Distributed Simulation of Complex Systems

Final Report for ARO Grant DAAH04-96-1-0285

Technical Report of the ISIS Group
at the University of Notre Dame
ISIS-97-007
October , 1997

M.D. Lemmon and P.J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Interdisciplinary Studies of Intelligent Systems

Distributed Simulation of Complex Systems

Michael D. Lemmon
Dept. of Electrical Eng.
University of Notre Dame
Notre Dame, IN 46556, USA
(219)-631-8309
Fax:(219)-631-4393
lemmon@maddog.ee.nd.edu

Panos J. Antsaklis
Dept. of Electrical Eng.
University of Notre Dame
Notre Dame, IN 46556, USA
(219)-631-5792
Fax:(219)-631-4393
antsakli@saturn.ee.nd.edu

ARO GRANT: DAAH04-96-1-0285
Final Technical Report: October 14, 1997

Summary

During the period from June 1996 to July 1997, the department of electrical engineering at the University of Notre Dame developed a facility for the simulation of complex dynamical systems. This simulation facility will be used to support on-going basic research sponsored by the Army Research Office concerned with hybrid dynamical systems and distributed command and control networks. The development of this lab facility was funded by the Army Research Office's Defense University Research Instrumentation Program (ARO-DURIP) under grant DAAH04-96-0285. Grant funds were used to equip and setup the lab facility. The facility consists of a computer cluster networked together under a real-time POSIX.1b compliant operating system [1]. The cluster consists of SUN workstations serving as supervisors to a group of networked micro-controllers connected to servo systems. The purpose of the lab is to provide a physical testbed in which to implement and test recently developed methods for the analysis, verification, and synthesis of distributed real-time control systems [2] [3]. In addition to supporting hybrid systems research, the lab facility is being used as an instructional lab for a senior level course in digital and real-time control. The following technical report discusses the scope and purpose of the lab facility, describes the facility in some detail, and discusses the research and teaching projects that have been supported by the lab.

Research Objectives and Motivation

Complex systems such as vision-based robotics , flexible manufacturing systems, and distributed battle management systems generally consist of networked computers concerned with the control and supervision of the overall process. These systems are expected to adapt to unexpected changes in their operating environment while maintaining a specified level of performance. Such systems are often referred to as **hybrid control systems** [3] since they combine event-driven and time-driven components. Due to the complexity of such systems, robust performance can rarely be assessed solely through the formal analysis of idealized mathematical models. Detailed hardware simulations are often required to verify theoretical expectations against the actual system implementation. The purpose of this equipment grant was to develop a simulation facility which would be readily accessible to academicians interested in the analysis and synthesis of hybrid systems.

The complex systems simulation facility (CSSF) consists of a network of workstations and micro-controllers. As originally conceived the lab was designed to satisfy a variety of needs associated with the hardware simulation of complex systems. These needs are:

- high speed computing to provide real-time simulation of complex processes,
- facilities for the coordination and synchronization of distributed dynamical processes,
- a simulation capability that can readiliy interface to application specific hardware,
- real-time operating environment (operating system) to allow on-line process montioring, interaction with real-time hardware systems, and the study of man-machine interfaces,
- a rich software development environment to allow rapid prototyping of different applications,
- and the flexibility to do real-time hardware simulations for a variety of application testbeds.

The proposed facility meets these needs by using a distributed network of computer workstations to perform hardware/software simulation tasks. The computers are networked through a dedicated local area network using TCP/IP protocol [4]. Workstations run a POSIX.1b compliant operating system (Solaris) to give programmers access to fine-grained timing utilities. All of the basic components of the facility are made up of off-the-shelf products conforming to international standards in order to minimize the amount of custom software that had to be developed.

Facility Description

The ARO-DURIP lab facility consists of a cluster of 9 computers over a 10baseT ethernet. Five of these computers are SUN Ultra 1 workstations running Solaris 2.5 with its real-time extensions enabled. Three of these workstations serve as *supervisors* for a cluster of 4 dSpace miniboxes. The other two workstations act as disk servers for the entire network. The DSpace miniboxes are network aware 386 PC's with a DS1102 microcontroller [5] card using the TMS320 chip [6] . The miniboxes act as local agents in the network responsible for controlling analog hardware representing a complex physical plant. The miniboxes are connected to one of two types of analog systems. There is a 12-state analog computer (ComDyna Inc.) which provides a flexible way of simulating complex systems with large state spaces. The miniboxes can also be interfaced to servo-motor assemblies (Quanser Inc.) that provide a modular way of modeling a modest number of interacting physical processes as would be found in flexible manufacturing facilities. These facilities permit the lab to perform hardware simulations in which various plant controllers must coordinate their actions by communicating over a local area network.

