CONF-890522--2

CONF-890522--2

DE89 001525

AUTONOMOUS DISCOVERY AND LEARNING BY A MOBILE ROBOT IN UNSTRUCTURED ENVIRONMENTS*

F. G. Pin, G. de Saussure, P. F. Spelt,† D. L. Barnett,‡ S. M. Killough,** and C. R. Weisbin

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6364

Invited Paper: Submitted to 1989 IEEE International Conference on Robotics and Automation, May 14-19, 1989, Scottsdale, Arizona

---

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

† On leave from Psychology Department, Wabash College, Crawfordsville, IN 47933

‡ Computing and Telecommunications Division, Martin Marietta Energy Systems, Inc., P.O. Box 2008, Oak Ridge, TN 37831-6364

** Instrumentation and Controls Division, Martin Marietta Energy Systems, Inc., P.O. Box 2008, Oak Ridge, TN 37831-6007

CONF-890522--2

AUTONOMOUS DISCOVERY AND LEARNING BY A MOBILE ROBOT IN UNSTRUCTURED ENVIRONMENTS

François G. Pin, Gerard de Saussure, Philip F. Spelt,
D. L. Barnett, S. M. Killough, and C. R. Weisbin

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
P.O. Box 2008
Building 6025, MS-6364
Oak Ridge, TN 37831-6364

ABSTRACT

This paper presents recent research activities at the Center for Engineering Systems Advanced Research (CESAR) in the area of autonomous discovery and learning of emergency and maintenance tasks in unstructured environments by a mobile robot. The methodologies for learning basic operating principles of control devices, and for using the acquired knowledge to solve new problems with conditions not encountered before are presented. The algorithms necessary for the robot to discover problem-solving sequences of actions, through experimentation with the environment, in the two cases of immediate feedback and delayed feedback are described. The inferencing schemes allowing the robot to classify the information acquired from a reduced set of examples and to generalize its knowledge to a much wider problem-solving domain are also provided. A demonstration of the successful implementation of the algorithms on our HERMIES-IIB autonomous robot is then presented. The demonstration is based on a scenario in which the autonomous robot learns the functioning of a process control panel during a training session on a simulator. The robot is then asked to navigate through an unknown dynamic environment to search for, find, and dock at the real control panel, read and understand the status of the panel's meters and dials, and manipulate the panel devices to solve an emergency maintenance problem, possibly never encountered before. Conclusions are drawn concerning the applicability of the methodologies to more general classes of problems and implications for future work on autonomous discovery and learning of complex tasks by mobile robots are discussed.

INTRODUCTION

The Center for Engineering Systems Advanced Research (CESAR) was founded at Oak Ridge National Laboratory by the Department of Energy's Office of Energy Research/Division of Engineering to conduct basic research in the area of intelligent machines. Within this framework, CESAR has undertaken several research activities related to navigation, surveillance, and manipulation by autonomous mobile robots in unstructured dynamic environments. Key to successful performance of autonomous robots in a-priori unknown and unstructured environments are their coupled capabilities of acquiring appropriate information concerning their environment and of rapidly processing this information to support flexible decision making and task execution. While the knowledge concerning the environment dynamics must be acquired through sensors and real-time sensor data interpretation techniques, most of the meta-knowledge necessary for decision-making and coping with unpredictable problems and events can be acquired through programming by humans or through autonomous learning. When dealing with increasingly wide task domains in complex and realistic environments, the programming option rapidly becomes impractical and overwhelming for the knowledge engineer and programmer. A self-didactic capability thus appears as an essential component of an autonomous robot in complex unstructured environments.

This paper presents the results of recent investigations in this area as implemented on our HERMIES-IIB autonomous mobile robot. The next section describes the algorithms developed to allow the robot to discover, through experimentation with its environment, the functioning of control devices as well as the correct sequences of actions necessary to achieve a predetermined goal. Both cases of immediate feedback and delayed feedback from the environment during this discovery phase are discussed. The algorithms allowing classification of the acquired knowledge and generalization to a larger task domain are then presented. The demonstration illustrating the successful implementation of these algorithms on our HERMIES-IIB autonomous mobile robot is presented in Section 3. Conclusions and implications for future work are discussed in Section 4.

