed to be uniform. If a temperature distribution at several points along any of the cylinders is of interest, then the cylinder can be divided into several subsections, each with its own resistance and capacitance properties. Our system is fairly symmetric so the

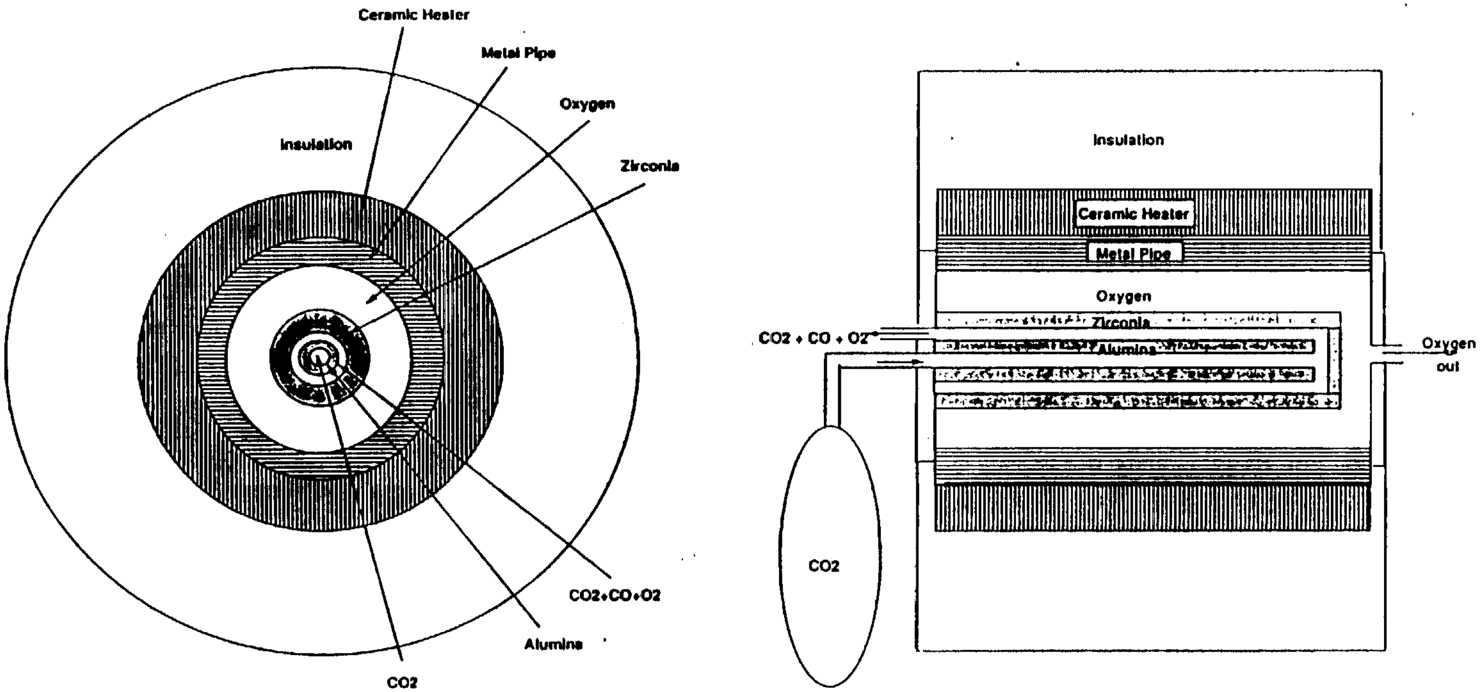


Figure 9. Topology of Zirconia Tube

temperature distribution along the cylinders is expected to be approximately uniform; the cylinders were thus divided into at most two sections. This allows for a smaller, less complex model with better simulation response, yet with acceptable accuracy. The Bond Graph thermal model of the system is shown on *Figure 10*.

Rather than include four identical tube descriptions in the model, the thermal characteristics of a one-tube description were changed to reflect the presence of four tubes. For a four-tube system, there is four times as much tube component mass, so the thermal capacitance values for the one-tube system are multiplied by four (remember, thermal capacitance is linearly proportional to volume). Similarly, a four-tube system will provide four times the load, so the resistance to entropy flow for the tube system components was divided by four.

