69

Conf-9104287--2

Presented at the EPRI 1991 International Conference on Power Plant and Power System Training, Simulators, and Modeling, April 15-17, 1991, Miami Beach, FL.

REAL-TIME DISTRIBUTED SIMULATION USING THE MODULAR MODELING SYSTEM INTERFACED TO A BAILEY NETWORK 90 SYSTEM

R.M. Edwards
J.A. Turso
Nuclear Engineering Department
The Pennsylvania State University
231 Sackett
University Park, Pennsylvania 16802

H.E. Garcia
M.H. Ghie
Electrical and Computer Engineering Department
The Pennsylvania State University
104A Electrical Engineering East
University Park, Pennsylvania 16802

S. Dharap
Computer Science Department
The Pennsylvania State University
University Park, Pennsylvania 16802

S. Lee
Mechanical Engineering Department
207 Mechanical Engineering Building
The Pennsylvania State University
University Park, Pennsylvania 16802

CONF-9104287--2

DE93 003196

FG02-89ER12889

0- 70

REAL-TIME DISTRIBUTED SIMULATION USING THE MODULAR MODELING SYSTEM INTERFACED TO A BAILEY NETWORK 90 SYSTEM

ABSTRACT

The Modular Modeling System (B&W MMS) (1) was adapted for real-time simulation testing of diagnostic expert systems in 1987.(2,3) The early approach utilized an available general purpose mainframe computer which operated the simulation and diagnostic program in the multitasking environment of the mainframe. That research program was subsequently expanded to intelligent distributed control applications incorporating microprocessor based controllers with the aid of an equipment grant from the National Science Foundation (NSF).(4) The Bailey NETWORK 90 microprocessor-based control system, acquired with the NSF grant, has been operational since April of 1990 and has been interfaced to both VAX mainframe and PC simulations of power plant processes in order to test and demonstrate advanced control and diagnostic concepts. This paper discusses the variety of techniques that have been used and which are under development to interface simulations and other distributed control functions to the Penn State Bailey system.

INTRODUCTION

The NSF supplied Bailey NETWORK 90 system currently supports a Department of Energy (DOE) funded Penn State project that is developing a demonstration of Intelligent Distributed Control for the Experimental Breeder Reactor (EBR-II).(5) The liquid sodium cooled EBR-II is located at the Idaho National Engineering Laboratory and is operated as an experimental facility by the Argonne National Laboratory. It is the United States' main breeder reactor technology demonstration program. The most important demonstration currently being pursued is on-site fuel reprocessing and actinide burning based on a metal alloy fuel cycle. However, other technologies related to future nuclear power plant applications, such as advanced controls and diagnostics, are also being pursued on an experimental basis and permanently incorporated in the plant when they have been proven beneficial, and needed, to maintain the safe and efficient operation of the facility.

The focus of the initial Penn State research involving EBR-II is the steam plant. The EBR-II steam plant is a small but complete steam electric power cycle that produces 20 MWe and includes a natural circulation steam drum boiler system, superheaters, turbine generator, feedwater system and condensate system. The evaporators and superheaters are heated with liquid sodium. Several of the original (1960s vintage) analog controllers have already been replaced by digital equivalents implemented in a Bailey NETWORK 90 system.(6) The EBR-II plant thus already contains some of the essential architecture that can be expanded as necessary to conduct advanced diagnostics and controls demonstrations. Our approach to developing demonstrations for the EBR-II steam plant first developed batch mode MMS simulations of the steam plant.(7) The current effort is interfacing the MMS simulations to the Penn State Bailey system in order to perform real-time testing of advanced control and diagnostic concepts.

The Penn State College of Engineering DECNET communication system makes available a large variety of existing computer systems that can be used to conduct distributed simulation of power plant processes, advanced controls and diagnostics. The computers available on the network include a VAX 8550 mainframe, 11/780, 11/785, and MICROVAX workstations under the VMS operating system. Most of the College's PC computers and UNIX workstations are also being connected to the DECNET system. Computers operating under UNIX have also been interfaced to simulations operating in the VAX/VMS environment using standard TCP/IP communication protocols. The DECNET system is further interfaced to a university communication network that contains an IBM 3090/600 mainframe and gateways to other computer networks.