DISCOVERY AND LEARNING METHODOLOGIES

The learning methodologies presented here were developed within the context of a paradigm encompassing long term research goals in the area of intelligent machines for surveillance, maintenance, repair and emergency handling in complex, hazardous or unstructured environments. In this paradigm, mobile robots will be responsible for performing routine tasks autonomously and, as necessary, will be called upon to handle repair and emergency operations. The robots will be provided with learning capabilities which will allow them to discover novelties or changes in their environment characteristics, train themselves to perform tasks better or to learn new tasks, and use this acquired experience and knowledge to augment their problem-solving abilities. What is presented below represents the very initial steps in these directions with emphasis being given to integration, implementation, and feasibility demonstration of the progress simultaneously with the methods development. Our initial investigations focussed on discovery and learning by an autonomous robot of the functioning of a control system. Our approach involves three major steps of learning: an initial trial-and-error learning phase in which the robot discovers various actions and response sequences to operate a given system which has controlling properties on the robot environment; a second phase in which observed attributes and their values are used in an inference scheme to classify the control system states into categories; and, finally a hypothesis-generation phase in which, based on previously solved problems and inferred categories, the robot selects a sequence of actions to try to solve a new problem presented by the control system. It should be noted that these three phases occur cyclically as the robot acquires or infers new information about the functioning of the control system.

Discovery Phase

In this phase, the robot is assumed to have a knowledge of the basic actions it can perform with objects; that is, the robot knows the list of "primitive" actions which it is allowed to exercise on objects in the environment, such as grasp an object, pull or push an object, move an object right, left, up or down, observe changes in the status of an object (e.g. lamp on or off), etc. In the initial discovery, the robot uses its vision system to take pictures of the system with which it is to work. The gray-value images are transformed into binary images using thresholding and "patches" of contiguous pixels with similar values are identified. Each patch is labeled as a potential object, on which the robot attempts to exercise its manipulation primitives. The successful manipulation actions are recorded as paired items in a list of indexed responses: R_i (object j, action k). The robot has thus established the list of potential features F_i (the set of objects) and the list of potential responses R_i that are pertinent for discovery of the functioning of the investigated system.

The robot's vision system's primitives can be used to establish the status (e.g., lamp on or off) or changes in the status (e.g., meter needle moved, handle moved) of the feedback devices by comparing new pictures of the devices with the original pictures. For each feature F_i, the robot considers a given list of possible feedback B_j which correspond to the characteristics that can be observed with the available sensor suite. A list of potential attributes A_k (F_i, B_j) for the studied system is then generated. Each attribute is given a range of possible value according to the type of feature it includes, and to the capabilities of the sensor-data interpretation routines in the related primitives. For example, in the current implementation on HERMIES-IIB, two types of feedback, illumination and position, are considered by the robot with respective discrete value ranges of (On, Off), and (Left, Middle, Right). At the end of this process, the robot has acquired a means of establishing the status (or state) of the system as the vector $V(v_k)$ of indexed attribute values v_k, detecting feedback from the system by observing changes in the attribute values, and experimenting with the system by using the responses Ri.

The robot continues its learning activities by experimenting with the system. To do so, it is asked to reach a goal or a series of goals, specified as a required change in the value of one (or several) attribute of the system. Alternatively, the robot can investigate changes in the attribute values, one at a time, as a series of self-set goals. The objective of the training is to establish, for a given goal G_k, the appropriate sequence of responses (the sequence containing only necessary and sufficient responses) $S_{ik} = \{R_l, R_m, R_n...\}$ which relates an initial system state V_i to the goal G_k:

$$\mathrm{Vi}(v_j) \xrightarrow[S_{ik}]{\{R_l, R_m, R_n...\}} G_k \; .$$

The functioning of the system investigated is assumed to be complex, but realistic, in the sense that the sequences S_{ik} contain a non-trivial number of responses and may vary significantly with the goals and initial states considered, according to the physical law of the system they represent. However, in this initial development, we have assumed that any response R_i occurs at most once in a correct sequence S_{ik} and that subsequently attempted responses in that sequence do not cancel previously attempted responses (i.e., $R_j \not\Leftrightarrow \overline{R_i}$, where the overbar denotes the complement of a response).

