N90-29068

RICE-OBOT I: AN INTELLIGENT AUTONOMOUS MOBILE ROBOT*

R. deFigueiredo, L. Ciscon, D. Berberian

Department of Electrical and Computer Engineering
Rice University
Houston, Texas 77251-1892

Abstract

The *Rice-obot I* is the first in a series of Intelligent Autonomous Mobile Robots (IAMRs) being developed at Rice University's Cooperative Inteligent Mobile Robots (CIMR) lab. The Rice-obot I is mainly designed to be a testbed for various robotic and AI techniques, and a platform for developing intelligent control systems for exploratory robots. In this paper, we present the need for a generalized environment capable of combining all of the control, sensory and knowledge systems of an IAMR. We introduce *Lisp-Nodes* as such a system, and we develop the basic concepts of nodes, messages and classes. Furthermore, we show how the control system of the Rice-obot I is implemented as sub-systems in Lisp-Nodes.

1. Introduction

The *Rice-obot I* is the first in a series of Intelligent Autonomous Mobile Robots (IAMRs) being developed at Rice University's Cooperative Inteligent Mobile Robots (CIMR) lab. Rice University has developed strong relationships with several groups at the Johnson Space Center, and thus the mobile robotics program has emphasized technologies applicable to Space Robotics and exploratory roving vehicles. The Rice-obot I is mainly designed to be a test platform on which various control, hardware and AI concepts can be easily inserted and tested. Also, we are interested in developing onboard intelligent command systems required for autonomous exploratory robots. Furthermore, we want the robot to be able to perform repairs and maintenance on objects, as a space robot might do to a satellite. To achieve these goals, Rice-obot I was designed to include several advanced capabilities. Among these, the most important are:

A. To be totally autonomous. This means that all of the computing is onboard, the system uses radio links to the basestation (thus no cables), and it has an onboard power system.

**Supported by the NASA Grant NAG-9-208, the State of Texas Grant TATP 2982, and a grant from Texas Instruments*

B. To be able to navigate in an unstructured environment. By unstructured we mean that an incomplete (or nonexistent) map of the area is given. This implies that it should possess map-making capabilities.

C. To incorporate advanced sensors, including a laser 3-D mapper and stereo vision.

D. To communicate with users at the basestation at a high level; this maximizes the amount of information transmitted and minimizes the data bandwidth of the link.

E. It should incorporate dual dexterous arms and a set of tools for manipulation and assembly/disassembly of objects.

and, most importantly,

F. To be highly "modular" both in hardware and software, so that various AI systems can be integrated and interchanged relatively easily. This enables us to use already developed, high-level AI techniqes from other sources with very little modification.

A fair amount of research has been done on the various areas associated with mobile robotics, including path planning (Weisbin[3], Thorpe[13], Meng[5,6]), sensor integration (Hirzinger[8], Harmon[12], and Thorpe[13]), and various people on topics from obstacle avoidance to advanced control algorithms. It is important to note that developers have relied on a large variety of AI systems as the basis for their intelligence including expert systems, neural nets, blackboard systems, and semantic nets. However, it has become apparent that no single knowledge representation scheme is sufficient for dealing with the myriad of different tasks, situations, and objects encountered by an IAMR. In that light, various researchers have developed different schemes for integrating the various AI and control system ([2], [4], [10], [16], [18]). It is even more complicated to make such systems "modular" (although Brooks' system is a good example). We came to the conclusion that we needed a fundamental, underlying environment in which we could install several different AI techniques concurrently, and that could easily handle the multitasking and multiprocessors in an IAMR. Thus, we have developed the Lisp-Nodes environment. In Lisp-Nodes, all of the knowledge and control of the robot is embodied as systems of nodes. The node network is divided into major and minor sub-systems which communicate with one another through a high-level protocol. With this system, we can replace pieces of the overall system without affecting the remaining parts, and it can expand in an organized and controlled manner.