An anticipated advanced controls room upgrade project at EBR-II is envisioned to be centered on a network of UNIX computers and workstations interfaced to their plant data acquisition and control equipment.(8) Both serial and IEEE 488 parallel interfaces between UNIX computers and the Penn State Bailey System are therefore being explored to effect hierarchical distributed diagnostics and control as an intelligent control concept for testing via large scale distributed simulation.

THE PENN STATE BAILEY NETWORK 90 SYSTEM

The Penn State Bailey system is configured in two cabinets as shown in the block diagram of Figure 1. An Intel 386 based PC Engineering Workstation (EWS) operates the Bailey CAD software for programming controllers and Bailey Process Control View software for graphics based process monitoring. A multifunction controller is housed within process control unit (PCU) #1 of the standard cabinet and an enhanced controller is contained in PCU #5 of the smaller MINI 90 enclosure. Three computer interface modules (CIUs) are individually contained in PCUs #2, #3, and #4 contained in the standard cabinet. The PCUs are interfaced on a Bailey INFINET 90 super loop communication network. The CIUs provide the interface between the Bailey microprocessor based controllers and power plant simulations and higher level diagnostic and control functions executed in other computers. Each CIU provides 2 serial interfaces and capability for an optional IEEE 488 interface.

VAX SIMULATION INTERFACED TO THE BAILEY SYSTEM

The multifunction controller provides general purpose C language programming capability that has been interfaced with standard control blocks to demonstrate a reconfigurable control strategy based on learning theory.(9,10) In this first application involving the Bailey system, a real-time simulation of the EBR-II condensate system (Figure 2a) operates in the VAX 11/780. The deaerating heater pressure and level and several other process variables are transferred to the Bailey system through a serial line from the VAX 11/780 to a CIU which simply stores the data in a point table. The multifunction controller is programmed with Bailey standard PI control blocks similar to those in actual use at EBR-II except that the analog inputs (pressure, level, etc) are obtained from the CIU point table instead of from actual field termination units. Similarly, the outputs of the Bailey PI control algorithms (condensate and steam supply valve position commands) are also stored in the CIU point table where they are read by the VAX simulation.

The learning systems reconfigurable control strategy (9,10) is programmed in the multifunction controller using the C language and interfaced to standard control block operations in the multifunction controller using a Bailey supplied C utilities library. The objective of this particular application was to evaluate the performance of learning algorithms in process control. This reconfigurable control strategy evaluates the performance of the condensate system based on the process variables coming into the controller and learns to take control actions that improve performance. In this initial demonstration, there was only a choice of two possible actions: 1) select the standard PI control algorithm for level regulation using the condensate valve or 2) select an alternate PI control algorithm for pressure regulation using the condensate valve. By choosing the proper control action, the reconfigurable controller mitigates the consequences of failures in the deaerator steam supply. Other approaches to diagnosing and mitigating the consequences of system case of the EBR-II condensate system simulation, the MMS model consists of 1 pump, 1 closed feedwater heater, 1 deaerating heater, 1 cooler, 5 pipes, 3 valves and associated connectors representing a total of 19 differential equations.

The connection of these subsystems to one another and eventually the rest of a simulated plant takes place at various boundaries characterized by their mass flow rate, pressure and enthalpy of the fluid. The MMS protocol to achieve modularity defines a resistance boundary as one that takes pressure as an input variable and calculates flow rate and a capacitance boundary as one that takes flow rate as an input variable and calculates the pressure. The downstream boundaries of the EBR-II high pressure feedwater train simulation (the inlet of the feedwater pump, the outlet of the number 3 heater drain, and the outlet of the feedwater pump recirculation line) are setup as resistance boundaries. The complementary boundaries at the upstream end of the condensate system simulation are setup as capacitance boundaries. This setup means that the high pressure feedwater train simulation calculates the flow rates at these boundaries while the condensate system simulation calculates the pressure at these boundaries. Each simulation needs the variables calculated by the other simulation at the boundaries in order to calculate the response of the combined system.

