AI Magazine Volume 8 Number 1 (1987) (© AAAI)

RESEARCH IN PROGRESS

Charles R. Weisbin

Intelligent-Machine Research at CESAR

Abstract The Oak Ridge National Laboratory (ORNL) Center for Engineering Systems Advanced Research (CESAR) is a national center for multidisciplinary long-range research and development (R&D) in machine intelligence and advanced control theory. Intelligent machines (including sensor-based robots) can be viewed as artificially created operational systems capable of autonomous decision making and action. One goal of the research is autonomous remote operations in hazardous environments This review describes highlights of CESAR research through 1986 and alludes to future plans.

Many technologies of interest to the Department of Energy (DOE) require hazardous operations in which intelligent machines could be used to advantage. Closest to home is the handling of radioactive material around reactors and processing plants, but similar problems arise with explosives and other hazardous chemicals. Underground mining puts humans at risk on a large scale, and underwater operations from exploration to maintenance of equipment and, eventually, to seabed mining have a similar potential. In fact, underwater robots have already been put to use (for example, Jason Jr. for surveillance of the remains of the sunken *Titanic*).

At a minimum, such a machine should be able to perform a useful task (lift something, cut something, observe something) and move between some staging point and the location of the task. Such goals lead to further required capabilities. The path to the task might be obstructed, and the items to be lifted or cut might not be quite the same size or shape or in the same location each time. The machine, therefore, must have sensors to tell it about its environment, and it must act in appropriate ways upon the information supplied by its sensors (for example, move left around the pillar, or cut the pipe closer to the valve). Decisions about which action to take can be made by a human if sufficient time is available, and the required information base and reasoning chain are limited. Our long-range goal is to enable the machine to make as many decisions as possible, thus freeing the human to spend more time in a supervisory and review capacity.

The experience of ORNL with the development of control systems (including remotely controlled machines for repairing equipment in reprocessing facilities) makes it a natural place to do research in intelligent machines operating in hazardous, unstructured environments and to develop prototypes with which to test the results of this research. CESAR was established at ORNL in 1983 as a national, multidisciplinary center for research in machine intelligence and advanced control theory and the application of this research to problems related to energy production and use. Potential benefits include reduced risk to humans in hazardous situations, machine replication of scarce expertise, minimization of human error induced by fear or fatigue, and enhanced capability using high-resolution sensors and powerful computers.

CESAR was created by the Division of Engineering and Geosciences, a part of DOE's Office of Basic Energy Sciences (Barhen et al. 1984; Weisbin et al. April 1985; Weisbin et al. June 1985; Hamel et al. April 1986). The center's current research objectives include development of methods for real-time planning with sensor feedback, determination

---

Charles R Weisbin is director of the Oak Ridge National Laboratory (ORNL) Robotics and Intelligent Systems Program and the Center for Engineering Systems Advanced Research, P O Box X, Building 6025, Oak Ridge, Tennessee 37831 He is also head of the Mathematical Modeling and Intelligent Control Section of the ORNL Engineering Physics and Mathematics Division and teaches an introductory course in artificial intelligence for the Computer Science Department of The University of Tennessee in Knoxville Weisbin is also a member of the Martin Marietta Corporation Artificial Intelligence Steering Committee

This article first appeared in the *ORNL Review* and is reprinted here by permission of the author

of concurrent algorithms for optimal implementation on advanced parallel computers, formulation of a learning theory for enhanced knowledge acquisition and interpretation, modeling of the dynamics of flexible structures, generation of automated sensitivity analysis for model simplification and parameter identification, formulation and testing of a comprehensive uncertainty analysis methodology, generation of a machine vision system based on principles of human vision, and inclusion of this research within a system integration framework encompassing concept demonstration and feasibility (Weisbin et al. 1987).

The kinds of research necessary for such a program and the results that we have already obtained will perhaps be better understood if I first describe the particular machine which we are developing as a testing ground for our research results. It is the hostile environment robotic machine intelligence experiment series (HERMIES).

The HERMIES-II Robot

The current experimental focus of the CESAR program is the mobile system HERMIES-II. William R. Hamel and Stephen M. Killough of ORNL's Instrumentation and Controls Division are the principal architects of HERMIES' evolution into a major research facility. HERMIES-II is a low-cost system developed for initial CESAR experimental activities with autonomous sensor-based robotic systems for use in unstructured work environments. Although limited in its basic performance capabilities, HERMIES-II incorporates mobility and manipulation as well as recently improved sensory feedback functions.

Description

HERMIES is a self-powered mobile robot system comprising a wheel-driven chassis, dual-manipulator arms, on-board distributed processors, and a directionally controlled sensor platform. HERMIES-II (Hamel et al. April 1986) is propelled by a dual set of independent wheels having a common axle alignment and driven by separate direct-current (dc) motors (see figure 1).

The on-board computer and electronic equipment are located in an enclosure mounted above the drive chassis, and the dual-arm manipulator torso is located above the computers and electronics. The manipulators are five-degree-of-freedom (DOF) units manufactured by Zenith/Heathkit and used on the HERO home robot. The torso assembly for the arms also includes a shoulder pitch motion for each arm and a base for single-shoulder rotation. The two-arm shoulder assembly has a total of thirteen DOF, and all axes are driven by stepping motors controlled directly by the Z8 microprocessor dedicated to manipulator control.

Sonar scan data are preprocessed on board HERMIES and then transmitted via a 2400-baud RS-232 radio link to either the NCUBE or LMI Lambda computers for navigation planning. A ring of five sensors, each of which consists of a phased array of four Polaroid transceivers, using sonar allows for a narrow effective beam width and rapid scan. The original stepping motor drives for the sensor pan-tilt control have been replaced with high-speed dc servodrives to permit the sonar ring to be stepped quickly. Consequently, the time required to scan a 180° region in front of HERMIES has been reduced from 80 to 7 s.

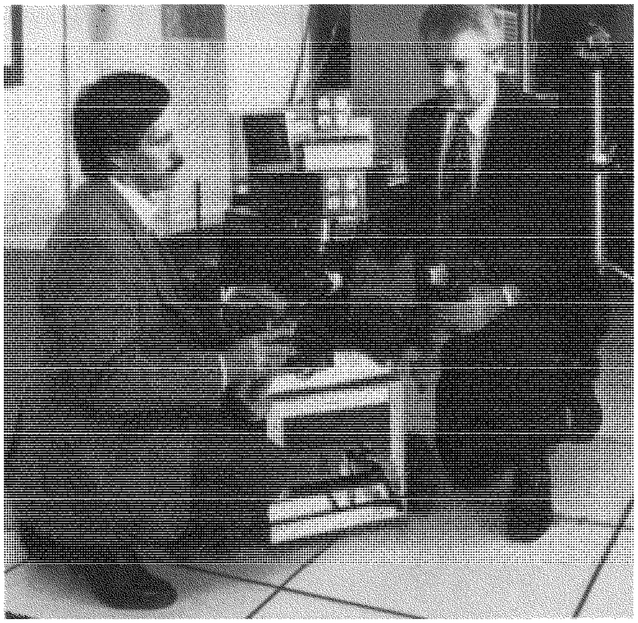


Figure 1 Examining the HERMIES-II Robot at CESAR.

The dc servodrive of the tilt platform has been designed to accommodate not only the sonar array but also an infrared range detector and dual Sony miniature charge-coupled-device (CCD) black-and-white cameras. The CCD cameras are part of a new image-processing system obtained to incorporate computer vision into HERMIES' sensor suite. The overall system is an International Robomation/Intelligence P-256 unit, which provides a pixel array of 256 x 256 spatial resolution with 8 bits characterizing possible brightness levels and an integral systolic array processor for reasonably high-speed execution of standard image operations. Much of our current image-processing research is being developed on our 64-node NCUBE hypercube parallel processor, in anticipation of a June 1987 target by which 16 NCUBE nodes (computational power roughly equivalent to 24 Vax 11/780s) will be mounted on-board HERMIES-II.