2. Hardware Overview

The robot includes an intelligent base, two robotic arms, several on-board computers, a stereo vision system, a 3-D laser mapper and an Ultrasonic system (see following figures). The robot is mounted on a commercially available mobile base which contains its own processor and power. The rest of the components mount onto a tubular steel frame which, in turn, bolts to the frame of the base. A standard VME bus is used. To augment this, there is a board-to board bus connecting the vision equipment for high-speed picture transfer. The main processor is a 68030 based computer made by LYNX Systems. The processor runs a real-time version of UNIX especially designed for control of devices such as robotic arms. The motors on the robot, including those on both arms, the Z-tables and the pan/tilt/aim of the cameras, are controlled (through the UNIX processor) by two servo controller cards on the VME bus. The two robots arms each have five degrees of freedom; they have the advantage over other arms that they are very light (15 lbs. apiece), yet they have high accuracy (approx. 1/5 mm. with modifications we have made) and a relatively high payload (5 lbs.). The arms mount on the Z-tables, thus adding an additional degree of freedom. The instrument pod mounts on a pan/tilt unit connected on the top of the robot. The four major items on the pod consist of two cameras, a laser ranging system, and one ultrasonic ranging system.The vision system is augmented by real-time processing boards that perform low-level processing of the image, separate the image into "blobs", and pass the information to the UNIX processor. The robot communicates via radio frequency RS232 to a windowing workstation, which acts as the user interface. A special LISP processor, the TI Explorer II, is also mounted on-board the robot to handle most of the high-level Artificial Intelligence tasks. The Explorer contains a multiprocessor Odyssey board for general high-speed signal processing. The Odyssey acquires high-speed vision data through an extension to the high-speed vision bus. Most of the low-level control and command of the robot occurs in the UNIX processor. For the most part, the two computers communicate using an Ethernet link. This on board network is especially useful during debugging, because it can be easily connected to the major on-campus computer network.

3. Lisp-Nodes Introduction

Lisp-Nodes is based upon *nodes*. A node can embody many different concepts. It could be a single if/then expression in an Expert System, or a cell in a Neural Net. Alternately, one node could embody all of the low-level vision processing for a 3-D vision system. The difference between the former two examples and the latter can be viewed as the amount of coupling; that is, the former is loosely coupled, the latter is tightly coupled. Both types of coupling are needed on a mobile robot. Some sub-systems such as the kinematics of the arm, work best as a tightly coupled routine. Others, like the global knowledge base, need a loosely coupled network of nodes, while the vision system needs some of both. Lisp-Nodes allows all levels of coupling within a uniform environment. It is important to note that Lisp-Node's proprioceptive knowledge is mostly at the node and interconnection level; i.e. the system learns by creating and destroying nodes and their connections. Thus the