The boundary variables (pressures, enthalpies and flow rates) and additional process variables for display and monitoring purposes are placed in or removed from a DEC VMS global data section at a specified sampling interval (e.g., every 2 seconds) by the MMS FORTRAN based simulation as diagrammed in Figure 3. The global section of the MMS simulation is made directly accessible to two stand-alone communication programs called a receive client and a send client operating in the same computer as the simulation. Data received from other processes outside the simulation are placed in the global section, data to be sent to other processes is read from the global section and sent to other computers using the DECNET system and TCP/IP communication protocol. Simulations executing faster than real-time call appropriate system subroutines to delay themselves as necessary to match a real-time clock at each sampling interval.

Analytical Evaluation of Distributed Simulation Results

The primary objective of conducting a distributed simulation is to achieve better real-time performance. However, distributed simulation, as described above, introduces an additional approximation which can potentially cause significant errors and even numerical instability. In a combined simulation containing both subsystems, the boundary variables are appropriately interchanged more frequently at the integration step size of the numerical integration algorithm. In the distributed simulation, the boundary variables are interchanged at the somewhat longer period of the sampling interval. This sampling is an additional approximation to the dynamics of the actual process. In our development of distributed simulation to meet real-time execution needs, we are carefully examining the results of the distributed simulation against a combined simulation of the same subsystems. With the MMS, combining subsystem simulations into a single simulation is a relatively straightforward process due to the inherent modularity of the MMS concept.

A system fault used to benchmark the analytical results of the distributed simulation evaluates the failure of the feedwater pump automatic recirculation valve from the normally closed position to the fully open position (Figure 2b).(5) The EBR-II feedwater pump recirculation valve automatically opens when the flow through the feedwater pump falls below a threshold value. This system fault, which is very near the boundary of the two distributed simulations, is also considered to be a worst case for assessing the analytical capability of the distributed simulations. Figure 4 shows the response of the feedwater flow, recirculation line flow, feedwater pump outlet pressure and deaerator pressure using the distributed simulations. The results from the combined simulation are within the plotting accuracy of Figure 4 (3 significant digits). To assess the gain in real-time simulation capability achieved by the distributed simulations the total CPU time to calculate all the results for the 360 second transient is compared to the CPU time of the combined simulation. The distributed high pressure feedwater train simulation used 15 CPU seconds and the condensate system simulation took 10 CPU seconds on a VAX 8550. The CPU time of the combined simulation was equal to the sum of the two distributed simulations, 25 seconds. When operated in parallel on separate dedicated computer systems, the distributed simulation thus provides a 40% improvement in real-time performance over the single combined simulation for this particular simulated event.

UNIX Computers Interfaced To Distributed Simulations

Since the EBR-II advanced control room project (8) is envisioned to utilize a network of real-time UNIX workstations, one of the destinations of simulation results and sources of control actions for the distributed VAX simulations that has been developed is to UNIX based workstations. Various interactive operator interfaces to monitor simulated plant process data and to direct control actions (controller setpoints, gains, etc) in the simulation are under development using X Windows and VI Corporation DATAVIEWS graphical data base software.(19) In the UNIX environment the rough equivalent of DEC VMS global sections is shared memory which can be similarly made accessible to many multi-tasking display, diagnostics, and higher level automated decision making and control strategies.

As shown in Figure 3, the UNIX computer manages the distributed simulation by collecting and distributing simulated plant data as requested by the send and receive clients of the individual subsystem simulations. A simulated plant data base is maintained in shared memory by the server concept of UNIX network programming and is accessible to other processes operating in the UNIX computer. An interactive simulation operator interface permits control actions to be directed to the distributed simulation by changing values of assigned variables in the centralized data base. Diagnostic and higher level automated control functions can also be operated as additional processes in the UNIX computer and effect changes in the simulated plant by changing data in the shared memory.

The serial interface of the VAX computer directly to the Bailey in our early work was undertaken for expediency due to the availability of a serial line and Bailey supplied FORTRAN libraries. For higher volume and more reliable data transfer requirements anticipated for larger scale simulations and actual plant diagnostics and control requirements, an IEEE 488 interface installed in a Concurrent Computer 6350 real-time UNIX workstation is being developed.(4) In this configuration, the 6350 computer will participate in the DECNET communications of simulated plant data and control actions and act as a bridge between the DECNET and the Bailey CIU.