Control System Architecture

The current HERMIES-II control system consists of a main microcomputer and a satellite microprocessor. The main microcomputer is a single-board computer based on the Intel-8088 microprocessor and the IBM 0PC backplane. It controls an on-board 320K byte floppy disk drive and passes commands to a Zilog Z8603 single-chip microcontroller, which performs the robot's manipulator control functions.

Parameters are passed from the 8088 to the Z8 via a 9600-baud serial link. The dual wheels of the robot are driven independently by two gear-head dc motors that provide a linear speed of 0.154 m/s (0.5 ft/s). HERMIES-II's position is open-loop controlled through the on-board 8088 by real-time monitoring of the wheel encoders and on-off control of the drive motors.

The 8088 microcomputer uses the polyForth operating system and the Forth computer language. Forth is a flexible language designed for control applications, combining the ease of high-level programming with speeds approaching that of an assembly language. Forth word definitions have been used to construct a HERMIES command language for controlling the basic functions of the robot. The radio link is used to issue these commands to the on-board 8088 in a direct ASCII format. As an example, the Forth word command "2 0 FMOVE" causes the robot to move forward 2 ft (0.6 m).

Intelligent-Machine Navigation

The CESAR research in intelligent-machine navigation is currently led by Charles Jorgensen and Gerard de Saussure, supported by Ron Fryxell, Donna Jollay, Sitharma Iyengar, Nageswara Rao, Robert Ricks, Deanna Barnett, Moshe Goldstein, and Francois Pin. Collision-avoidance algorithms fall roughly into two categories: (1) if the position of an obstacle is known, the algorithms mathematically attempt to find optimal paths satisfying obstacle constraints and (2) if the position of an obstacle is unknown, environment-navigation algorithms are usually of the generate-test-move variety in which a tentative path is proposed and tested for potential collisions. The move is executed if no collision is detected; otherwise, a new tentative path is generated (Jorgensen, et al. 1986; Goldstein et al. 1987; Iyengar et al. 1986).

Sensory Feedback

Robot sensors include stereoscopic vision systems; fixed and mobile sonar range finders; laser range finders; touch, stress, and torque sensors; and collision detectors. Particular attention has been given at CESAR to the sonar systems used extensively for HERMIES-II navigation. Low-cost sonar devices function by sending a multifrequency sound pulse outward from a transducer in a cone-shaped wave front. The difference between time of emission and time of return is measured and an estimated distance calculated on the basis of how far the wave could travel in one-half the period measured.

Several well known difficulties occur when a robot uses sonar information to construct spatial distance maps from different scanning positions. First, sonar is sensitive to temperature changes. For example, if a sonar is calibrated at 27° C (80° F) and the actual room temperature is 16° C (60° F), a measured range of 11 m (35 ft) would be overestimated by 19.8 cm (7.8 in.) simply because of the temperature difference. Second, sonar is vulnerable to specular reflection and interacts with the texture of materials. The detectability of reflected sonar depends on signal energy and frequency. Frequencies useful in medical imaging are not practical for robotics. An example of this effect occurred in our early experiments using robot manipulators that attempted to grasp polyurethane foam blocks having extremely high sonar absorbency. For all intents and purposes, the blocks become sonar invisible. Other sonar problems result from the typically broad (35°) conical shape of the sonar beam. A sonar map made by a robot in the CESAR laboratory (see figure 2) illustrates some of these effects. Current work at CESAR involves exploring the use of edge finding with vision image processing as well as sonar.

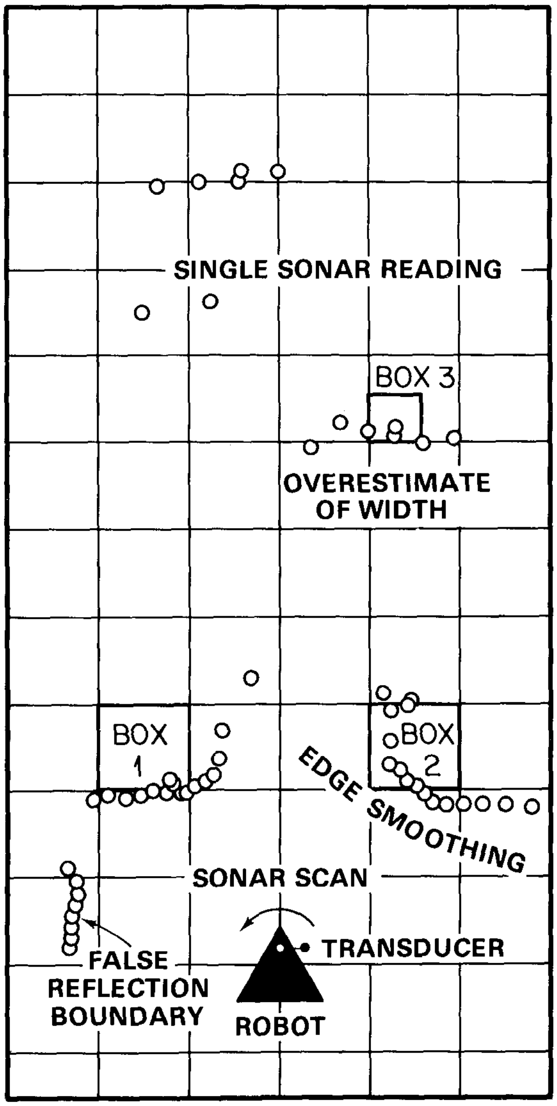


Figure 2. Sonar Recognition of Simple Obstacles

Navigation Control in Unexplored Terrain

It is not always easy for a robot to recognize a problem situation. Consider a simple maze problem (see Figure 3) in which the robot is given this control algorithm: When in a new area, first turn toward the goal you wish to reach. Take a sonar reading to see if the path is clear. If a path is clear, move. If it is not clear, take the first open path on either side of the line. Go one-half the distance to the goal. When you arrive at that location, turn toward the goal, and repeat the process.

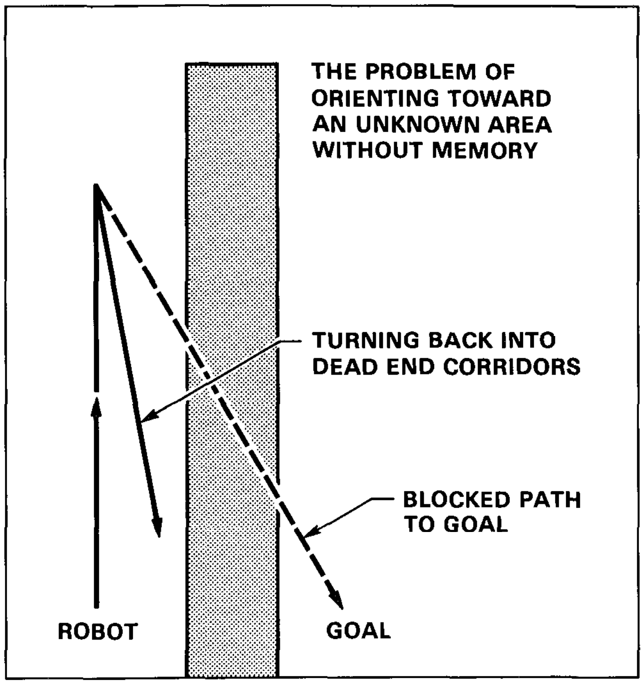


Figure 3. The Need for Memory.

At first glance, such an algorithm appears usable. The clear path nearest to an ideal straight line is always the one taken. The half-distance criterion also ensures that if the robot is far from the goal, it will move to it rapidly and will make small, careful moves as it gets closer. However, the robot has no memory. As shown in figure 3, the robot's goal is directly on the other side of the wall. If the robot follows the initial algorithm, it will scan the corridor and after about a 90° left turn find the first open path halfway to the goal. The robot will begin to move up the corridor away from the goal. After a short distance, the robot will be far enough so that half the distance can be traveled by making a turn back toward the goal. What happens? The robot again moves into the dead-end corridor. In other words, without memory, the robot would loop recursively and never reach the goal. With a memory, previously blocked areas can be designated off limits for a time, gradually squeezing the robot out of dead-end situations. Still other problems occur when navigation environments change quickly over time. Traversal can require continuous creation of new goals because unexpected obstacles invalidate previously formed navigation plans.