We will refer to the couple (V_i, G_k) as an exemplar for the learning process and to S_{ik} as the associated response sequence for this exemplar. Three algorithms were developed to support the autonomous sequence discovery activity corresponding to three possible characteristic behaviors of the system to be learned: (1) Immediate feedback is provided after each correct response, (2) feedback is delayed until all correct responses have been included in S_{ik} in a non-ordered fashion and, (3) feedback is delayed until the correct responses, appropriately sequenced, have been included in S_{ik}. To allow the robot to rapidly experience and learn many exemplars and response sequences, we have selected here to deal with types of systems which can easily and readily be reset to any given initial state (e.g., sets of light switches as immediate feedback systems, multiple converging and diverging hydraulic pipes and valves for delayed feedback systems with non-ordered sequences, safe locks as delayed feedback systems requiring ordered response sequences).

Immediate Feedback. Considering the first exemplar (V_1, G_1), the robot sequentially attempts the responses R_i of its response list until feedback is provided indicating that the first correct response for the exemplar has been found. The response is stored in memory and the robot repeats this process until the entire sequence S_{11} has been found for the exemplar. The triplet (V_1, G_1, S_{11}) is stored in long-term memory and a new initial state of the system is requested by the robot which repeats the discovery and memorizing process for this new exemplar. If the robot determines that the state of the system in an exemplar which is currently presented has been previously treated, it recalls the corresponding S_{ik} which was stored and attempts it. In the following sections we will show that the sequence search process which is initially unguided for the first exemplars, can become guided and considerably sped up for following exemplars using classification and inferencing schemes.

Delayed Feedback for Non Ordered Sequences. When the correct sequence necessary to reach the goal does not need to be performed in an ordered fashion and the feedback is delayed until the goal is attained, the robot attempts the possible responses of its list according to their ranking index (R_1, R_2, R_3, ...). Since for non-ordered seqences feedback is guaranteed to occur when or before all responses have been attempted, the robot will reach the goal during the first pass through the response list. Because the objective here, however, is to discover the sequence containing only the necessary responses, the robot proceeds according to the following algorithms: the system is reset to this exemplar original state. Since the last response attempted prior to receiving feedback, say R_l, belongs to the correct sequence, it is attempted first. The responses next attempted are again those of the list according to their ranking order. In this second attempt, therefore, the attempted sequence proceeds as (R_l, R_1, R_2, R_3 ...), until feedback is obtained, say following R_m. The system is reset to this exemplar original state, and the sequence now attempted proceeds as (R_l, R_m, R_1, R_2, R_3 ...). The process repeats until all responses in the last attempted sequence have been shown to precede a feedback. This sequence contains only necessary and sufficient responses and is stored in memory with the exemplar characteristics V_i, and G_k. The robot then moves on to a new exemplar.

Delayed Feedback for Ordered Sequences. When the responses in the correct sequence need to be performed in an ordered fashion and feedback is delayed until the goal is reached, the search space for the robot grows combinatorially with the number of possible responses. What was attempted here was to decrease to a minimum the number of sequences which the robot needs to try prior to receiving a feedback. (Since in this case feedback is not guaranteed for a sequence containing all responses R_i). The algorithm thus proceeds as follows: If n is the total number of available responses, the robot generates lists containing 2n-1 responses. The first n responses of each list are the n possible responses Ri, arranged by increasing index in the first list, and rearranged according to the rule of factorial divisibility in the following lists. The last n-1 responses in each list are a duplication of the first n-1 responses in the list. If n=5, for example, the first list which will be attempted by the robot is: (R_1, R_2, R_3, R_4, R_5, R_1,

R_2, R_3, R_4), the second list: (R_1, R_2, R_3, R_5, R_4, R_1, R_2, R_3, R_5), etc. Following each unsuccessful list, the system is reset to its exemplar initial state prior to the robot attempting the next list. When feedback is obtained, the correct ordered sequence is included within the currently attempted list, and the robot proceeds by elimination to determine all necessary and sufficient responses in the list. The solution (V_i, G_k, S_{ik}) is stored in memory and the robot moves on to a new exemplar.