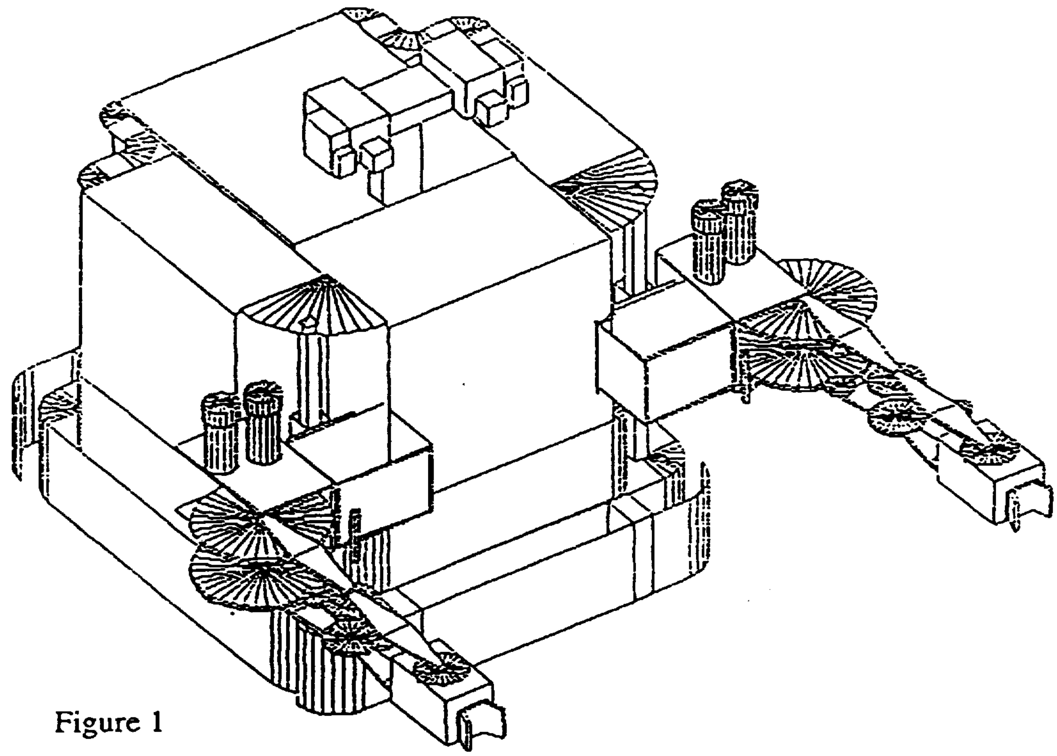

Figure 1

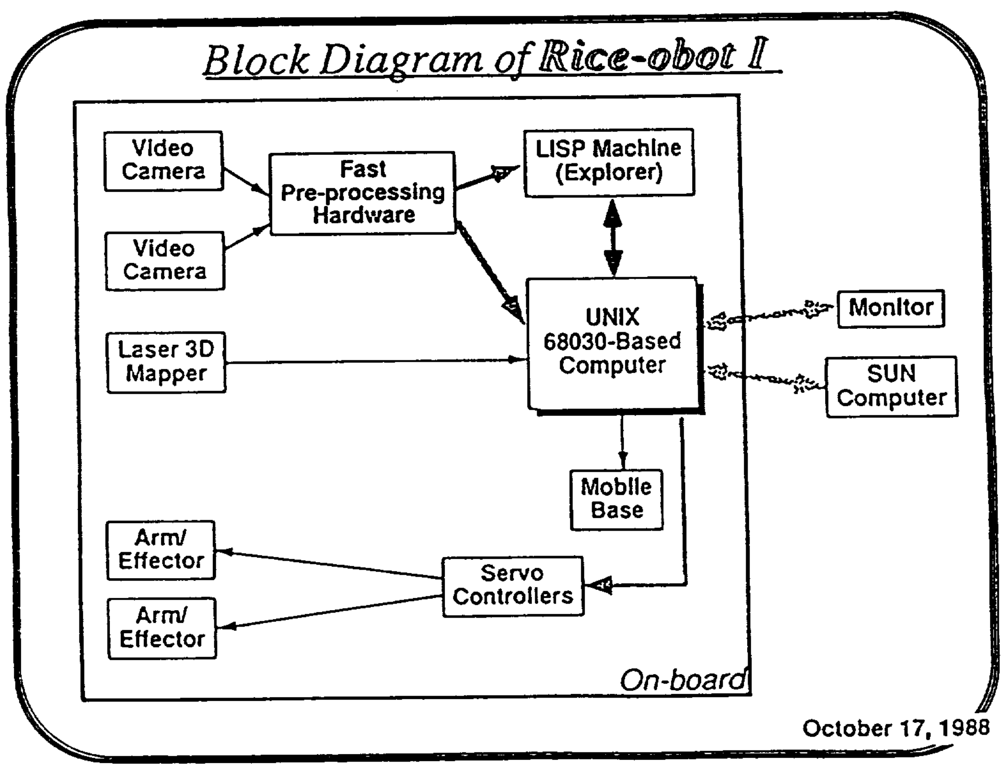

Figure 2

more tightly coupled the sub-system, the harder it is to be taught new techniques.