Chemical Reaction Model and Oxygen Production

The process of the thermal dissociation of carbon dioxide into carbon monoxide and oxygen is essentially a chemical reaction, which can be easily modeled using the Bond Graph methodology. Chemical, thermic, and hydraulic/pneumatic forms of power are considered here. The number of moles of the reactant and the volume in which the reaction occurs are known. This model is modularly connected to the tube model at the point where the input gas has been heated. The molar flow rate of oxygen produced is then multiplied by the efficiency of the Zirconia tube separation process, which is a temperature and applied tube voltage dependent quantity. The system's oxygen production rate is thus predicted. The Bond Graph model of the chemical reaction system is shown on *Figure 11*.

The model has been calibrated for the one-tube system. In this case, the dynamical behavior of the model and the real system are very similar indeed. The four-tube model has not yet been calibrated since no measurements for the four-tube system are available yet.

Future Work

The future work on remote supervisory control of high autonomy ISMU process plants will build on the strong base established in the past. It is essential to next address the problems of increasing autonomy by introducing carefully balanced concepts of redundancy, autonomous diagnosis and repair capabilities, and graceful degradation of performance with reduced resource availability. The focus will be on further extension and development of core technology applicable to diverse

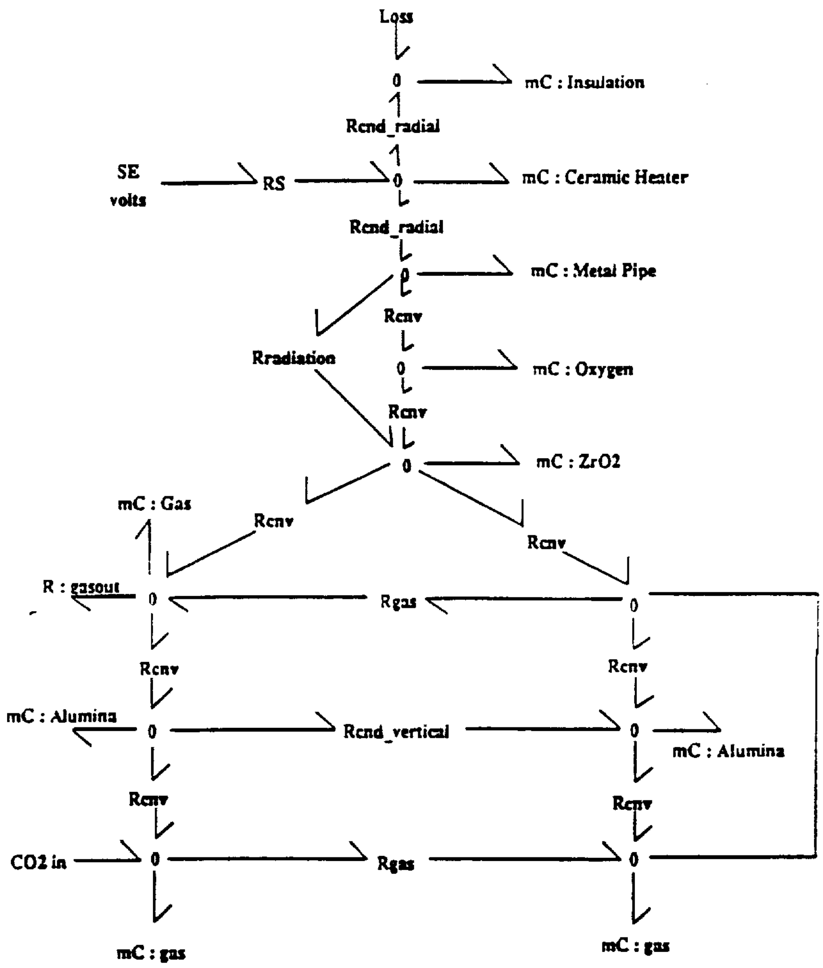


Figure 10. Thermal Bond Graph Model of Zirconia Tube

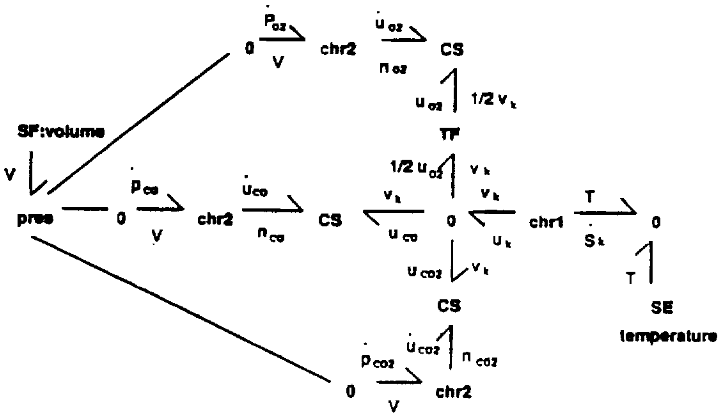


Figure 11. Chemical Reaction Bond Graph of Zirconia Tube

processes and plant designs. This will include:

1. Software engineering of common modules that are easily adaptable to different applications.
2. Implementation of advanced concepts in the higher levels of hierarchical planning, sensing, control, and exception handling, emphasizing additional knowledge-based, neural network, and fuzzy artificial intelligence concepts for increased autonomy and graceful degradation of performance with reduced resource availability due to resource sharing and/or resource failure.
3. More robust interaction with remote commanders under conditions of realistic time delays in the communication system. Real-time communication system performance will be analyzed by means of timed petri nets.
4. Integration of computer vision and other advanced sensory capabilities (including sensor fusion) for world state assessment as well as fault detection, diagnosis, and recovery.
5. Incorporation of light weight, low power, flight/space qualified hardware in the computer/communication systems.
6. Further development of software design tools for optimum process design and mass/power reduction.

References

Cellier, F.E. *Continuous System Modeling* (1991). Springer-Verlag, New York.

Cellier, F.E., and F. Mugica Alvarez. Systematic design of fuzzy controllers using inductive reasoning techniques. Submitted to *IEEE Transactions on Control System Technology.*

Cellier, F.E., L.C. Schooley, M.K. Sundareshan, and B.P. Zeigler. Computer-aided design of intelligent controllers: Challenge of the nineties (1992) in: *Recent Advances in Computer Aided Control Systems Engineering* (M. Jamshidi and C.J. Herget, eds.). Amsterdam: Elsevier Science Publishers.

Doser, A. Multiple user communications for telescience (1992). M.S. Thesis: Department of Electrical and Computer Engineering, University of Arizona (available also as: *Technical Report TSL-028/92*, Telescience Laboratory, Electrical and Computer Engineering Department, University of Arizona, Tucson, Arizona.

Farrenkopf, G. Full-dynamics simulator of the Martian oxygen production prototype (1992). Technical Report, Telescience Laboratory, Electrical and Computer Engineering Department, University of Arizona. In preparation.

Farrenkopf, G. and A. Doser. Remote command and simulation of an oxygen production plant. *SERC Newsletter.* 3 (1992) 1: pp. 5-6.

Hollman, J.P. *Heat Transfer*. New York: McGraw-Hill, 1986.

Lee, C.C. Fuzzy logic in control systems: Fuzzy logic controller -- part I. *IEEE Transactions on Systems, Man, and Cybernetics*. 20 (1990) 2.

Perry, R.H., and D.W. Green, eds. *Perry's Chemical Engineer's Handbook*. New York: McGraw-Hill, 1991.

Schooley, L.C., F.E. Cellier, B.P. Zeigler, A. Doser, G. Farrenkopf, J. Kim, Y. Pan, and B. Williams. Automation and control of an oxygen extraction plant. *Annual Report* (1991) NASA/SERC.

Taglianetti, F. The functional interface of the distributed intelligent controller to the Martian oxygen production plant prototype. Senior Project (1992), Department of Electrical and Computer Engineering, University of Arizona.

Wang, F.-Y. Building knowledge structure in neural nets using fuzzy logic. Submitted to *Journal of Fuzzy and Intelligent Systems*.

Wang, F.-Y., and F. Wu. Sensor fusion and process fault diagnosis for Martian oxygen production plant using neural nets. *Technical Report #42* (1992), Robotics and Automation Laboratory, Department of Systems and Industrial Engineering, University of Arizona.

Zeigler, B.P. The DEVS formalism: Event-based control for intelligent systems. *Proceedings of IEEE* 77 (1989) No. 1: 27-80.

Zeigler, B.P., and J. Kim. Extending the DEVS-scheme knowledge-based simulation environment for real-time event-based control. Submitted to *IEEE Transactions on Robotics and Automation*.