Classification Phase

The classification phase of the learning process presented here draws from the approach of "instance-to-class" generalization[1] based on examples which the robot experienced and solved in the discovery phase. The robot first groups all experienced exemplars into categories. Each category consists of the set of exemplars for which an identical goal was reached using the same response sequence. The robot uses a set of discrimators (e.g. "i=i"; equal values for attribute Ai, "i>j"; values for attribute A_i greater than values for attribute A_j, ...) to identify true relationships between the attribute values of all exemplars in each category and to classify the categories uniquely using conjunctions of those relationships. This classification process is repeated everytime a new exemplar is added to a category and does not verify the conjunction of relationships for that category.

Hypotheses Generation Phase

In this last phase of the learning process, the robot uses the knowledge it has acquired in the previous phases to generate hypotheses of solution sequences for new exemplars. The robot first determines the initial attribute value vector $V_i(v_k)$ for the new exemplar. It then identifies the categories with the same goal G_k as the exemplar and attempts to verify the relationships of each category with the exemplar attribute values. If a perfect match is obtained for a category, the response sequence of that category will be the preferred attempted sequence. If no perfect match exists, a preference factor is calculated for each category. The preference factor is the sum of the "discriminating weights" of the relationships verified by the new exemplar in each category. Sequences are attempted in the order defined by the calculated preference factors. If none of the attempted sequences turn out to be successful, the robot returns to the original discovery phase to derive the correct sequence for this exemplar.

PROOF OF PRINCIPLE DEMONSTRATION

In order to focus our research activities, prove the correctness of our general approach, test and verify the basic methodologies and algorithms, and better identify areas for further investigation, several experimental scenarios were developed and some associated demonstrations were implemented on our HERMIES autonomous mobile robots. In this section, we present one such demonstration which was conducted to test HERMIES-IIB's capabilities in unsupervised learning, autonomous navigation in unstructured and dynamic environments, handling of contingencies, goal recognition, reading and understanding of complex control devices, vision-guided manipulation, and innovative problem-solving based on prior learning.

General Description of HERMIES-IIB

HERMIES-IIB (Hostile Environment Robotic Machine Intelligence Experiment Series IIB) is the latest member in a series of progressively more capable and sophisticated robots. The first HERMIES robots provided valuable experience in planning, world modeling, and communication, but the intensive computations and high-level decision making in these experiments were performed off board in computers linked by radio with the robot.[2–4] HERMIES-IIB stresses computational autonomy; hence, the need for powerful on-board computing capabilities.[5] HERMIES-IIB manipulators are too primitive to perform tasks requiring significant strength or high precision; however, their capabilities were sufficient to support autonomous vision-guided manipulation in the context of simple maintenance tasks at a control panel, and consequently demonstrate some of the learning algorithms presented here.

HERMIES-IIB is a self-powered robot sytem consisting of a wheel-driven chassis, dual manipulator arms, on-board distributed concurrent processors, and a directionally controlled sensor platform (see Fig. 1). The robot is propelled by a dual set of independent wheels having common axle alignment and driven by separate DC gear head motors powered by pulse-width modulated servo amplifiers.

The batteries and drive chassis components are located in the robot body's rectangular lower part. An IBM 7532 and above it, a VME rack are mounted in the robot frame's trapezoid-shaped upper part just above the drive chassis. A dual-arm manipulator torso is mounted above the IBM 7532, forward of the VME rack and on the outside of the robot's "skin." The manipulators are five-degree-of-freedom units manufactured by Zenith/Heathkit and used on the Hero home robot.

