CONF-9007134-1

DE90 014458

A HARDWARE/SOFTWARE ENVIRONMENT TO SUPPORT R&D IN INTELLIGENT MACHINES AND MOBILE ROBOTIC SYSTEMS*

Reinhold C. Mann
Center for Engineering Systems Advanced Research
Engineering Physics and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6364

To be presented at the DARPA Workshop on Distributed Intelligent Control Systems, July 17–19, 1990, Pacifica, CA.

*Operated by Martin Marietta Energy Systems, Inc., under contract DE-AC05-840R21400, U.S. Department of Energy.

A Hardware/Software Environment to Support R&D in Intelligent Machines and Mobile Robotic Systems*

Reinhold C. Mann
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
P.O.Box 2008
Oak Ridge, TN 37831-6364
mnn@ornl.gov

Abstract

The Center for Engineering Systems Advanced Research (CESAR) serves as a focal point at the Oak Ridge National Laboratory (ORNL) for basic and applied research in intelligent machines. R&D at CESAR addresses issues related to autonomous systems, unstructured (i.e. incompletely known) operational environments, and multiple performing agents. Two mobile robot prototypes (HERMIES-IIB and HERMIES-III) are being used to test new developments in several robot component technologies.

This paper briefly introduces the computing environment at CESAR which includes three hypercube concurrent computers (two on-board the mobile robots), a graphics workstation, VAX, and multiple VME-based systems (several on-board the mobile robots). The current software environment at CESAR is intended to satisfy several goals, e.g.: code portability, re-usability in different experimental scenarios, modularity, concurrent computer hardware transparent to applications programmer, future support for multiple mobile robots, support human-machine interface modules, and support for integration of software from other, geographically disparate laboratories with different hardware set-ups.

Introduction

CESAR at ORNL focuses its research on the development and experimental validation of intelligent control techniques for autonomous mobile robots able to plan and perform a variety of tasks in unstructured environments. The purpose of this paper is to provide a brief description of the hardware and software environment at CESAR which has been evolving in order to support research in several robot component technologies. The material presented in this paper is excerpted from several reports and articles published previously by CESAR staff. Selected references are given in this paper. A full CESAR publication list if available from the author upon request.

Assignments for the robot(s) originate with the human supervisors in a remote control station, and the robot then performs detailed implementation

---

* Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences and by the Office of Nuclear Energy of the Department of Energy, under contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

planning and executes the tasks. Since the operational environment is generally dynamic, the robot must be in sensory contact with its surroundings to capture and recognize changes which bear on its task objectives and , if necessary, replan its behavior. These capabilities imply that the robot has cognitive capabilities that enable it to form and modify a model of the world around it and relate this world model to the task objectives. Research is also conducted to enable the robot to learn from its past experience, and thus improve its performance.

CESAR's principle current objectives are: (a) to achieve a level of technology that enables the autonomous performance of classes of navigation and manipulation tasks of human scale in a spatially complex environment; (b) to use these performance tasks to focus research objectives. Application drivers for this basic research effort include, among others, robotics for advanced nuclear power stations, and environmental restoration and waste management activities.

CESAR is developing a series of mobile autonomous robot vehicles named HERMIES (Hostile Environment Robotic Machine Intelligence Experiment Series) as experimental testbeds for validation and demonstration of research results. The newest research robot, HERMIES-III, includes the functional capabilities that permit research in combined mobility/manipulation, and allows us to experiment with cooperative control of multiple robots having different capabilities.

Hardware Environment

HERMIES-IIB and HERMIES-III are the currently operational mobile robots at ORNL/CESAR. HERMIES-IIB stands 1m high and weighs 91kg. Rechargeable batteries supply 20W of power providing about 20 minutes of untethered running time. Peak movement speed is 0.7m/s. Sensors include four Sony CCD cameras and a number of Polaroid sonar transceivers mounted on a rotatable turret. The computer architecture consists of a VME rack housing a Motorola 68020 CPU and a variety of I/O boards interfaced via a BIT-3 communication link to an NCUBE (NCUBE, Inc., Beaverton, OR) hypercube computer. The hypercube consists of 16 nodes with 512 Kbytes RAM each and an Intel 80286 I/O processor, which also serves as host for the hypercube. Each node processor is a 32 bit microcomputer with on-chip floating point and communications hardware. This gives HERMIES-IIB roughly 16 MIPS in the on-board hypercube. HERMIES-IIB is equipped with two Zenith/Heathkit five degress-of-freedom arms which give the robot extremely limited manipulative capability. This has not been a drawback, however, since the robot was not intended for research in manipulation.

The HERMIES-III mobile robot consists of an omni-directional wheel-driven chassis, a seven degrees-of-freedom manipulator (CESARm), an Odetics laser range camera, multiple CCD cameras (two stereo pan and tilt mechanisms), an array of sonar transceivers, and an on-board computer system that includes five Motorola 68020 CPUs in four VME racks, and an NCUBE hypercube concurrent computer. CESARm is a compliant, high capacity-to-weight ratio (~ 1/10) robot manipulator, with an adjustable gripper, which is equipped with a JR^3 force-torque sensor, and a LORD tactile sensor pad.

Both robots can be operated completely autonomously, in which case they can communicate via RS-232 wireless modems to an off-board computer. They can also be interfaced through ethernet to a local are network of computers, as schematically shown in Figure 1. This network includes a Silicon Graphics IRIS 4D workstation, a microvax, and an NCUBE hypercube computer with 64 processors.