An important part of the lab is its software environment. Lab workstations use a POSIX.1b compliant operating system (Solaris 2.5) with its real-time extensions enabled. The graphical user interface (GUI) for these programs was built using TCL/TK scripts [7]. Networking software included traditional network programming methods such as sockets [4] as well as more recent message passing protocols such as MPI [8]. The dSpace miniboxes are networked with the SUN workstations using dSpace propriety software. The dSpace software was particularly useful because it included real-time data logging utilities [10] as well as an adanced programming interface (API) to such higher level languages as "C" [9] and MatLab.

Figure 1 illustrates the actual layout of the lab facility (note that disk servers are not shown). In this diagram, we see that the supervisory workstations and dSpace miniboxes are connected over a 10baseT ethernet. This local network is then connected to the main campus network using a LAN bridge. One end of the dSpace miniboxes is connected to this local network. The other end is connected to the Quanser servo systems or the ComDyna analog computer. The servo systems can also be interconnected by direct physical coupling of the motors.

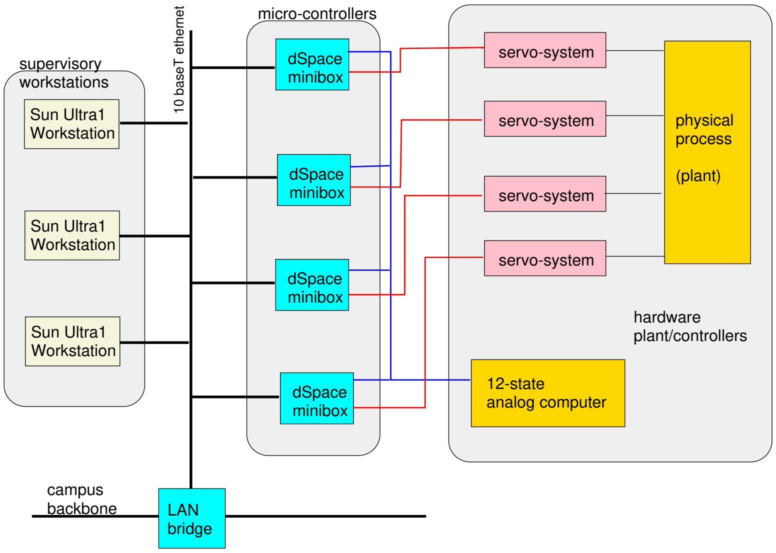

Figure 1: ARO-DURIP Lab Layout

Significant Accomplishments

This section lists the most important tasks which were accomplished under this grant.

1. **Selection and purchasing of lab equipment:** The selection and ordering of lab equipment was completed within the first 6 months of the grant period.

2. **Equipment Installation:** Lab equipment was delivered within 6 months after equipment was ordered. Once delivered, the computers and networking equipment were installed. Installation was completed by June 1997.

3. **Networking Software Development:** An important part of the lab's infrastructure is the network software environment. The lab uses a POSIX.1b compliant operating system (Solaris 2.5) as well as the message passing interface (MPI) protocol which represents a community standard in networking software for parallel computation. The lab also uses dSpace proprietary daemons to control communication between dSpace miniboxes and SUN worstations. This software was installed and tested by February 1997.

4. **Demonstration Development:** As an initial demonstration of the lab's capabilities, two projects were proposed and completed. The first project concerned the development of a senior-level teaching laboratory for digital and real-time control. The second project concerned the development of control protocols in a distributed computing environment ensuring that two dynamical servo systems do not violate a mutual exclusion constraint. The first project

with the teaching lab was completed in May of 1997. The second project was completed in July of 1997. The technical reports generated by these projects will be found in the appendix. A brief summary of both projects is provided below.

(a) **Senior level teach lab:** The ARO-DURIP lab provided a very flexible facility for the hands-on teaching of real-time control principles. Also, using the lab for teaching purposes allowed us to support its maintenance using departmental resources, thereby guaranteeing the long life of this lab. The development of this lab also provided lab staff with sufficient hands-on experience with the lab's networking software. The objective of this lab was to provide students with a simple way of downloading "C" programs implementing a deadbeat controller. The interface between student and computer was developed using a TCL/TK script which made the appropriate system calls to the dSpace daemons. A complete description of the lab exercise will be found in [11] located in the appendix.