The sonar sensing sytem, an array of Polaroid range finders, consists of 25 individual transceivers, arranged in six 2×2 matrix clusters. Twenty-four of these sonar transducers are mounted and operated as phased-array range-finding elements to reduce the effective sonar beam from approximately 30 degrees of the individual transducers to about 12 degrees for the phased-array clusters. Five of these clusters are mounted in a ring on the periphery of the rotatable robot head; the sixth cluster is mounted on a tiltable platform attached to the head. The remaining sonar transceiver (not shown on Fig. 1) is located on the front side of the robot, near the mid-section and between the manipulators. It serves as a collision-avoidance sensor during navigation.

A video data acquisition system forms the heart of the machine vision hardware. Currently, the system uses two Sony CCD black-and-white cameras, one of them equipped with a wide-angle lens. Frame acquisition is via a Maxvideo system from Datacube.

The IBM 7532, an industrial version of the IBM AT, provides mass storage with a 20M-byte hard disk and a 1.2M-byte floppy disk as well as 2M-byte RAM. Six AT-style and two XT-type expansion slots are available on the computer's back plane. Four of the eight slots are used for I/O devices and memory expansion, while the other four slots are available for NCUBE parallel processing boards, each containing four processor nodes—for a total of sixteen nodes—arranged in an hypercube configuration. Communication between the IBM 7532 and the 20-slot double-high industrial VME rack is by an 8-megabaud parallel link made by the Bit-3 computer company, with a transfer rate of 1M-byte per second.

Computer programs controlling the robot's behavior are mostly written in C and can be organized into four classes: the HERMIES primitives, the expert systems and associated navigation and learning routines, the image analysis routines, and the control and integration programs that reside on the NCUBE host. The expert systems may be executed from either MS-DOS or AXIS while all of the image analysis routines have been developed in C for execution on the NCUBE concurrent processing computer.

Demonstration Results

Figure 2 illustrates the layout and actual events in the experimental demonstration. Initially, the robot is at its "idle" position at point A. At the training station, a robot's brain emulator is in communication with a control panel simulator. Note that several emulators could be trained simultaneously on several plant control device simulators depending on the variety of surveillance, maintenance, repair or emergency-type tasks which the robot may be requested to perform or, alternatively, representing the different types of process control devices actually existing in the plant. In this experiment, HERMIES-IIB is to learn the functioning of a plant process control panel. The control panel includes six control devices, four buttons and two levers, (see Fig. 1) and nine feedback devices, two meters, four LEDs in the buttons, the two levers, and a danger light. Due to the current limitations on its manipulation capabilities, the robot is given the elemental knowledge of the basic actions possible with each device (e.g., buttons can be pushed, levers can be grasped and moved right or left, meters can be read, lamps can light up to indicate a correct action, etc.). An initial state i of the panel is defined by the vector $V_i(v_k)$, the components of which are the attribute values defining the sytem (in this demonstration, left meter needle position and upper lever position are two examples of attributes of the system, with possible respective values of low, medium or high and left, center, or right). The overall control scheme of the panel is programmed on PCs located within the actual panel and in the simulator. This control scheme, unknown to the robot, requires that, for each initial state of the panel, a specific sequence of actions on the control devices be performed to achieve the predetermined goal of "turning-off the danger light." During the training session on the simulator, the robot's brain emulator is exposed to a set of initial states (the training set) which the panel may take. The role of the expert system brain is to discover, by trial and error experimentation with the control panel, the correct sequence of responses which achieves the goal for each exemplar in the training set and infer categories from these discoveries. In this trial and error phase, we assume that immediate feedback about the correctness of actions is provided to the robot through the nine feedback devices. When the training session is completed, the acquired knowledge is stored as "long term memory" on a PC diskette and is transferred, when necessary, to the robot.

An emergency situation is then simulated by turning on the danger light on the real panel. The location of the panel, as well as that of all other objects in the room, is unknown to the robot. The panel control devices are also arbitrarily moved to set the panel in an initial state possibly never dealt with before by the robot (i.e., not included in the training set). The relative coordinates of a subgoal location B are sent to the robot via a RF-link, and a "GO" statement is issued. HERMIES-IIB's task is to autonomously navigate from A to B avoiding or removing several types of static and moving obstacles. From location B, HERMIES-IIB is to find the control panel, move up to the panel and manipulate the control devices in the exactly correct sequence necessary to turn-off the danger light. It is assumed that there are no obstacles between location B and the panel.