Another important aspect of nodes is that each node runs concurrently; so different groups of nodes can be processing different information simultaneously. This is critical since many different systems on the robot need to process information at the same time. For example, some nodes could be processing vision data while others plan a path.

Nodes communicate with each other using messages. Messages in Lisp-Nodes act just like messages in a computer network; they can send packets of information between any two nodes. The message packets can contain any information ranging from a string to a complex LISP expression. Each node can respond to many messages, and messages can be added and deleted from its capabilities. By modifying what messages are sent and received, the interconnection of nodes is also changed.

Another important capability of LispNodes is its ability to group nodes together; a group (or *class*) could be all nodes with a specific property, or all nodes pertaining to a particular subsystem. One example usage of classes is for Minsky's frame concept. Grouping enables an arbitrary node to communicate with a whole class of nodes without necessarily knowing which nodes are in the class; thus classes act very much like blackboards. Classes are formed by creating a *class-node* that processes and passes messages to all nodes within the group. Figure 3 shows a class node distributing a message.

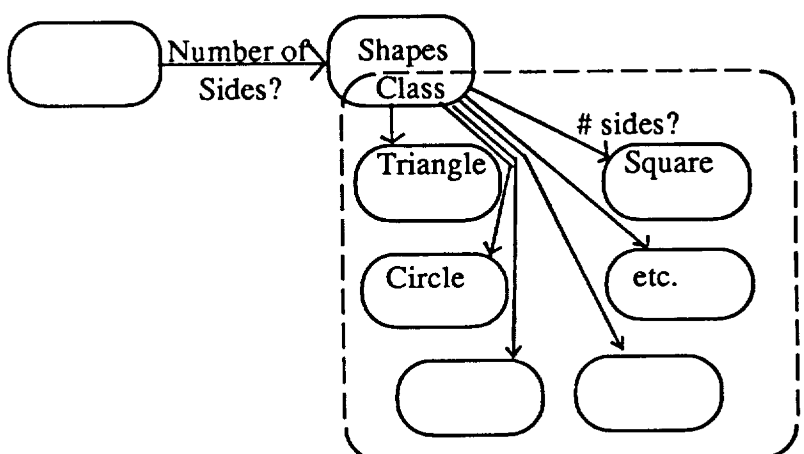

Figure 3

4. Subsystem Organization

As has been mentioned, all of the sensor, control, and knowledge representation systems on the robot are connected to the Lisp-Nodes environment. This, in itself, does not impose any order on the system. The control system organized on top of the nodes has to be structured enough to enable the robot to repeatably and predictably perform complicated functions, yet flexible enough to be easily modifiable. Thus we chose a knowledge-base driven system consisting, at the top level, of a very few distinct major subsystems which communicate and interact using a fairly simple set of rules. These subsystems are implemented using the classes in Lisp-Nodes, and thus act like mini-blackboards. These main subsystems are further divided into sub-subsystems as necessary. The sub-subsystems act the same as the main systems, except for two main differences: a) in these sub-subsystems, classes can overlap (i.e. a node can be in several different sub-subsystems simultaneously), and b) they can be created and destroyed. Whenever a node or class is created/destroyed, the classes it belongs to are informed of the change; these classes, in turn, can inform their parent-classes of the change, which inform the next higher level, etc. To ensure consistency, major subsystems (and their subsystems) route all requests for creating/destroying nodes that are contained in a different major subsystem through that system's class. The major subsystems that are being implemented on the Rice-obot are as follows:

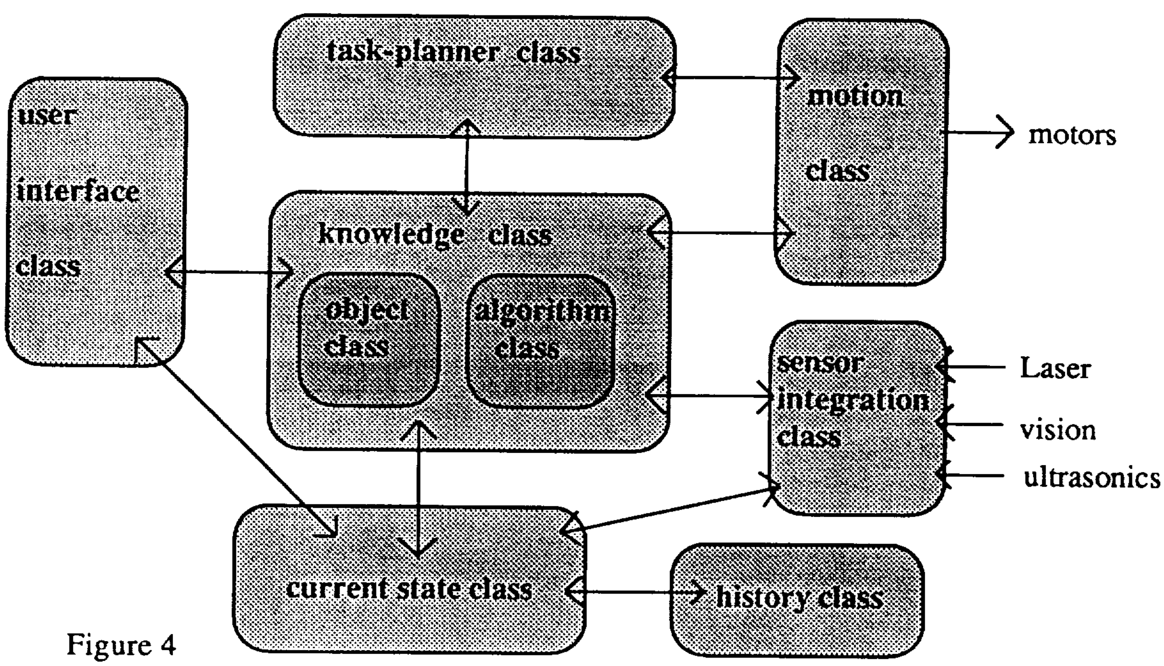

Figure 4

The central system in the robot is the knowledge-class. This system stores knowledge not only about objects and their interrelationships, but also about techniques for subdividing tasks, and the relationships/dependencies of tasks. The task-planner class is responsible for developing goals and sub-goals, for resolving conflicts, and for generating the most efficient plan. It queries the current-state class, the knowledge class, and the user-interface (when necessary) about how to perform its duties. In addition to maintaining a record of its "current" tasks, the task-planner can develop *theoretical* plans and simulate their progression to determine the best plan. The user interface contains nodes for controlling and creating the environment that the human interactor sees. It controls simple commands/responses sent and it relays information as a debugger. Ultimately, it will also contain a meta-english language (i.e. a structured and simplified natural language) for high-level interaction. The current state class represents the robot's most up-to-date perception of its surroundings and its status. This class contains several different representations of the objects and relationships in its vicinity. These include polygonal representations of objects (mostly used by the path planner), dependency nets, solid models and abstract semantic nets. Depending on which subsystem is sending/receiving information to this class, different representation are presented in response. The history class is a *selective* history of the robot's actions and surroundings. The sensor-integration class deals with developing a complete model of the environment. It creates "sketches" from each sensor and integrates them to form a complete model. This model is passed to the current-state class to enhance its model. The integration-class uses information from the knowledge class to identify objects and to aid in the scheme construction. Finally, the motion-control class handles the obstacle-avoidance, low-level trajectory planning, and local path planning. It draws upon the current-state class for a map of the surroundings (which may be incomplete).