(b) **Mutual Exclusion Project:** An important problem in real-time control concerns the guarantee that mutual exclusion constraints are respected by the networked controllers. The development of a communication protocol which ensures mutual exclusion can be done in a variety of ways. This project chose a standard (though slow) method in which a semaphore created by one of the supervisory workstations was shared by all processes on the network. The project provided laboratory staff with hands-on experience in UNIX systems and network programming. Details of the project will be found in the appendix. Future work intends to use the lab to study the use of "fast timing based algorithms" [12] and to see if recently developed verification tools from the hybrid systems [2] community can be used to guarantee system safety. The results of this task will be found in [13] and [14] which are both located in the appendix.

5. **Lab Documentation:** An important part of this lab effort involved the on-line documentation of lab resources. The development of such documentation makes it easier for new users of the lab to acquire the necessary skills in using lab facilities. This task wrote and made available over the world-wide web several pages describing the laboratory's goals and ongoing projects [11]. We also developed an on-line course on advanced Unix programming to acquaint new users with the system level programming skills needed to use the laboratory [15]. Some of the selected webpages are found in the attached appendix. This task was completed by August 1997.

List of Publications and Technical Reports

The following technical reports and webpages were generated by this grant.

1. Complex Systems Simulation Facility Homepage, Michael Lemmon, URL: http://www.nd.edu/~lemmon/aro-durip/, August 1997

2. dSpace Software Environment in the ARO-DURIP Facility, Technical Report of the ISIS Group at the University of Notre Dame, ISIS-97-003, March 1997.

3. Mutual Exclusion and Critical Sections in Robotic Manufacturing Systems: an example, Technical Report of the ISIS Group at the University of Notre Dame, ISIS-97-005, October 1997.

4. Advanced UNIX Programming: lecture notes, Michael Lemmon, Technical Report of the ISIS Group at the University of Notre Dame, ISIS-97-006, August 1997.

List of Scientific Personal

The following personnel were supported by this grant:

1. Dr. Michael D. Lemmon, associate professor of electrical engineering, University of Notre Dame
2. Dr. Panos J. Antsakls, full professor of electrical engineering, University of Notre Dame
3. Mr. Christopher J. Bett, doctoral student, electrical engineering, University of Notre Dame, est. grad. date: May 1998.
4. Mr. John Moody, doctoral student, electrical engineering, University of Notre Dame, est. grad. date: Dec. 1997.
5. Mr. Xuping Xu, doctoral student, electrical engineering, University of Notre Dame, est. grad. date TBD.
6. Mr. Kevin He, doctoral student, electrical engineering, University of Notre Dame, est. grad. date TBD.
7. Mr. Xenophon Koutsoukos, doctoral student, electrical engineering, Univeristy of Notre Dame, est. grad. date TBD.

Report of Inventions

There were no inventions generated by this project.

References

[1] B. Gallmeister, Programing for the Real World: POSIX.4, O'Reilly and Associates Inc., 1995.

[2] R. Alur, T. Henzinger, and P-H Ho, Automatic Symbolic verification of embedded systems, *IEEE Trans. on Software Engineering*, 22:181-201, 1996.

[3] Hybrid Systems, editors: Grossman, Nerode, Ravn, and Rischel, 1993, Springer-Verlag, lecture ntoes in computer science 736.

[4] W.. Stevens, UNIX Network Programming, Prentice-Hall 1990.

[5] DS1102 Software Environment Reference Guide, dSpace, Paderborn Germany 1994.

[6] TMS320 Floating point DSP code generation tools: getting started guide, Texas Instruments, 1996.

[7] J.K. Ousterhout, Tcl and the Tk Tookit, Addison Wesley, 1994.

[8] W. Gropp, E. Lusk, and A. Skjellum, Using MPI: portable parallel programming with the message passing interface, MIT Press, 1994.

[9] Host-DSP Interface Library CLIB User's Guide, dSpace, Paderborn Germany, 1994.

[10] Real-time Trace Module User's/Reference Guide, dSpace, Paderborn Germany, 1994.

[11] Michael Lemmon, Complex Systems Simulation Facility Homepage,
URL: http://www.nd.edu/~lemmon/aro-durip/, August 1997

[12] R. Alur and G. Tabenfeld, Fast timing based algorithms, distributed computing, 10 (1996), pp. 1-10.

[13] dSpace Software Environment in the ARO-DURIP Facility, Technical Report of the ISIS Group at the University of Notre Dame, ISIS-97-003, March 1997.

[14] Mutual Exclusion and Critical Sections in Robotic Manufacturing Systems: an example, Technical Report of the ISIS Group at the University of Notre Dame, ISIS-97-005, October 1997.

[15] Advanced UNIX Programming: lecture notes, Michael Lemmon, Technical Report of the ISIS Group at the University of Notre Dame, ISIS-97-006, August 1997.