PC COMPUTER INTERFACED TO THE BAILEY

The Penn State Bailey Engineering Workstation includes an Intel 386 based PC computer which has also been used to implement a real-time simulation of the EBR-II condensate system, again using the MMS. In this exploratory application, the actual EBR-II deaerator level and pressure control algorithms were programmed in an enhanced controller module. The FORTRAN based MMS PC simulation was linked to the Bailey CIU using a Bailey supplied C Language Library for PC operations.

The speed of the 386 based simulation is comparable to the VAX 11/780 system capability when the user load on the VAX is reasonable. Since the VAX computers are general purpose computers servicing a variety of users simultaneously, a dedicated 386 PC can sometimes out perform the real-time performance of a VAX mainframe simulation. It appears reasonable to consider meeting large scale MMS real-time simulation needs using a network of PCs interfaced to the Bailey System and this approach may be pursued in the future as needed and as additional PCs and simulation software become available to us.

SUMMARY

Real-time simulation testing of advanced control and diagnostic concepts has been and will continue to be a key element of the Penn State research program for improving power plant operations. The current research activities incorporate a Bailey NETWORK 90 microprocessor based control system that has been interfaced to both VAX mainframe and PC based MMS simulations. UNIX workstations have also been interfaced to VAX distributed simulations in order to provide an interactive operator interface to the simulation, advanced diagnostics, and control functions.

ACKNOWLEDGEMENTS

The development of a demonstration of Intelligent Distributed Control for the EBR-II steam plant is being conducted as part of a DOE University Grant DEFG07-89ER12889, Modification No. A000. The Penn State Bailey NETWORK 90 system was acquired as an NSF grant, ECS-8905917. However, any findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of DOE or NSF.

REFERENCES:

1) Modular Modeling System (MMS): A Code for Dynamic Simulation of Fossil and Nuclear Power Plants. Overview and General Theory. CS/NP-2989, Palo Alto, California, Electric Power Research Institute, March 1983.

2) R.M. Edwards and G.E. Robinson, "Real-Time Modeling and Controls Analysis Using the IBM Advanced Control System", EPRI 1988 Conference on Power Plant Training Simulators and Modeling, Charlotte, North Carolina, June 15-17, 1988.

3) R.M. Edwards, J.J. Carper, R. Lindsay, and G.E. Robinson, "Real-Time Testing of a Diagnostic and Control Guidance Expert System," The Seventh Power Plant Dynamics, Control and Testing Symposium, Vol 2, May 15th to 17th 1989, Knoxville Tennessee, pp 67.01-67.09.

4) E.S. Kenney, "FINAL TECHNICAL REPORT: Engineering Research Equipment Grant - Microprocessor Based Controllers", NSF Grant ECS-8905917, The Pennsylvania State University, January 18, 1991.

5) R.M. Edwards, D.W. Ruhl, E.H. Klevans, and G.E. Robinson, "Development and Testing of a Diagnostic System for Intelligent Distributed Control at EBR-II," Proceedings of The 1990 International Fast Reactor Safety Meeting, Vol. III, August 12-16, 1990, Snowbird Utah, pp 299-308.

6) N.C. Messick and M.P. Lucas, "Application of Microprocessor Based Controllers in the Breeder Reactor Program," Proceedings of the 1985 Seminar on Power Plant Digital Control and Fault-Tolerant Microcomputers, NP-4769-SR, Palo Alto California, Electric Power Research Institute, September 1986, pp 4.61-4.74.

7) D.W. Ruhl, "Modeling the Experimental Breeder Reactor-II Steam Plant Using the Modular Modeling System," Nuclear Engineering Master of Science Thesis, The Pennsylvania State University, August 1989.

8) L.J. Christensen, J.I. Sackett, Y. Dayal and W.K. Wagner, "Plant Automation: EBR-II Experience and Future Plans with GE ALMR," The Seventh Power Plant Dynamics, Control and Testing Symposium, Vol 1, May 15th to 17th 1989, Knoxville Tennessee, pp 37.01-37.05.