Learning during Autonomous Navigation

Ideally, an autonomous vehicle should collect information about its local environment and at the same time build or modify a global world model that can be useful for general purposes. Iyengar and others have developed a method that enables a mobile robot to select and navigate paths in unexplored terrain while it systematically acquires information about the terrain as it navigates (Iyengar et al 1986).

Learning begins by classifying information about the space a robot explores. Figure 4 illustrates four independent traversals about obstacles whose locations are unknown to the robot before it begins. Traversals are represented using spatial graphs that map the history of robot obstacle-avoidance movements onto a two-dimensional (2-D) coordinate system composed of edges (the paths traveled) and nodes (stopping points, turning points, or path intersections). The spatial graph provides a real-time data structure to record past movements; however, it is not efficient for planning future movements because no data are retained about the shapes of obstacles, areas of the room requiring further sensory analysis, or regions that are clear for maneuvering. Thus, a second type of graph structure called a *Voronoi diagram* is used to bound obstacles using polygons that can subsequently be labeled and associated with higher-order learning processes.

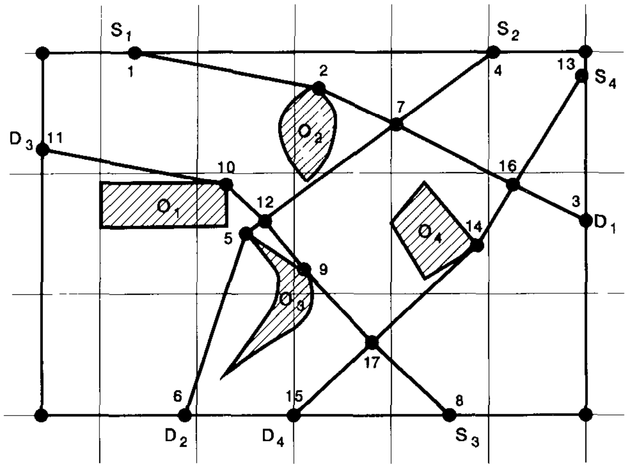


Figure 4. Four Traversals Completed from Starting Points (S_i) to Destinations (D_i) around Obstacles (O_i) Using Sensor-Based Navigation.

Consider determination of a new path from source point S_5 to destination D_5, as in figure 5. A virtual source point S_5^1 and destination D_5^1 are found from the Voronoi diagram corresponding to the nearest graph node points from the four

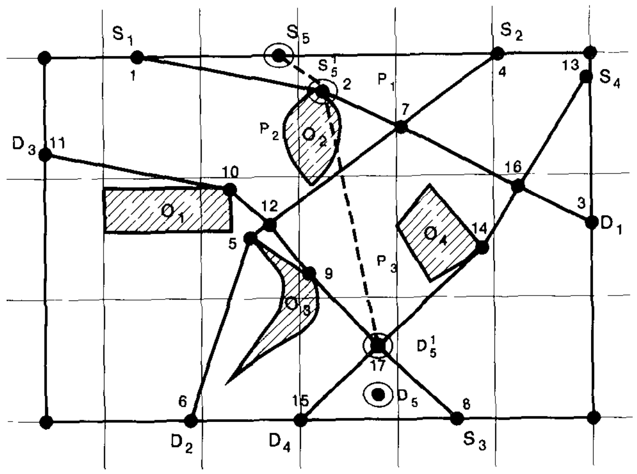


Figure 5 S_5 Source Point and D_5 Destination Point.

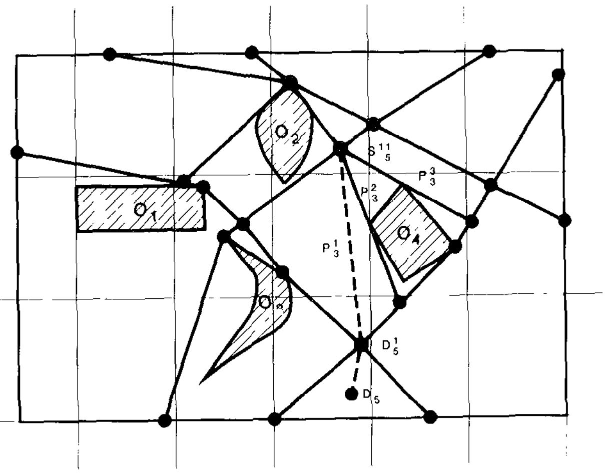


Figure 7. Exploration of Polygon P_3

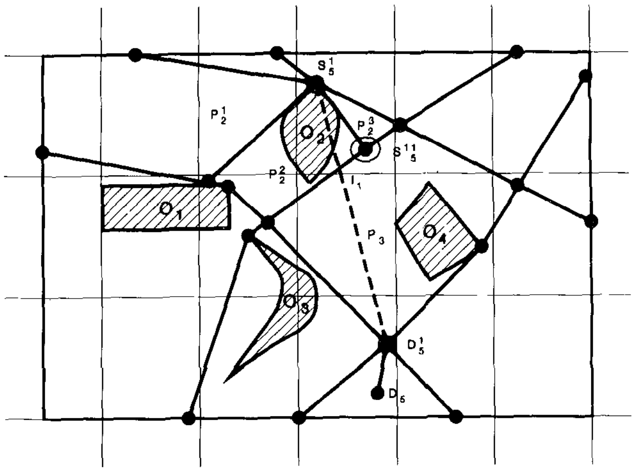


Figure 6. Exploration of Polygon P_2.

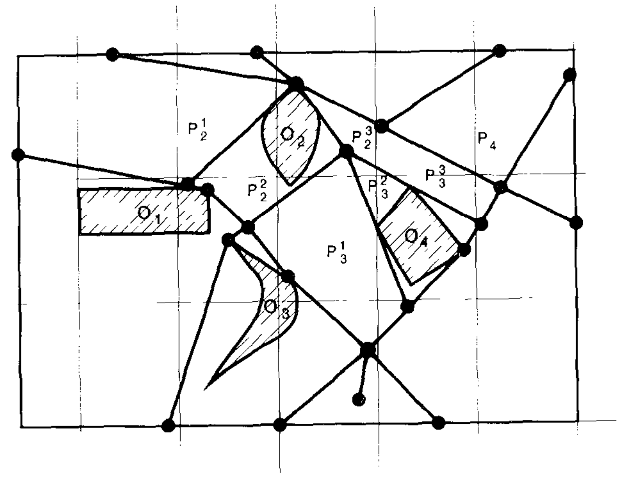


Figure 8. Terrain Model after the Path from S_5 to D_5 Is Consolidated.

previous explorations. The paths from S_5 to S_5^1 and D_5^1 to D_5 are determined using localized sensor-based navigation.

The polygon P_2 contains the source end of the line $S_5^1D_5^1$. The region P_2 is scanned using the sensor, and the polygon P_2 is partitioned into the regions P_2^1, P_2^2, and P_2^3, as in figure 6. The regions P_2^1 and P_2^3 are free polygons, and the region P_2^2 is an obstacle polygon with respect to the vertex S_5^1. At this point, the source end of $S_5^1D_5^1$ is contained in the polygon P_2^2. The robot navigates along the obstacle boundary nearest to $S_5^1D_5^1$, arriving at intersection S_5^{11}. Next, the path $S_5^{11}D_5^1$ is planned.

As illustrated in figure 7, the polygon P_3, which was previously unexplored, is partitioned (using sensor data) into the regions P_3^1, P_3^2, and P_3^3. P_3^1 and P_3^3 are free polygons, and P_3^2 is an obstacle polygon. At this stage, P_3^1 contains the source end of $S_5^{11}D_5^1$; the path $S_5^{11}D_5^1$ is directly traversed. The final leg from D_5^1 to D_5 is traversed using sensor-based obstacle avoidance.