HERMIES-IIB starts by making a wide angle sonar scan of the environment and planning a collision free path to the reachable point closest to B. As the robot moves toward its destination a sonar scans the area ahead of the robot. If the sonar detects an unexpected obstacle in its path, it stops within 2 feet, diagnoses the nature of the obstacle and takes appropriate action. A conflict resolution and error propagation scheme[6] is used to combine data from the sonar and vision in this path-planning and collision-avoidance algorithm. Several unexpected events are generated to illustrate the robot's capabilities for detection, contingency handling, and dynamic replanning. At point S, HERMIES-IIB stops for an unexpected obstacle, small enough to be picked up and moved out of the way. On its way to point A_1, HERMIES-IIB faces a large moving obstacle obstructing its path, and is forced to backtrack and globally replan its navigation. At point M, a pedestrian forces HERMIES-IIB to stop but quicly moves out of the way, allowing the robot to pursue its original path. When HERMIES-IIB has reached location B, the vision system searches for the panel as a black box with aspect ratio of 2 to 3. Upon discovering a potential candidate for its goal, the robot initiates its navigation toward this potential target. Once the robot is close enough to the panel for the meters to be recognized by the vision system, the panel is identified as the desired goal, and the robot progressively moves to a position allowing it to read and manipulate the panel control devices.

The vision system is first used to determine the location of all devices on the panel face and evaluate their state (attribute value). In the case of the analog meters, for example, the region of the binary image identified as a meter is searched to find groups of pixels that form lines. Since the meters have two needles (one is a preset or limit needle) the images are searched to find the most prominent pair of lines. A Hough transform is used to convert the needle position data from cartesian to polar (r, θ) values and a corresponding attribute value is assigned.

The knowledge learned during the training session is subsequently used to infer the correct sequence of actions on the control devices to turn off the danger light, and the vision-guided manipulation is used to perform these manipulation tasks. For this particular demonstration we showed[7,8] that only 14 exemplars in the training set were necessary before the emulator "brain" had acquired enough knowledge about the panel control scheme to solve any new problem (initial states not previously encountered) with no error in the sequence of control actions. In Refs. 7 and 8, we also showed that the practicality of autonomous robots discovering, learning, and memorizing new tasks was great indeed. For problems based on relatively simple devices such as the control panel presented here, all humans who were involved in the comparison study (our CESAR laboratory cooperative students) exhibited a combination of high error rates, repeated errors, false recall or non-memorization, and no consistent improvement of performance with increasing numbers of exemplars. HERMIES-IIB, on the other hand, rapidly derived the necessary knowledge and understanding of the control system to achieve a fully error-free performance on any new problems. Of course, HERMIES-IIB did not forget any of the rules it learned or exemplars it experienced.

DISCUSSION AND CONCLUSIONS

The successful implementation of the demonstration just described validates some of the concepts of autonomous robot learning for surveillance, maintenance, repair or emergency-type of activities in unstructured environments. In particular, the feasibility for an autonomous robot to discover and memorize feedback features and response devices, and to infer from these discoveries some characteristic behavior of a complex control system has been demonstrated. We have also shown the practicality of using unsupervised learning to increase the robot's knowledge in a wide task domain and to improve its problem-solving capabilities to cope effectively (with no error) with new problems or situations, without increasing the programming burden of the knowledge engineer. Finally, the HERMIES-IIB demonstration presented here clearly suggests areas requiring further research and developments. For example, ours and others' recent investigations have shown that the efficiency of the coupled discovery and inferencing processes can be dramatically improved by taking into account the information corresponding to negative examples (incorrect actions). The classification and hypotheses generation phases leading to "no error" performances can also be improved and sped-up through utilization of dynamically adaptive discriminators and weighting functions. These developments, in turn, would support extensions of the unsupervised learning methodologies to include self-generation of exemplars. Clearly, further research is needed in the areas of temporal and approximate reasoning methodologies to support more advanced and flexible learning strategies. Investigations in these directions as well as research on novel methodologies to allow the robot to learn its own capabilities (e.g. primitives) by observation of humans are pursued in ongoing projects at CESAR.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the valuable assistance and contribution of the entire CESAR team in designing and implementing HERMIES-IIB and the experimental demonstration, and in particular Eric Lyness, Chad McGougan, Mike Pav and Glenda Oliver in the learning area, and Michelle Clinard, Mike Kedl and Judd Jones in the image-processing area. Research for this article was sponsored by the Engineering Research Program of the U.S. Department of Energy's Office of Basic Energy Sciences, under contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

