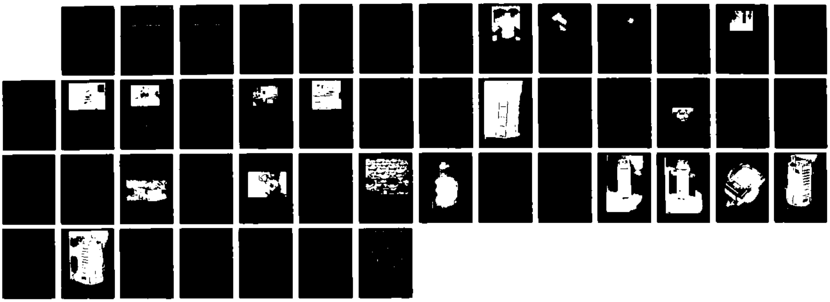
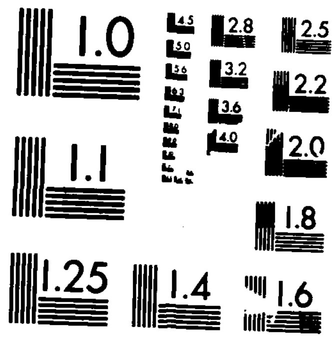
AD-A183 179 SURVEY OF MOBILE ROBOTS(U) MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB A M FLYNN DEC 85 NAVSEA-450-90G-TR-0004 1/1

UNCLASSIFIED F/G 23/3 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC FILE COPY

(12)

NAVSEA Technical Report
No. 450-90G-TR-0004

AD-A183 179

SURVEY OF MOBILE ROBOTS

Anita M. Flynn

Research Scientist

MIT Artificial Intelligence Laboratory

December 1985

NAVAL SEA SYSTEMS COMMAND
Washington, D. C. 20362-5101

NAVSEA Technical Report
No. 450-90G-TR-0004

SURVEY OF MOBILE ROBOTS

Anita M. Flynn

Research Scientist

MIT Artificial Intelligence Laboratory

December 1985

NAVAL SEA SYSTEMS COMMAND
Washington, D. C. 20362-5101

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 450-90G-TR-004	2. GOVT ACCESSION NO. AD-A183179	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SURVEY OF MOBILE ROBOTS		5. TYPE OF REPORT & PERIOD COVERED Technical Report; Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Anita M. Flynn		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Robotics and Autnomous Systems (SEA 90G Naval Sea Systems Command Washington, D.C. 20362-5101		12. REPORT DATE December 1985
		13. NUMBER OF PAGES 43
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Algorithm	Map Making	Sensors
Artificial Intelligence	Mobile Robot	Sonar
Autonomous Vehicles	Navigation	Vision
Laser Range Finder	Obstacle Avoidance	
Manipulator	Path Planning	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document consisting of text and graphics reports on a survey of robots developed from 1967 to the present. The report outlines the work which has been done in the past on mobile robots and summarizes certain projects currently being pursued. Special emphasis is placed on how these efforts have approached or solved the problem of modeling the environment and using the model for the purposes of navigation.

Table of Contents

Chapter One: Survey of Mobile Robots 3

1.1 Shakey 1967-1969 3
1.2 The Jet Propulsion Laboratory Mars Rover 1970-1973 7
1.3 The Stanford Cart 1973-1981 10
1.4 MELDOG 1977-1981 13
1.5 Hilare 1977- 14
1.6 Robart I 1980-1981 17
1.7 SCIMR 1981 21
1.8 Australian Robot 1980- 23
1.9 Autonomous Free Swimmer 1981- 23
1.10 Trent Robot - UK 1983- 25
1.11 Ground Surveillance Robot 1983- 26
1.12 The Unimation Rover 1983 27
1.13 Topo 1983 29
1.14 The CMU Rover 1982- 30
1.15 Robart II 1982- 32

Table of Figures

Figure 1-1: Shakey 4
Figure 1-2: Shakey's Grid Model 5
Figure 1-3: The Grid Model Cannot Clearly Represent the Obstacle as a Square 6
Figure 1-4: The JPL Mars Rover 8
Figure 1-5: The JPL Rover's Map 9
Figure 1-6: The Stanford Cart 11
Figure 1-7: The Cart's View of the World 12
Figure 1-8: MELDOG - Guide Dog Robot 14
Figure 1-9: Hilare 15
Figure 1-10: Hilare's World Model 16
Figure 1-11: Choosing a Path Through Midpoints of Edges of Adjacent Cells 17
Figure 1-12: Robart I 18
Figure 1-13: SCIMR 21
Figure 1-14: Ground Surveillance Robot 24
Figure 1-15: Omnidirectional Base 28
Figure 1-16: Androbot's Topo 30
Figure 1-17: The CMU Rover 31
Figure 1-18: Robart II - A Front View 34
Figure 1-19: Rear View - Controlling Computers and Interface Electronics 35
Figure 1-20: Drive Wheel Base 36
Figure 1-21: Low Level Computers and Sonar Transducer Driver Boards 37
Figure 1-22: Top Level Processor and Speech Synthesis Computers 39

Chapter One

Survey of Mobile Robots

Starting in the late sixties, research groups in the United States, France, England and Japan supported a few large projects in autonomous vehicles, and recently, due to the advent of inexpensive and powerful microprocessors, dozens of companies have entered the mobile robot market. The survey which follows outlines the work which has been done in the past on mobile robots and summarizes some of the projects being pursued now. Special emphasis is placed on how these endeavors have tackled or solved the problem of modeling the environment and using such a model for the purposes of navigation.