5. Results

The hardware systems on the Rice-obot I are nearing completion. Figure 5 shows the assembled Rice-obot I. The Explorer has been mounted onboard the robot, and the arms, pan/tilt, z-tables and superstructure of the robot are mounted. Only the onboard battery power system, some of the arm-control circuitry and the laser mapper are not yet built. We have a fully functional version of Lisp-Nodes running onboard, and it is interfaced to the low-level sensor and motion control subroutines. Furthermore, the major sub-system class nodes exist, and a preliminary set of rules for their interaction. We have also installed a local path-planner based upon the paper by Alex Meng[6]. Meng's path planner is functional on the Explorer, although it has not been fully integrated into the Lisp-Nodes environment. This path-planning system has already provided two useful results: a) it has demonstrated and tested the basic mobility and control of the robot and b) it has demonstrated the ability of the robot to easily attach developed subsystems. The main areas to be developed on the robot are the sensor integration subsystem and the current-state system. Furthermore, it is not yet clear what extensions (if any) need to be made in the structure of the interaction between subsystems to ensure consistency. This will become most critical as the current-state system is further developed, since it communicates heavily with other subsystems.

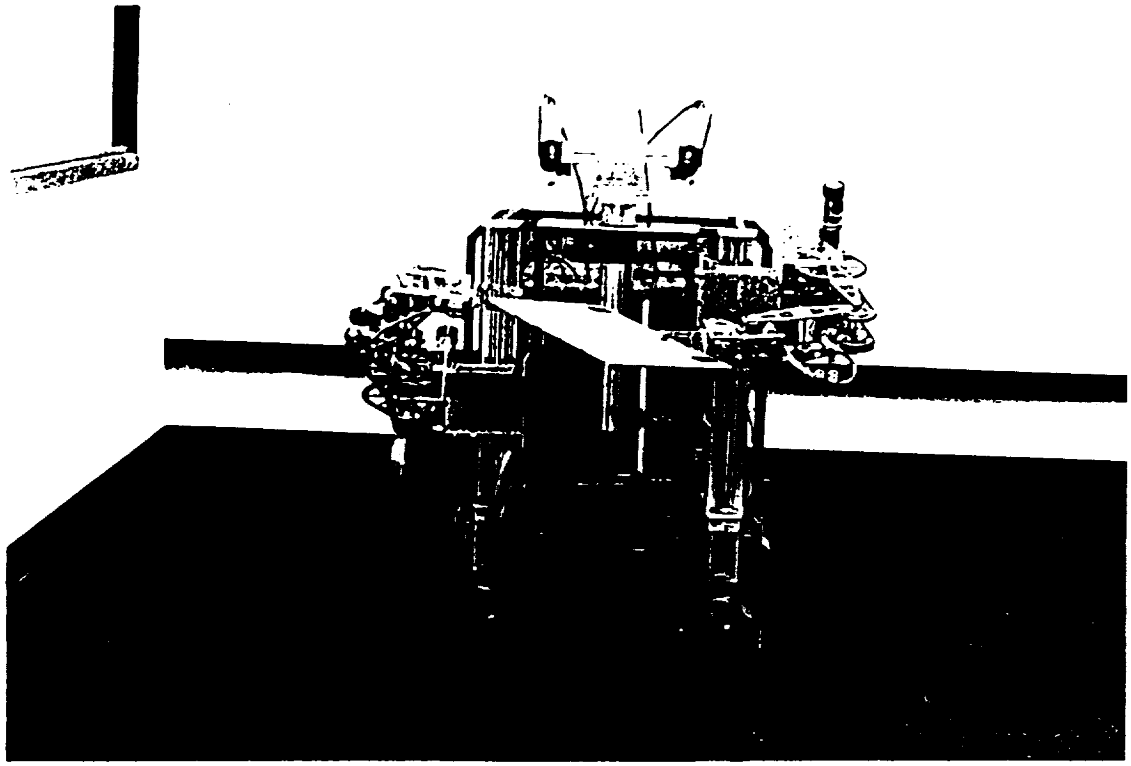

Figure 5

6. Significance towards Exploration & Space Robotics

One of the basic problems inhibiting development in intelligent robotics is the lack of a basic overall "operating system" for complex, multisensory mobile robots on which new concepts can be easily integrated with other already-developed pieces. Every time a new mobile robot is developed, the entire sensory/control/perception system has to be completely recreated. Furthermore, current intelligent control systems are very inflexible to modifications and cannot easily embody several different knowledge structures. Thus they tend to be "specialized" for a particular task; exploratory robots need a more broad-based control system. Lisp-Nodes provides the basis for such a control system. Furthermore, The Rice-obot I is a good tool for developing such concepts as Lisp-Nodes because of the advanced operating systems (UNIX and Explorer) onboard, because of the high bandwidth of data transfer among the subsystems, and because of the advanced sensors available. The Rice-obot I is the first step in the Rice University CIMR lab's goal of having multiple mobile robots cooperating in performing tasks in a real-world environment.