9) H.E. Garcia, A. Ray, and R.M. Edwards, "Reconfigurable Control of Power Plants Using Learning Automata," IEEE Control Systems Magazine, Vol. 11, No.1, January 1991, pp. 85-92.

10) H.E. Garcia, Asok Ray, and R.M. Edwards, "A Reconfigurable Strategy for Distributed Digital Process Control," Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems, August 15-17, 1990, Yale University, Connecticut, pp 184-188.

11) R.M. Edwards, A. Ben-Abdennour, and K.Y. Lee, "Robust Control Design Verification Using the Modular Modeling System," this conference.

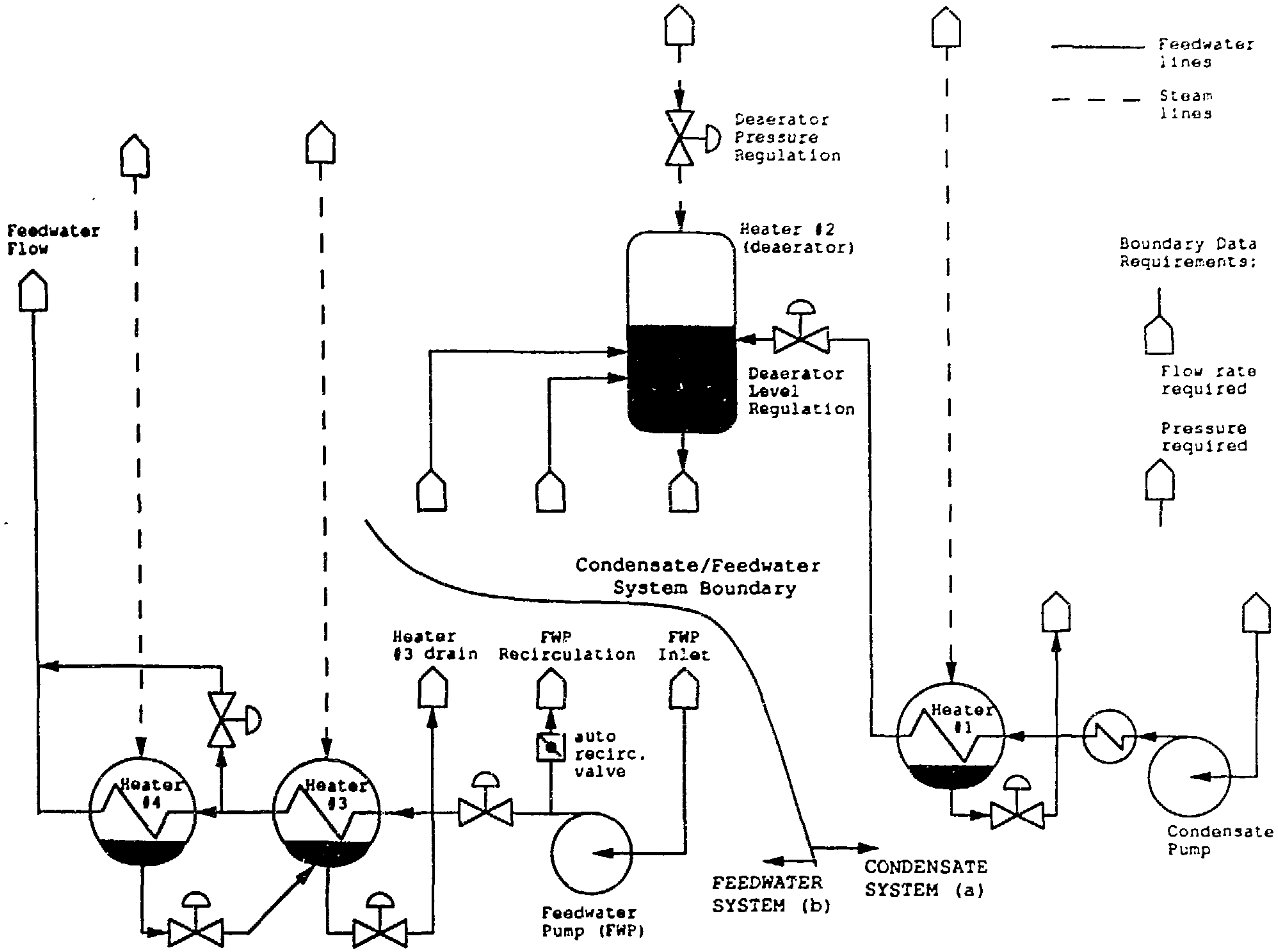

Figure 2. Modeling of the EBR-II Condensate System (a) and High Pressure Feedwater Train (b) for Distributed Simulation.

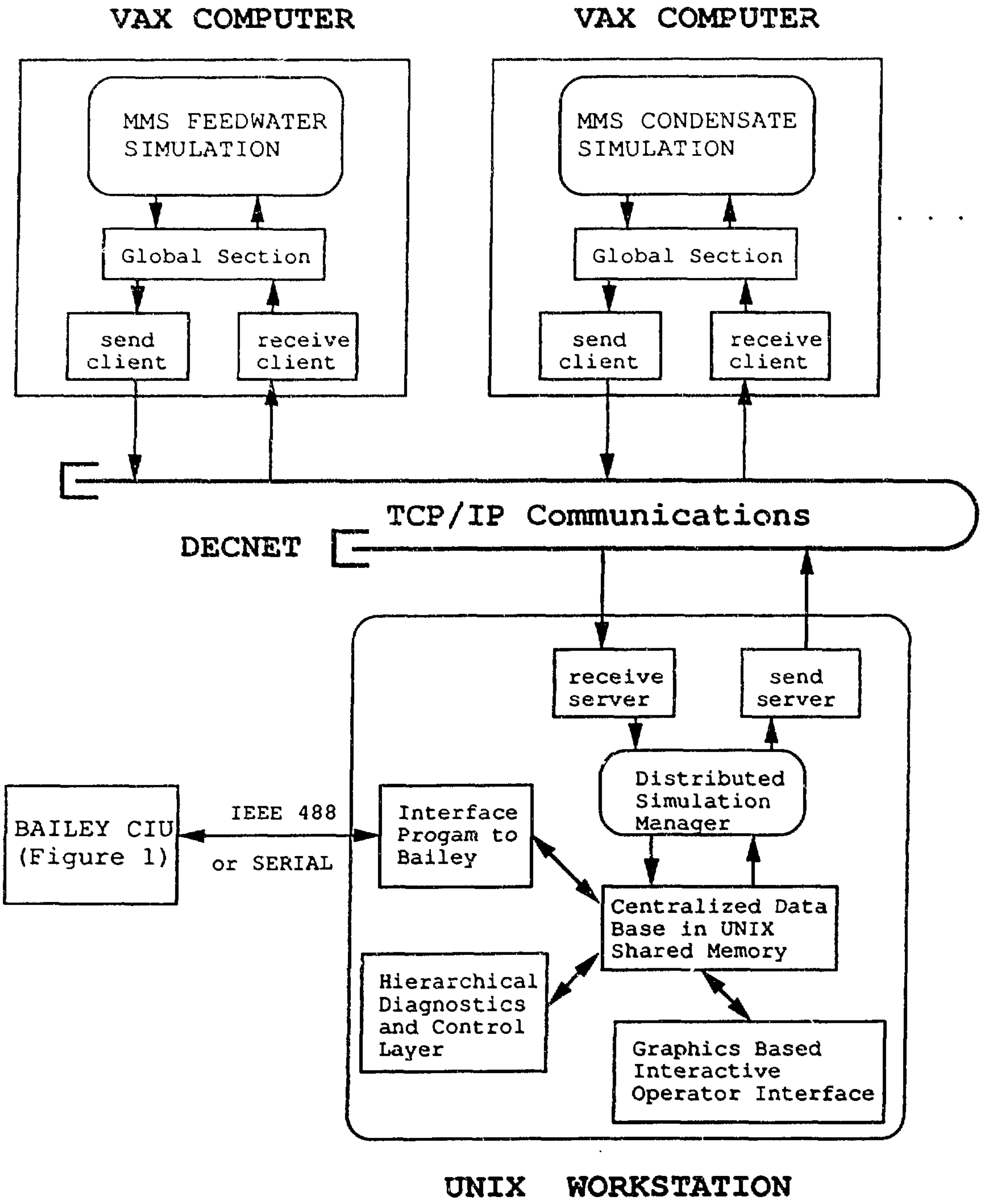

Figure 3. Block Diagram of Distributed Simulation Architecture Using VAX, UNIX and Bailey Systems.