The final spatial graph of the terrain is given in figure 8. Note that the obstacles O_2 and O_4 are bounded by polygons smaller than those shown in figure 4. Also, the polygons P_2^1, P_2^3, P_3^1, and P_3^3 are declared to be free polygons. Finally, regions P_2^3 and P_3^3 are combined to form a single free polygon. As additional paths are traversed, more and more polygons are explored, and the spatial graph is consolidated.

Extensions to Navigation Planning

Ron Fryxell, Robert Ricks, Deanna Barnett, Donna Jollay, and Gerard de Saussure are exploring the feasibility of using the process intelligent control (PICON) package as a basis for decision making and robotic control (Weisbin, deSaussure, and Kammer 1986). This software package was written by LISP Machine, Inc., for implementation on its Lambda

Characteristic	Action
1. Stationary and over 3 ft tall.	1. Start the navigation algorithm from the current position.
2. Stationary and less than 3 ft	2. Move forward to the obstacle, pick it up with the manipulator arms, put it to one side and proceed to the original destination. Anything shorter than 3 ft is guaranteed to be light enough to lift.
3. Has moved out of the way.	3. Proceed to the original destination.
4. Is moving away from the robot.	4. Wait for the obstacle to clear the path and proceed to the original destination.
5. Is moving toward the robot.	5. Check to the left and right with sonar; if clear, move out of the way. If both sides are blocked, go back to starting position and recheck escape routes to left and right.

Table 1. Diagnosis and Action on Unexpected Obstacles.

machine, which includes a dedicated LISP processor running in parallel and asynchronously with a Motorola 68010 processor. The software is partitioned between PICON (the LISP expert system, which operates on the dedicated LISP processor) and RTIME (written in C and running on the 68010.) RTIME handles routine tasks such as communications with the robot, sensor data analysis, path planning, and unusual-condition sensing, and PICON handles the operator interface, status monitoring, operation sequencing, and problem diagnosis. RTIME continuously monitors the communication channels between it and PICON for commands and passes to PICON information on the status of its operations. In addition, RTIME passes information on unexpected occurrences deduced from sensor data to PICON for high-level analysis and waits for commands on how to intelligently react to these events.

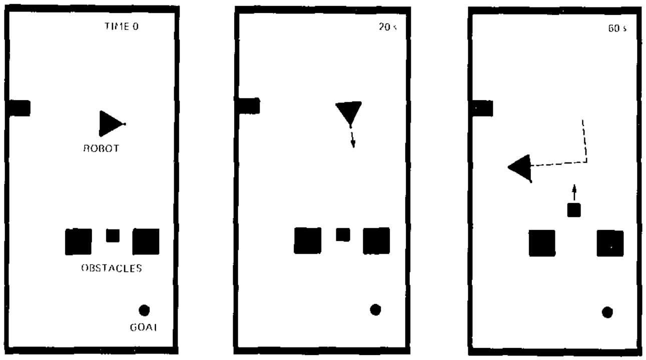
An example navigation problem illustrates the use of the system. A navigation module written in C is made available to RTIME to be activated by a message from PICON. The module requests information on the robot's initial position, the goal position, the navigation algorithms to be used, and the command to proceed; these data are passed from PICON to RTIME.

In each forward movement of the robot, the front-fixed sonar is continually activated, and if it indicates an unexpected obstacle within 0.7 m (2 ft), the robot stops and reports the fact to RTIME along with the distance actually moved. RTIME stops the navigation algorithm, reports the situation to PICON, and waits for commands. PICON then requests information from RTIME about the unexpected obstacle: Two front sonar readings are taken at a fixed-time interval followed by a reading at a higher elevation. RTIME then passes this information about obstacle characteristics (for example, size and shape) to PICON. A diagnostic rule base in PICON is used to determine an appropriate action (see table 1).

In this way, the robot can respond not only to a changed environment but a dynamically changing one as well. The system is easily modified to activate various robot responses and to accommodate a larger variety of sensors by simply modifying the diagnostic rule base and adding modules to RTIME. A simple robot navigation problem (and possible solutions using a PICON knowledge base of 32 if-then rules) is illustrated in figure 9.

In another approach toward navigation planning, Matthew Hall, formerly of ORNL's Engineering Physics and Mathematics Division, has used an electricity-conduction analogy. Obstructed squares are regarded as insulators, and clear squares are regarded as conductors. A potential difference is placed between HERMIES' current location and the goal, and HERMIES then proceeds along a path based on the line of maximum current density. When this path encounters an unobserved square, a sonar scan is requested, the world model is updated, and navigation continues. The main computational expense of this approach is incurred in computing the current density, which involves solving the Laplace equation. However, this calculation takes only a few seconds. This approach to path planning offers an alternative to heuristic approaches.

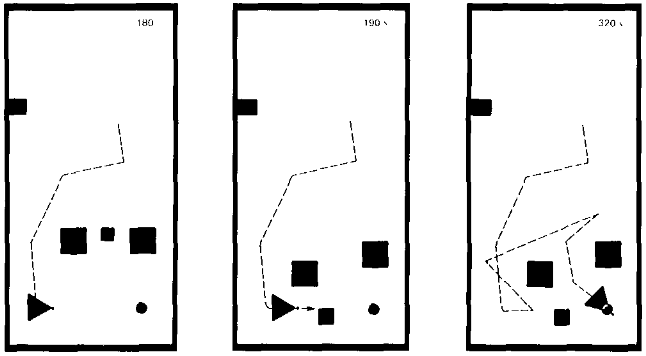
In the area of knowledge representation, Moshe Goldstein leads in the development of a 3-D world modeling capability (Goldstein et al. 1987) based on methods of combinatorial geometry (CG), which are used widely in Monte Carlo particle transport calculations. Discrete measurements of range information that quantify the distances from the sensor focal plane to the object surface, are transformed to the surface representation. First, each measured point on the object surface is surrounded by a small sphere with a radius deter-



A INITIAL ROOM CONFIGURATION

B ROBOT FACES GOAL FINDS OBSTRUCTION AND STARTS MOVING TO WITHIN 3 ft OF OBSTRUCTION

C SMALL BLOCK MOVED TOWARD ROBOT ESCAPE ROUTE FOUND TO THE ROBOT S RIGHT



D SMALL BLOCK REPLACED PATH CHOSEN TO THE ROBOT S RIGHT

E OBSTACLES MOVED TO BLOCK CLEAR PATH TO GOAL

F ALTERNATE PATH CHOSEN TO GOAL

Figure 9 Sample Solution to a Navigation Problem Involving a Dynamic Environment

mined by the range to that point. Then, the 3-D shapes of the visible surfaces are obtained by taking the (Boolean) union of all the spheres. The result is an unambiguous representation of the object's boundary surfaces. The prelearned partial knowledge of the environment can also be represented using the CG method with a relatively small amount of data. Using the CG type of representation, distances of desired direction, to boundary surfaces of various objects are efficiently calculated. This CG feature is particularly useful for continuously verifying the world model against the data provided by the range finder, and for integrating range data from successive locations of the robot during motion. The feasibility of the proposed approach has been demonstrated using simulations of a spherical robot in a 3-D room in the presence of moving obstacles and inadequate prelearned partial knowledge of the environment.

Advanced Computing

To enable a robotic system to work effectively in real time in an unstructured environment, a variety of highly complex mathematical problems, such as online planning, vision, sensor fusion, navigation, manipulation, dynamics, and control, must be solved. The computational requirements of these problems fall into the "supercomputer" class, but ultimately we need to solve them on board the autonomous robot. Jacob Barhen leads the CESAR effort in advanced computing to exploit concurrent computation (Barhen and Palmer 1986; Barhen 1985; Barhen and Halbert 1986; Barhen and Babcock 1984; Barhen August 1986), including the capability to dynamically balance the computational load among all processors in the system. He is supported in this effort by Ralph Einstein, Edith Halbert, Benjamin Toomerian, Judson Jones, Reinhold Mann, Charles Glover, and Michelle Clinard.