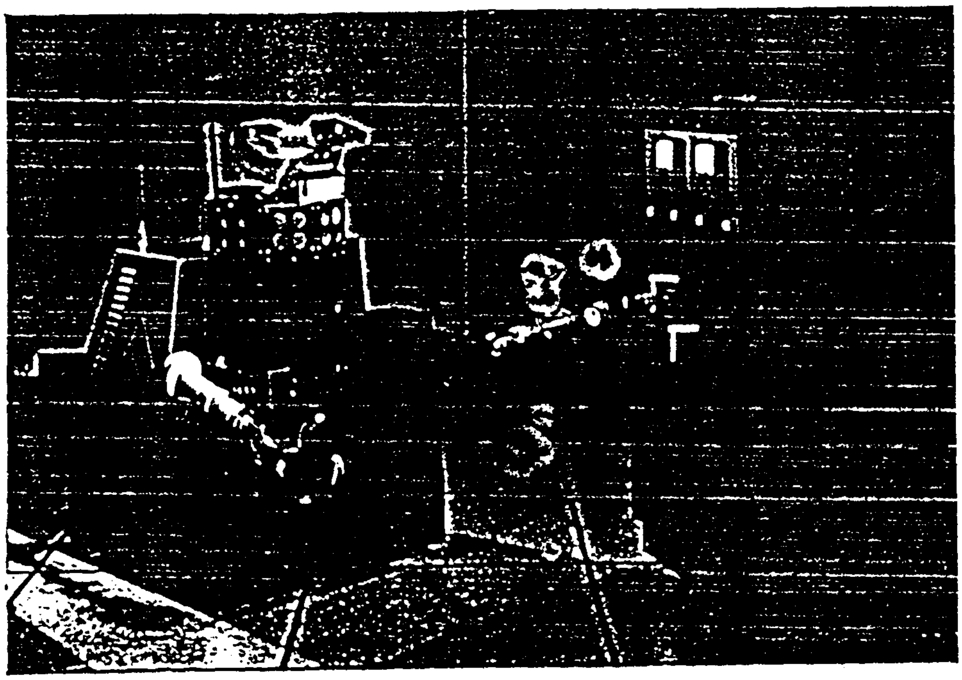

Fig. 1. HERMIES-IIB moving a lever of the control panel.