Software Environment

The computer programs that control HERMIES-IIB's behavior are mostly written in C and can be organized into four classes: the HERMIES primitives (i.e., functions that directly control platform motion, activate sensors, etc.), the expert system and associated routines for navigation and multi-sensor integration (error propagation and conflict resolution), the image analysis routines (a complete library that makes the concurrent hypercube hardware transparent), and the control and integration routines. An expert system may be executed from the NCUBE host processor; however, all of the image analysis routines and control and integration programs have been developed for execution on the NCUBE concurrent computer. A computer program that emulates the response of HERMIES-IIB is used for off-board development of the expert system rule base prior to implementation on the robot.

The expert system makes high level decisions and diagnoses unexpected occurrences. When a standard procedure is required, such as avoiding or removing an obstacle, or mapping an area, the expert system calls the appropriate routine which executes until completed or until an unexpected event generates an interrupt which returns control to the expert system. The rule base controls high level decisions and can call C-compiled navigation procedures. The rule base is loaded in an expert system shell, CLIPS, and linked to the navigation procedures. CLIPS and the navigation code run on one of the NCUBE nodes. Messages are passed from the NCUBE node to a host program which is linked to the HERMIES-IIB primitives on the VME rack.

Part of the CESAR effort is aimed at addressing crucial issues in systems integration so that research results can be integrated into the HERMIES prototypes. Recent experiments also included modules developed by groups at four university laboratories (Florida, Michigan, Tennessee, Texas) as part of a collaborative techbase development effort for which the HERMIES robots serve as user facilities. Detailed accounts of experiments with HERMIES-IIB can be found in the references.

The following material summarizes software development strategies in support of the latest experiment in which HERMIES-III was used to clean up a simulated chemical spill in the CESAR laboratory. The demonstration featured the capability to make smooth transitions between tele-operation and robot autonomy, the reconciliation of information in an a priori world model with information derived from sonars and CCD cameras, and the combined use of platform and manipulator degrees of freedom.

It is assumed that an a priori model of the environment surrounding the spill is known. The system uses this knowledge to create a path from the robot's current location to a location close to the spill. The robot then navigates to the spill, automatically avoiding unexpected obstacles en route. Once it has arrived, it senses the debris and uses a vacuum cleaner mounted on a manipulator to remove it. This process iterates until the sensing process can find no more debris. There are three main subtasks: path planning, path execution, and debris removal. An additional subtask permits operator intervention with the autonomous system. In every task, the operator is provided with a rich graphical description of the current state of the robot.

Software to operate and control the HERMIES-III robot in various experimental scenarios was developed around a simulated shared memory data structure. The shared memory model of interprocess communication was adopted because of its conceptual simplicity. This design decision made communication between various groups involved in the implementation effort relatively easy -- it was necessary only to define the format and the interpretation of the data structures written by each process without having to describe mechanisms by which these structures were communicated. Communications were assumed to be transparent by the authors of each module.

Some structure was imposed on the shared memory in addition to a simple list of variable names, types, and locations. Specifically, shared memory was divided into a number of blocks of contiguous memory, with one or more blocks associated with individual processes. The shared memory model is not without its problems. Perhaps the most obvious is that two processes may try to write to the same data item. In this situation, either of the processes may be correct depending on the state of the system or the time. Other problems include synchronization between processes and processes attempting to read a variable whose value has been only partially determined (e.g. only 4 bytes of an 8 byte record have been written). We avoided the first problem by specifying the system so that each process "owned" an area of shared memory to which only it could write. We solved the second problem by implementing a simple semaphore mechanism which guarded each area during updates.

Processes in this system communicate by accessing the shared memory, which is a replicated distributed data structure divided into exclusive-write areas (EWA). Each process making entries into the shared memory has associated with it one or more EWAs which only that process should change. In the event that multiple processes determine the values of single variables at different times, the "official" value is determined by a filtering process. Each EWA is a contiguous sequence of bytes. Shared memory is allocated by a special allocation process and is permanently memory resident. It has no internal structure at allocation time, rather, structure is imposed upon it at compile time through the use of compiler definitions. Addresses become available at run-time, through a call to the mem_attach() routine. Structure definitions and the relevant function prototypes are available by using included definitions.

The entire system is controlled by a single "state variable", and there is one (and only one) process in the system which determines the value of this variable. Decisions on the change from state to state are made by this process based on the current value of the state variable, and a state-dependent inspection of the contents of (possibly many) shared data areas. Individual processes inspect the value of this variable and respond in an appropriate manner.

Conclusions

Research and development at ORNL/CESAR centers on autonomous systems, unstructured environments, and multiple performing agents. A number of projects make use of the HERMIES mobile robot facilities at CESAR, and provide application focus for the R&D activities. Hardware and software environments have evolved to support the experimental part of the research. They facilitate software portability among systems, and re-use of applications software in different experimental scenarios. Message-passing concurrent computers have been incorporated successfully in our systems. Recent experiments show that the robot systems can perform robustly a variety of tasks of considerable complexity.

Selected References

C. R. Weisbin, "Intelligent Machine Research at CESAR", AI Magazine 4, Spring 1987, Vol. 8, No.1, (1987)

B. L. Burks et al., "Autonomous Navigation, Exploration, and Recognition", IEEE Expert, Winter 1987, pp 18-27, (1987)

C. R. Weisbin et al., "HERMIES-III: A Step Toward Autonomous Mobility, Manipulation and Perception", Robotica, Vol 8, pp 7-12, (1990)

J. P. Jones, "A Concurrent On-Board Vision System for a Mobile Robot", Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications, Pasadena, CA, (1988), CESAR-88/06

J. P. Jones, "On the Design of a Concurrent Image Analysis System", Proceedings of the Third International Symposium on Robotics and Manufacturing, ISRAM '90, Vancouver, Canada, 1990

F. J. Sweeney et al., "DOE/NE University Program in Robotics for Advanced Reactors", Program Plan, DOE/OR-884 Rev.2, May 1990

Figure 1: The CESAR robot/computer network.