Hypercube Ensembles

Hypercube ensembles refer to a multiprocessor design in which N (= 2^d) identical microprocessor nodes are connected in a binary d-dimensional cube topology; each processor has its own local memory and is connected to d nearest neighbors directly. Communication is performed by message passing; the furthest communication distance between any two processors in the ensemble is d. For illustrative purposes, a few hypercubes of low order are shown in figure 10. A hypercube looks topologically identical from the point of view of each node: There are no corner-versus-edge or root-versus-leaf nodes as in regular grids or trees. This symmetry is particularly attractive for simplifying the dynamic reconfiguration of the system.

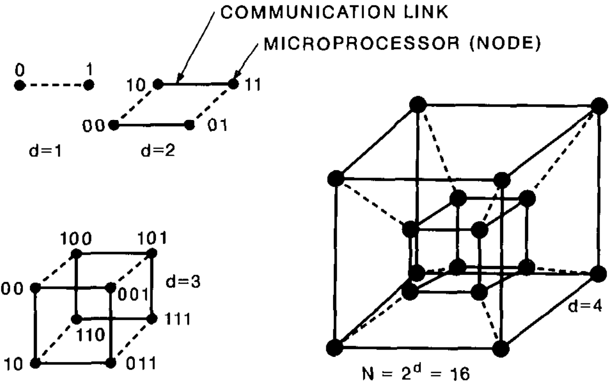


Figure 10 Hypercube Architecture in d Dimensions. An Order-d Hypercube Is Constructed Recursively from Two Order-(d-1) Cubes.

The CESAR NCUBE Hypercube

The concurrent computation system now being investigated at CESAR was developed by NCUBE Corporation (see figure 11). It contains 64 processors (6-d cube) in its initial implementation; the number of processors that can be accommodated is 1024, each designed to run conventional computer programs at about the speed of 1.5 VAX 11/780's. Each processor has 128K bytes of local memory (which can be upgraded to 500K/node) and can communicate directly with ten other processors through direct-memory access channels.

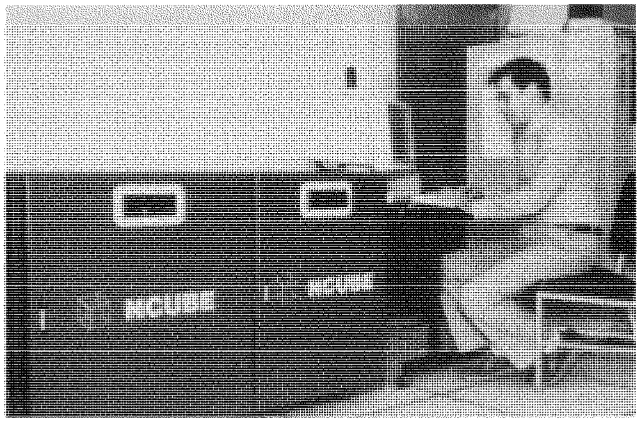


Figure 11. Operating the NCUBE Concurrent Processor in the CESAR Laboratory for Machine Intelligence. Up To 1024 Processors Can Be Enclosed in the Left-Hand-Side Box

The importance of this design for mobile robotics research is that when fully loaded, the hypercube has a capacity of approximately 500 million floating-point operations per second (500 Mflops) contained within a volume less than 1 m^3 (including cooling and power supply) with a power consumption of approximatley 8 kW. The system can easily be scaled down for less demanding and compact applications (for example, 8 Mflops in something the size of an IBM PC AT). Compilers for Fortran-77 and C languages are available. The NCUBE node processor (see figure 12) is a complex chip of about 160,000 transistors that integrate a memory interface; communication links; and a 32-bit, general-purpose processor, including 32- and 64-bit floating point on chip.

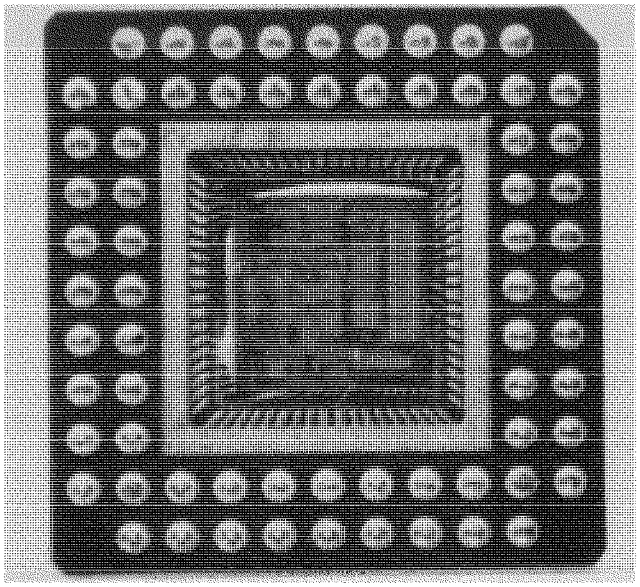


Figure 12. The NCUBE Node Processor.

Concurrent Algorithms for HERMIES Navigation

Matthew Hall originally developed the NCUBE software for driving HERMIES by partitioning tasks into separate processes. Each process can be executed in a different node on the array board except that processes requiring input-output (I-O) with the outside world must be executed on the controller board. Current HERMIES-II software includes six processes: (1) input of instructions from a terminal, (2) input of data from HERMIES, (3) output of instructions to HERMIES, (4) graphic display of the world model, (5) processing of sensor data and subsequent world modeling, and (6) navigation. Processes 1 through 4 must be run on the I-O board, while processes 5 and 6 can be run on the array board. As the complexity of HERMIES sensor data and environment increase, it is anticipated that processes 5 and 6 will be split into many different processes. Most recently, Judson Jones has extended the NCUBE software capability to allow positioning of the HERMIES-II manipulator arms.

Advanced Operating Systems with Embedded Reasoning

The research in this area is a major long-term endeavor toward implementing machine intelligence through real-time control of a dynamically reconfigurable multiprocessor architecture. Reasoning and control functions are intimately associated with an operating system that provides for interrupt capability, priorities, communication, scheduling, and so on. This development includes all research in the four tasks discussed in the following subsections.

Treatment of Precedence Constraints (ROSES) The robot operating system expert scheduler (ROSES) system (Barhen and Halbert 1986) is being developed to schedule precedence-constrained tasks for computation by an ensemble of concurrent processors. This endeavor is particularly difficult when the number of tasks required exceeds the number of available processors or when the interconnection topology of the task graph differs from the interconnection topology of the computation ensemble. Multiprocessor scheduling has been studied extensively (Coffman 1976; Graham et al. 1979); excellent reviews can be found in the literature. The task of multiprocessor scheduling is to determine the appropriate sequence of tasks for assignment to available resources given a number of tasks and their associated precedence constraints. The ROSES approach seeks near-optimal solutions by combining heuristic techniques to minimize scheduling time as a function of the number and relationship of tasks with data structures in order to most efficiently use available computer memory and algorithms to control the search process and eliminate dead ends.

Hard-Real-Time Capabilities Many tasks in intelligent autonomous systems are expected to have stringent execution deadlines. Such tasks are said to induce "hard-real-time" constraints on the system and present major difficulties in the design of the scheduling algorithms for a distributed operating system.

When previously unanticipated tasks, which must be completed prior to some absolute deadline, arrive at a processor node, methods must be developed to guarantee such compliance. The first attempt is to find an open window within the computation schedule already developed for that processor. If this attempt is unsuccessful, task shuffling within the schedule is considered (that is, a check to determine whether already scheduled nonperiodic tasks can be rescheduled at an earlier time to guarantee the newly arriving task is completed prior to its deadline). Finally, internode task bidding is considered for guaranteeing successful task completion. Algorithm development for such scheduling capability is currently under way.

Virtual Time In multiprocessor systems in which task computations are migrated among processor nodes to improve efficiency, one cannot guarantee that messages intended for a particular process will arrive in the temporal order in which they were sent. A virtual-time paradigm is being developed (Einstein and Barhen 1987) as a synchronization mechanism to cope with this problem. This research builds on a version of the Caltech hypercube simulator made available to CESAR.