7. Bibliography

[1] DeFigueiredo, R.J.P., Wang, K., *An "Evolving Frame" Approach to Learning with Application to Adaptive Navigation,* Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Atlanta, Georgia, 1986.

[2] Kohn, W., Skillman, T., *Hierarchical Control Systems for Autonomous Space Robots,* Boeing Electronics and Computer Service Company.

[3] Weisbin, C., de Saussure, G., Kammer, D., *Self -Controlled- A Real-Time Expert System For an Autonomous Mobile Robot,* Computers in Mechanical Engineering, Vol.5 No. 2, Sept., 1986.

[4] Isik, C., Meystel, A., *Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot,* IEEE Journal of Robotics and Automation, Vol.4, No.3, June, 1988.

[5] Meng, A., *Free Space Modeling and Geometric Motion Planning Under Location Uncertainty,* Artificial Intelligence Lab, Texas Instruments.

[6] Meng, A., *A Methodology of Map-Guided Autonomous Navigation with Range Sensor in Dynamic Environment,* Proceedings of the SPIE Conference on Mobile Robotics III, 1987.

[7] Hayati, S., *Hybrid Position/Force Control of Multi-Arm Cooperating Robots,* IEEE Journal of Robotics and Automation, 1986.

[8] Hirizinger, G. *Multisensory Robots and Sensor-Based Path Generators,* IEEE Journal of Robotics and Automation, 1986.

[9] Harmon, S.Y., *A Technique for Coordinating Autonomous Robots,* IEEE Journal of Robotics and Automation, 1986.

[10] Ahmad, S., *Real-Time Multiprocessor-Based robot Control,* IEEE Journal of Robotics and Automation, 1986.

[11] Nayeswara Rao, S.V., *Concurrent Algorithms for Autonomous Robot Navigation in an Unexplored Terrain,* IEEE Journal of Robotics and Automation, 1986.

[12] Harmon, S.Y., *Sensor data Fusion through a Distributed Blackboard,* IEEE Journal of Robotics and Automation, 1986.

[13] Thorpe, C.. *An Architecture for sensor Fusion in a Mobile Robot,* IEEE Journal of Robotics and Automation, 1986.

[14] Parodi, A.M. *An Intelligent System for an Autonomous Vehicle,* IEEE Journal of Robotics and Automation, 1986.

[15] Zacksenhouse, deFigueiredo, R.J.P., Johnson, D., *A Neural Network Architecture for Cue-Based Motion Planning,* to be presented at the 27th IEEE Conference on Decision and Control, Austin, Tx., Dec., 1988.

[16] Brooks, R., *A Hardware Retargetable Distributed Layered Architecture for Mobile Robot Control,* MIT Artificial Intelligence Laboratory.

[17] Flynn, A., Brooks, R., *MIT Mobile Robots- What's Next?* MIT Artificial Intelligence Laboratory.

[18] Skillman, T., Kohn, W., *Blackboard Based Hierarchical Intelligent Control System,* AIAA Journal Guide Cont. and Dyn.

[19] Nii, P., *Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures,* The AI Magazine, Summer, 1986.