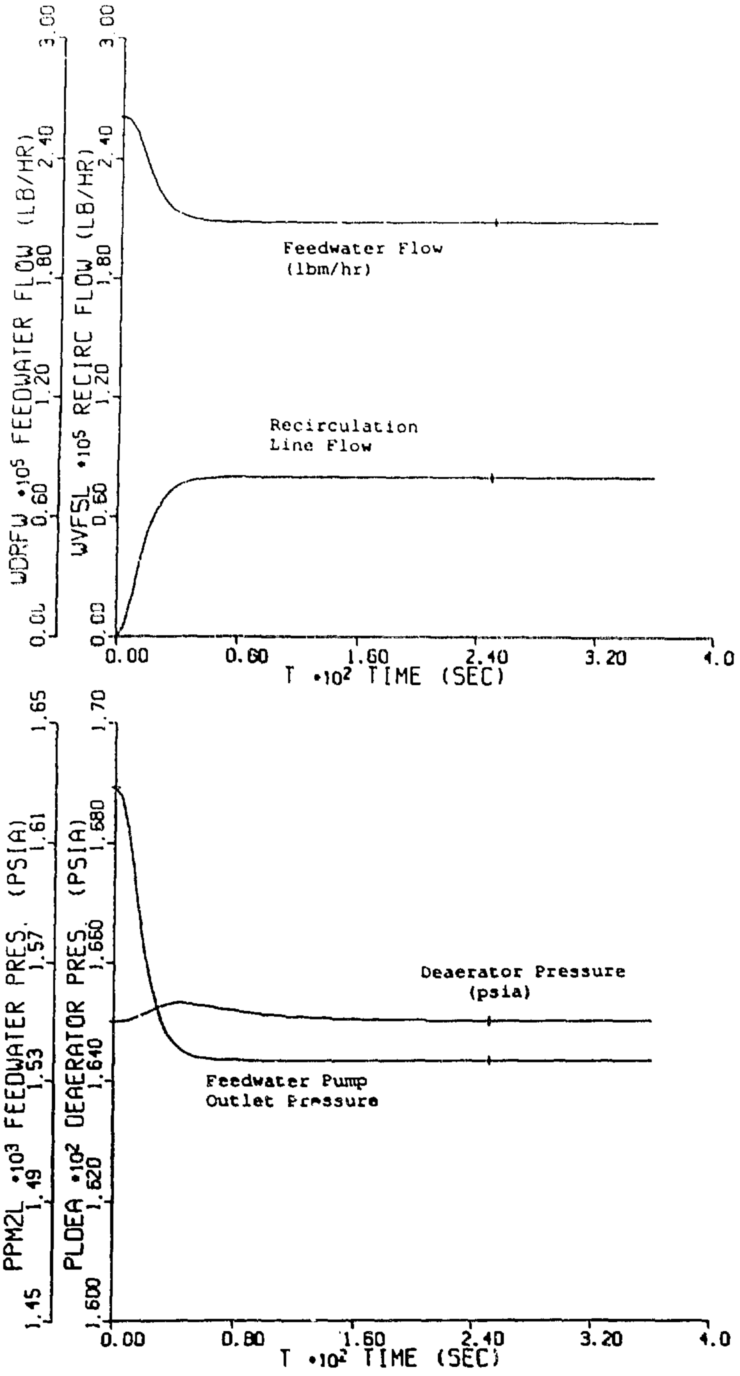

Figure 4. Distributed Simulation Results for a Failure of the EBR-II Feedwater Pump Recirculation Valve to the Fully Open Position.