The Caltech simulator as received by CESAR was such that (1) all nodes executed the same program, (2) message passing was assumed to occur only between nearest neighbors, and (3) all messages were received in the sequence they were sent. Because the CESAR emphasis is on problems characterized by structures irregular in time and space, modifications were made such that (1) nodes can now execute totally different programs, (2) message passing is from any process to any other at any time, and (3) tasks are initiated after the necessary messages have been received. The next phase of research involves the implementation of the virtual-time algorithms (including modification of tentative plans based on new messages from other processors that invalidate previous assumptions) into the NCUBE VERTEX operating system.

Simulated Annealing To address the load-balancing problem, Barhen has proposed using a simulated annealing method. Simulated annealing has been proposed as an effective method for determining global minima of combinatorial optimization problems involving many degrees of freedom. In analogy with statistical mechanics, each processor n could correspond to a lattice site in d dimensional space, and each process i would correspond to a particle. The kinetic energy of particle i is identified with the nonmessage-passing portion of the execution time of the corresponding process. A potential energy V_i represents the total time spent by process i for communication. To induce processes to spread out, a repulsive potential is introduced corresponding to the difference between the specific processor computational load and the average. The total energy of the system is then minimized to determine appropriate task allocations among processors.

Hypercube Algorithms for Robot Dynamics

The pioneering work of J. Y. S. Luh and C. S. Lin (1982) of Clemson University on scheduling of parallel computations for a computer-controlled mechanical manipulator has served as a benchmark for subsequent R&D of parallel algorithms for robot dynamics. Barhen and Einstein have examined the same problem using a modified version of ROSES, and results for the forward recursion (base-to-tip equations of motion) involving 144 tasks indicate that the speed of computation increases with the increasing number of processors—but only up to a point. Beyond this, adding processors is a waste of computing resources.

Direct running-time comparisons are not appropriate because of the different hardware used by Lin and the CESAR group; the individual NCUBE nodes were designed by NCUBE to be about an order of magnitude faster than previously used microprocessors. Timing studies are under way now to verify these assertions. However, the load balance reported by Barhen and Einstein for the most unbalanced node in this benchmark was about 95% for four processors compared with 5% reported by Luh and Lin in an architecture in which one processor is assigned for each joint.

Artificial Neuromorphic Systems

Artificial neuromorphic systems (ANS) are adaptive dynamical systems that can carry out useful information processing by means of their state response to initial or continuous input. The excitement behind ANS is the hope that they can provide practical solutions to some of the problems that have confounded computer science and artificial intelligence for the last 30 years. These include fully parallel search through spatiotemporal information patterns, automated acquisition of knowledge from observation correlated activity, and real-time control of complex systems through self-organizing associative recall of command sequences and fast adaptive optimization.

The CESAR research in this area (Barhen 1987) is relatively new and focuses on four major themes: (1)simulation of large scale neural networks on a hypercube supercomputer, (2) combinatorial optimization, (3) machine intelligence (for example, expert systems, learning, vision, multisensor integration, and so on), and (4) cross-disciplinary issues such as stability and storage capacity.

Machine Vision Based on Human Neural Mechanisms

The CESAR team has initiated a research effort to develop a robot vision system based on principles of human vision (for example, massive parallelism, dynamic feedback, and multilayer pattern recognition) (Jorgensen, Gawronski, and Holly 1986). During the first year (research was led by Charles Jorgensen and Richard Gawronski), the modeling effort concentrated on understanding the electrochemical processes in the retina that follow photochemical conversion of the light impinging on the human eye. Two types of neural models were considered. The first represents neural layers in terms of static 2-D linear equations using a linear matrix and linear feedback. The second considers dynamic 2-D nonlinear processes using matrixes of nonlinear differential equations. The first set of equations was parameterized using psychophysical data drawn from subjective judgments about the intensity of three visual illusions (see figure 13). The experimental results were used in a Fourier solution process, which was then applied to new illusions and compared with human subjective results through dimensional plots of transformed pixel intensities. The second set of equations was studied using a computer simulation operating on digitized picture matrixes.

Figure 13. Testing the New CESAR Machine Vision System Being Developed for the U.S. Army Human Engineering Laboratory.

Although near-term generation of neural-based processors is still out of reach, the development of systems having silicon-based neurons at the front end and pattern-recognition technologies at the higher levels might be feasible. A transition from static to dynamic models and models for higher-level perception remain to be developed.

Finally, appropriate task decomposition of image-processing algorithms for asynchronous parallel operation will be essential to achieve the required speed improvements. Current CESAR research in machine vision and multisensor integration using the NCUBE is led by Judson Jones and Reinhold Mann.

CESAR Research Manipulator

The CESAR research manipulator (CESARM) is being developed by Scott Babcock and Bill Hamel to support their studies in robot dynamics and control (see figure 14). The CESARM incorporates several fundamental characteristics important for mobile operation. Mobility requires that overall weight and power consumption be minimized; thus, manipulators must be designed with low weight-to-capacity ratios. The manipulator being developed weighs approximately 68 kg (150 lbs) and can lift about 13.6 kg (30 lbs)—a weight-to-capacity ratio of approximately 5, a factor of 4 improvement over typical industrial manipulators.

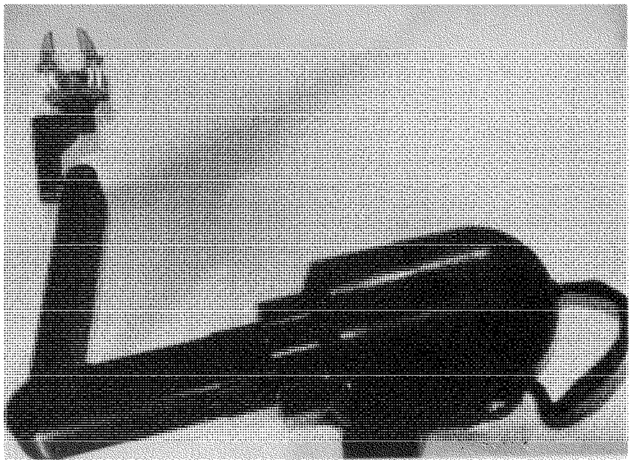


Figure 14 The CESAR Research Manipulator.

Drive motors for the upper arm roll, elbow pitch, and wrist pitch, yaw, and roll are centralized at the shoulder to minimize the inertia and the actuator size (see figure 15). Note that the 3-DOF wrist is cable driven.

Complex tasks will require high dexterity. The CESAR research manipulator has 7 DOF plus the parallel jaw gripper to be used as the initial end effector. The manipulator's low-friction drive train, together with the redundant degree of freedom, provides an ideal research tool for dexterous manipulation. The manipulator incorporates a unique 3-DOF spherical wrist whose singularities occur only at the extremities of motion. A flange interface between the wrist and the parallel jaw gripper will facilitate research using other end effectors, such as multifingered hands.

The CESAR experimental manipulator design is based on teleoperation technology developed at ORNL as part of DOE's Consolidated Fuel Reprocessing Program (Hamel and Babcock 1986). The CESARM control system is designed around the VME bus. The initial implementation uses three Motorola 68010 processors and the Forth programming language. Plans include upgrading to Motorola 68020 processors when practical and eventually interfacing CESARM to the CESAR hypercube.

Currently, mass and inertia properties of the manipulator are being determined experimentally for use in a mathematical model, and a simple torsional pendulum has been built for use in determining individual link inertias. The re-

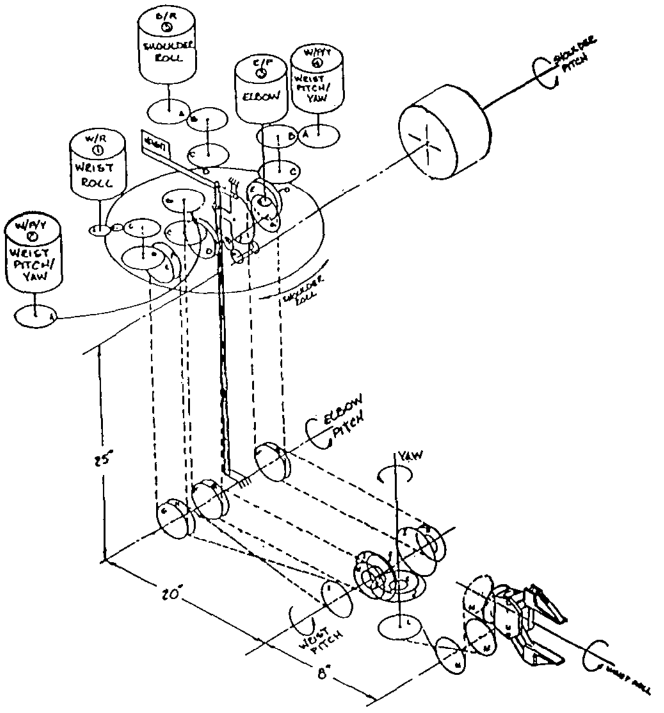


Figure 15. CESAR Research Manipulator Power Transmission.

maining model parameters, such as compliance and friction, will be identified from manipulator test data. This effort will provide a verified model with complete parametric data that should be useful to the entire robotics research community.