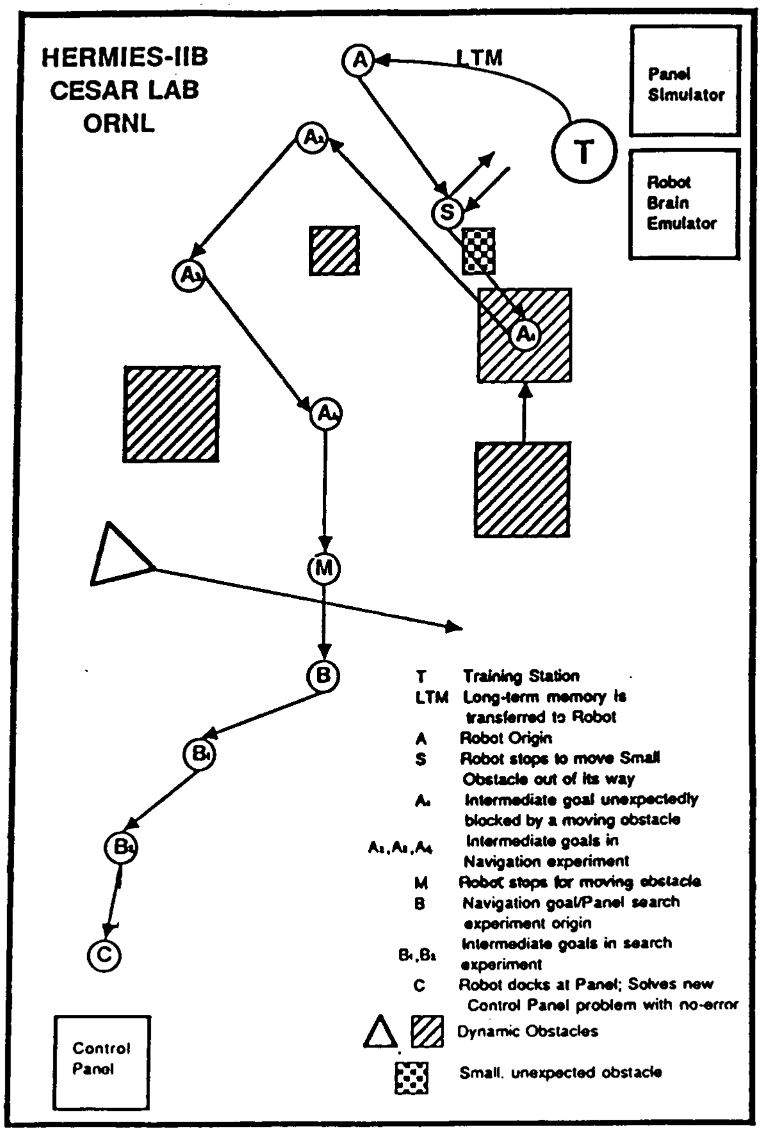

Fig. 2. A layout of the experimental area for HERMIES-IIB. The robot navigates from point A to point B, stopping at intermediate goals A_2, A_3, and A_4 and responding to dynamic obstacles at points A_1, S and M. The robot moves from point B to the control panel, stopping at intermediate goals B_1 and B_2. At point C the robot is close enough to read and manipulate the panel control devices.

REFERENCES

[1] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Machine Learning, an Artificial Intelligence Approach, Vol. I, Tioga Publishing Company, Palo Alto, Calif., 1983; and Vol. II., Morgan Kaufmann Publishers, Inc., Los Altos, Calif., 1986.

[2] C. R. Weisbin et al., "HERMIES-I: A Mobile Robot for Navigation and Manipulation Experiments," *Proc. Robots 9 Conf.,* Vol. 1, Detroit, Mich., June 1985, pp. 1-41.

[3] W. R. Hamel et al., "Autonomous Robots for Hazardous and Unstructured Environments," *Proc. Robots 10 Conf.,* Vol. 5, Chicago, Ill., April 1986, pp. 9-20.

[4] C. R. Weisbin, G. de Saussure, and D. W. Kammer, "Self-Controlled: A Real-Time Expert System for an Autonomous Mobile Robot," *Computers in Mechanical Engineering,* September 1986, pp. 12-19.

[5] B. L. Burks, G. de Saussure, C. R. Weisbin, J. P. Jones, and W. R. Hamel, "Autonomous Navigation, Exploration, and Recognition Using the HERMIES-IIB Robot," *IEEE Expert,* Winter 1987, pp. 18-27.

[6] M. Beckerman and E. M. Oblow, "Treatment of Systematic Errors in the Processing of Wide Angle Sonar Sensor Data for Robotic Navigation," ORNL/TM-10763, 1988. Available from National Technical Information Service, Springfield, Va., 22161

[7] P. F. Spelt, G. de Saussure, E. Lyness, F. G. Pin, and C. R. Weisbin, "Learning by an Autonomous Robot at a Process Control Panel," *IEEE Expert,* August 1988 (in press).

[8] P. F. Spelt, E. Lyness, and G. de Saussure, "Concurrent Simulation: Development and Training of a Learning Expert System for an Autonomous Robot," submitted to *Simulation.*