Near-term plans include experimental verification of controller structures similar to recently formulated inverse-dynamics position control and position-force control algorithms for manipulators having compliant drivetrains (Forrest-Barlach and Babcock 1987; Forrest and Babcock 1985). This experimental verification is important because the cycle time of typical industrial manipulators is limited by vibrational characteristics associated with drivetrain compliance, and current industrial robotic control is based on rigid-body dynamics. Our long-term goal is the coordinated control of two compliant manipulators mounted on a mobile platform.

Sensitivity and Uncertainty Analysis

Ed Oblow leads the CESAR effort in developing methods for automated derivative generation and systematic uncertainty analysis. Oblow is currently supported in this effort by Martin Beckerman, Francois Pin, and Brian Worley. First derivatives (sensitivities) are required for model-simplification studies and parameter identification of complex calculations such as robot dynamics. A comprehensive uncertainty analysis is required to handle numerical data uncertainties (such as occur in sensor measurements), rule and heuristic uncertainties (for searches and decision), and structural uncertainties for vision and language problems. Each of these uncertainties must be properly represented and combined for use in different decision-making environments. This work is jointly sponsored by DOE's Office of Nuclear Waste Isolation.

Sensitivity Analysis

Sensitivity theory has been used in many fields over the past two decades to assess the importance of variations in modeling data and parameters on calculated model results (Weisbin et al. 1982; Stacey 1974; Tomovic and Vukobratovic 1972; Frank 1978; Cacuci et al. 1980; Myers 1971). Sensitivity determination based on reruns is usually unwieldy for complex programs having large databases. Statistical methods are not comprehensive enough and require associated a priori engineering judgment of importance for much of the data set. Adjoint methods have required significant code development prior to implementation.

Oblow, Pin, and Worley have developed and validated an automated procedure for performing sensitivity analysis. The procedure uses a new Fortran precompiler that has computer calculus capabilities. The new compiler is called the gradient enhanced software system (GRESS); in it each Fortran line corresponding to storage operations is analytically differentiated, and total first derivatives are propagated using the chain rule. The GRESS system (Oblow 1983; Oblow 1985; Oblow, Pin and Wright 1986; Worley and Wright 1985) has now been successfully tested on several major codes. The major advantage of the GRESS language is its ability to process the model source program as data; no special effort is needed to specify the model.

Uncertainty Analysis

Oblow has also developed a theoretical framework for uncertainty analyses that can be used in artificial intelligence domains. The result is a hybrid uncertainty theory (O-theory) developed to bridge the gap between fuzzy-set theory and Bayesian inference theory (Oblow 1986). The intersection operator proposed combines belief from independent sources into a new belief set in a manner previously thought only possible in a probability framework using Bayes' theorem.

The basic operations defined for O-theory include dominance and order, union, intersection, complement, and general mappings. The formal connection between mass and probability derived by Oblow suggests that the distinctions between Dempster-Shafer theory (DST) and probability theory are far less consequential than previously believed. The Dempster-Shafer theory conception of uncommitted belief was shown to be compatible with classic probability theory. O-theory is currently being tested with reference to expert systems and sensor fusion by Martin Beckerman.

CESAR Program Development and Interactions

CESAR has organized and led two major national workshops. In November 1983 a workshop entitled "Research

Needs in Intelligent Machines'' was held to develop long-range goals and priorities. In August 1985 CESAR conducted a second workshop, ''Planning and Sensing for Autonomous Navigation,'' in conjunction with the International Joint Conference on Artificial Intelligence.

Based on R&D results to date, other federal agencies have elected to participate in the sponsorship of CESAR. The U.S. Army Human Engineering Laboratory (HEL) is supporting research in soldier-machine system development, which has applications for projects that require dexterous manipulation in hazardous environments, such as explosive ordnance disposal and vehicle refueling and decontamination. HEL is also supporting research that explores the feasibility of applying human neural models to advanced robotic vision systems. The Air Force Wright Aeronautical Laboratories is sponsoring research in concurrent computation that exploits the potential speed, compactness, and versatility offered by the CESAR hypercube ensemble machine with application to multisensor integration. The Strategic Defense Initiative program is supporting CESAR research in real time operating system development and neural networks. Finally, the DOE Office of Nuclear Energy is sponsoring a program coordinated by CESAR, that is intended to lead to the development of a surrogate man operating in nuclear reactor environments.

Summary and Conclusions

The following paragraphs present a summary of CESAR's achievements through fiscal year 1986.

The directionally controlled sensor suite of our HERMIES-II mobile robot was upgraded to include a phased array of sonar sensors and a new vision system. HERMIES-II now has full-duplex communication with our NCUBE hypercube ensemble computer and a dedicated LMI LISP machine and has been controlled by both as distributed brains in various experiments involving near-real-time task planning, path planning, and vision.

Navigation algorithms have been successfully generated and augmented using learning techniques that record and synthesize information from multiple journeys and allow for continuous transition from local- to global-path optimality. Deterministic navigation approaches based on analogy to electric conduction (that is, obstructing squares regarded as insulators) have been implemented and offer an alternative to existing heuristic search techniques.

A new research manipulator (CESARM) adapted from the Consolidated Fuel Reprocessing Program has been designed and built and now allows our analytical work on the modeling and control of manipulators to be experimentally verified. The manipulator weighs approximately 68 kg (150 lbs) and can lift approximately 13.6 kg (30 lbs), which is a weight-to-capacity ratio of about 5, a factor of 4 improvement over typical industrial manipulators (which are necessarily heavy and rigid so they can precisely perform repetitive operations).

A computer design based on a hypercube architecture is currently being investigated by CESAR. The system can accomodate 1024 32-bit processors, which were developed by NCUBE Corporation. Each processor is designed to have the power of about 1.5 VAX 11/780s and can be connected to their nearest neighbors; however, because very large scale integration technology is used, the total physical volume of the NCUBE machine is less than 1 m^3. The initial CESAR configuration is a six-dimensional cube that became fully operational in January 1986. Current research focuses on generalizing the simulated annealing global-optimization methodology to systems having a varying number of degrees of freedom and on exploring its applicability to the static and dynamic load balancing of large-scale, message-passing hypercube multiprocessors. A significant effort is under way to develop operating-system concepts such as virtual time. In addition, the development of parallel algorithms for machine vision is receiving increasing attention. In particular, work was initiated to study the applicability of human neural mechanisms to robotic vision. Mathematical models were developed that describe the first three layers of the retina; these models were evaluated using psychophysical experiments to parameterize the retinal equations and independently test their validity.

Our GRESS calculus precompiler is now being used for automated derivative generation that supports model-simplification studies and parameter identification. Each Fortran line corresponding to storage operations is analytically differentiated, and total first derivatives are propagated using the chain rule. The GRESS system has now been successfully tested on several major codes. The primary advantage of the GRESS language is its ability to process the model source program as data; no special effort is needed to specify the model.

A new uncertainty theory (O-theory) has been developed in an attempt to combine and synthesize the strengths of Dempster-Shafer theory), fuzzy-set theory, and Bayesian inference theory to retain the probabilistic basis of Bayesian inference theory, the beliefs and possibilities of Dempster-Shafer theory, and the mathematical diversity and rigor of fuzzy-set theory. The theory is examined with respect to multisensor integration and expert system applications.

In making the transition from the teleoperated systems of today to the autonomous systems of tomorrow, CESAR is completing preliminary plans for HERMIES-III. Barry Burke serves as the coordinator for this effort. This robot will have two dexterous arms (dual CESARMs instead of the one CESARM of HERMIES-II) and will outperform HERMIES-II in sensing the environment, because of its high-resolution vision and a laser scanner. HERMIES-III will also be smarter than its predecessor. Its brain will be a mobile hypercube parallel computer, which has enormous processing power for rapid reasoning, learning, and decision making.

CESAR's long-range goals include allocating tasks and facilitating cooperative problem solving among humans and machines. This is a major effort that Lynn Parker, Bill Hamel, Francois Pin, and others have only just begun. We will be a step closer to these goals when we complete the intelligent machine operating system, which fully exploits the hypercube processor for scheduling tasks, load balancing, and synchronization. We will continue research on modeling the human visual process at higher levels (for example, colinearity and periodicity) and for time-varying imagery. This research will be merged with conventional vision methodology for high-level scene analysis and with information from other types of sensors (sensing force, pressure, and so on). CESAR will also study control theory for dual-armed, closed-loop manipulation.

We hope that the long-term nature and continuity of the program will enable us to build commonsense knowledge representations, to deal wisely in using limited fast memory (what shall we allow the robot to forget?), and to develop algorithms to allow the robot to learn from experience. Finally, we expect to augment expert systems for robot control to include a measure of uncertainty used in decision making.

References

Barhen, J April 1986 Concurrent Algorithms for Asymetric Neural Networks, Technical Report CESAR-86/32, Oak Ridge National Lab.

Barhen, J August 1986 Hypercube Ensembles: An Architecture for Intelligent Robots In *Special Computer Architectures for Robotics and Automation*,ed J. Graham New York: Gordon and Breach

Barhen, J. 1985. Hypercube Concurrent Computation and Virtual Time for Robotic Applications In Proceedings of the International Conference on Robotics Research, 5 1–5 30

Barhen, J , and Babcock, S. M 1984 Parallel Algorithms for Robot Dynamics, Technical Report, MS84-499 Paper presented at Robotics Research: The Next Five Years and Beyond

Barhen, J ; Babcock, S M ; Hamel, W R ; Oblow, E M ; Saridis, G N ; deSaussure, G ; Solomon, A. D ; and Weisbin, C. R. 1984 Basic Research in Intelligent Robotic Systems Operating in Hostile Environments: New Developments at ORNL In Proceedings of the 1984 National Topical Meeting on Robotics and Remote Handling in Hostile Environments, 105–116

Barhen, J , and Halbert, E. C. 1986 ROSES: An Efficient Scheduler for Precedence—Constrained Tasks on Concurrent Multiprocessors In *Hypercube Multiprocessors '86*, ed M T Heath Philadelphia, Penn : SIAM

Barhen, J , and Palmer, J 1986 The Hypercube in Robotics and Machine Intelligence *Computers in Mechanical Engineering* 4(5):30–38

Cacuci, D. G et al. 1980 Sensitivity Theory for General Systems of Nonlinear Equations *Nuclear Science Engineering* 75: 88–100

Coffman, E G 1976 *Computer and Job Scheduling Theory* New York: Wiley

Einstein, J R., and Barhen, J 1986 Virtual-Time Operating-System Functions for Robotics Applications on a Hypercube, Technical Report CESAR-86/34, Oak Ridge National Lab

Forrest-Barlach, M G , and Babcock, S. M January 1987 Inverse Dynamics Position Control of a Compliant Manipulator *IEEE Journal of Robotics and Automation.*

Forrest, M G , and Babcock, S M 1985 Control of a Single-Link, Two-Degree of Freedom Manipulator with Joint Compliance and Actuator Dynamics In Proceedings of the 1985 International Computers in Engineering Conference, 189–197

Frank, P. M 1978 *Introduction to Sensitivity Theory* New York: Academic

Goldstein, M ; Pin, F G.; deSaussure, G ; and Weisbin, C R 1987 3-D World Modeling with Updating Capability Based on Combinatorial Geometry. Paper presented at the Jet Propulsion Laboratory Space Telerobotics Workshop

Graham, R. L ; Lawler, E L ; Lenstra, J K ; and Kan A H G R 1979 Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey *Annals Discrete Math* 5: 169.

Hamel, W. R., and Babcock, S M 1986. Manipulator Technology: The Critical Element of Useful Autonomous "Working" Machines, Technical Report, CESAR-86/56 Paper presented at the International Congress

Hamel, W R ; Babcock, S M ; Hall, M. C G ; Jorgensen, C C ; Killough, S M ; and Weisbin, C. R 1986 Autonomous Robots for Hazardous and Unstructured Environments *Robots-86*

Iyengar, S. S.; Jorgensen, C C ; Rao, S V. N ; and Weisbin, C R. 1986 Robot Navigation Algorithms Using Learned Spatial Graphs *Robotica* 4: 93–100

Jorgensen, C C.; Gawronski, R ; and Holly, F. 1986 Modeling Early Stages of Human Retinal Processes Technical Report, ORNL/TM-10031, CESAR-86/05, Oak Ridge National Lab.

Jorgensen, C C.; Hamel, W R ; and Weisbin, C R January 1986 Autonomous Robot Navigation *Byte*, 223–235

Luh, J. Y S , and Lin, C S 1982 Scheduling of Parallel Computation for Computer Controlled Mechanical Manipulator *IEEE Transactions of Systems, Man and Cybernetics* SMC-12: 214

Myers, R H 1971. *Response Surface Methodology* Boston: Allyn and Bacon

Oblow, E M 1986 O-Theory—A Hybrid Uncertainty Theory Technical Report, ORNL/TM-9759, CESAR-86/02, Oak Ridge National Lab

Oblow, E. M 1985. GRESS—Gradient Enhanced Software System—Version D Users Guide: July 1985, Technical Report, ORNL/TM-8339, Oak Ridge National Lab

Oblow, E M 1983 An Automated Procedure for Sensitivity Analysis Using Computer Calculus: May 1983, Technical Report, ORNL/TM-8776, Oak Ridge National Lab

Oblow, E M ; Pin, F G ; and Wright R Q. 1986. Sensitivity Analysis Using Computer Calculus: A Nuclear Waste Isolation Application *Nuclear Science and Engineering* 94(1): 46–65

Stacey, Jr , W. M 1974 *Variational Methods in Nuclear Reactor Physics.* New York: Academic

Tomovic, R , and Vukobratovic, M 1972 *General Sensitivity Theory* New York: American Elsevier

Weisbin, C R ; Hamel, W R ; and Barhen, J 1987 Research and Development Program Plan for the Center for Engineering Systems Advanced Research (CESAR). Technical Report ORNL/TM-10232, Oak Ridge National Lab

Weisbin, C. R ; deSaussure, G ; and Kammer, D W 1986 Self Controlled: Real-Time Expert System for an Autonomous Mobile Robot *Computers in Mechanical Engineering* 5(2): 12–19.

Weisbin, C R ; Barhen, J ; Swift, T. E ; deSaussure, G ; Jorgensen, C C ; and Oblow, E M 1985 HERMIES-I: A Mobile Robot for Navigation and Manipulation Experiments. In Proceedings of the Robots-9 Conference, 1–24

Weisbin, C R ; Barhen, J ; deSaussure, G ; Hamel, W R ; Jorgensen, C C ; Lucius, J L ; Oblow, E M ; and Swift, T E. 1985 Machine Intelligence for Robotics Applications In Proceedings of the 1985 Conference for Intelligent Machines, 47–57

Weisbin, C. R. et al 1982. *Advances in Nuclear Science and Technology 14*, eds Levine and Becker New York: Plenum Press

Worley, B. A., and Wright, R. Q. 1985 An Automated Procedure for Calculating Time Dependent Sensitivities in ORIGEN-2 Technical Report ORNL/TM-9771, Oak Ridge National Lab