1.1 Shakey 1967-1969

Some of the earliest and yet at the same time most sophisticated work in applying artificial intelligence to robots was done at the Stanford Research Institute in the late sixties on an automaton named Shakey [Nilsson 69, Coles 69]. Shakey, Figure 1-1, operated off a large time-sharing computer, an SDS 940, by radio link and had both a FORTRAN executive for control and I/O, and a LISP executive for maintaining its world model.

Its main sensor was a rotatable camera, and with this sense of vision and its many levels of software, it was able to navigate, explore and learn. This was some of the earliest work in machine vision, and one lesson learned was that vision was a hard problem. Shakey also had natural language capability. A person could type in an English sentence command, and Shakey would parse the sentence and call up the appropriate FORTRAN or LISP programs to carry out the command.

Shakey's view of the world came from two models: a grid model and a property list model. The grid model divided the room up into nested 4x4 arrays called cells, where each element of the array was called a square. The entire world consisted of one cell, in which

Figure 1-1:Shakey

each square could be marked as full, partly full or empty. Partly full squares could then be represented as cells and further subdivided into 4x4 arrays of squares. Thus the room could be resolved to any desired level of detail, while its representation would require only a minimal amount of computer memory. From the model, obstacle-avoiding trajectories could be calculated as shown in Figure 1-2. It was more difficult however, to plan journeys by using the grid model than by using a fully divided large array [Rosen 68]. Additional information had to be maintained to help programs using the grid model, such as depth of the cell in the model, coordinates of the cell, lengths of the sides, and pointers to parent squares or cells.

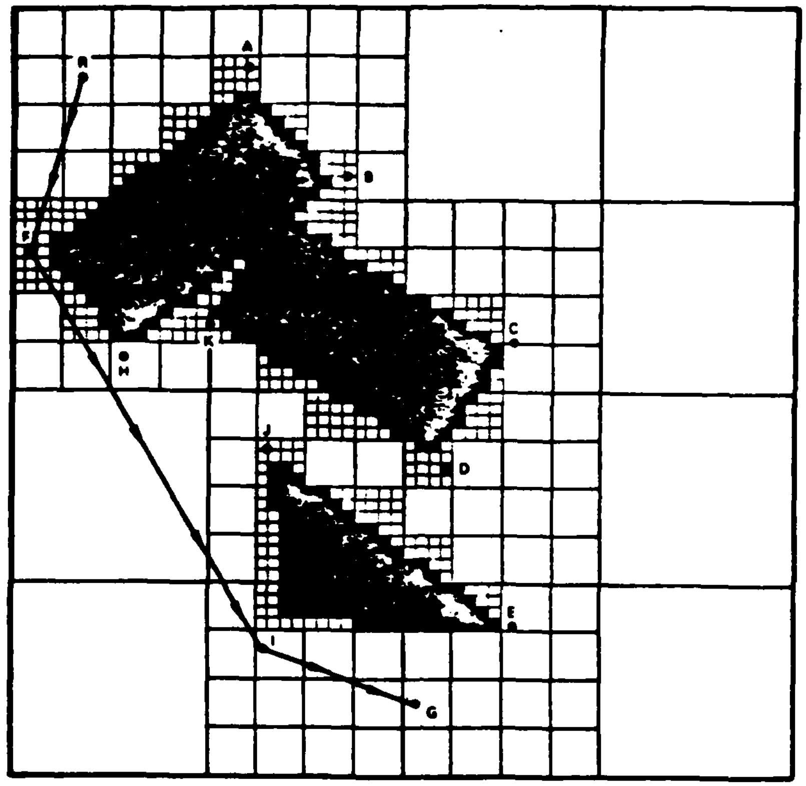


Figure 1-2:Shakey's Grid Model

Vision was used as an input to the grid model. The camera would take a picture, convert it to a line drawing, determine floor boundaries of objects, and calculate free floor space. It would then add full and empty areas into the grid model.

One problem was that the robot's position was "dead reckoned" by keeping track of wheel rotations, and errors due to slippage caused Shakey to miscalculate his position. This forced the vision system to incorporate objects incorrectly into the grid model. Because of this, it was noted that effective reorientation techniques would be an important area for future study.

Although the grid model was usable for journey planning when the robot was only concerned about free or empty areas, the grid model was not suitable for other functions such as object identification.

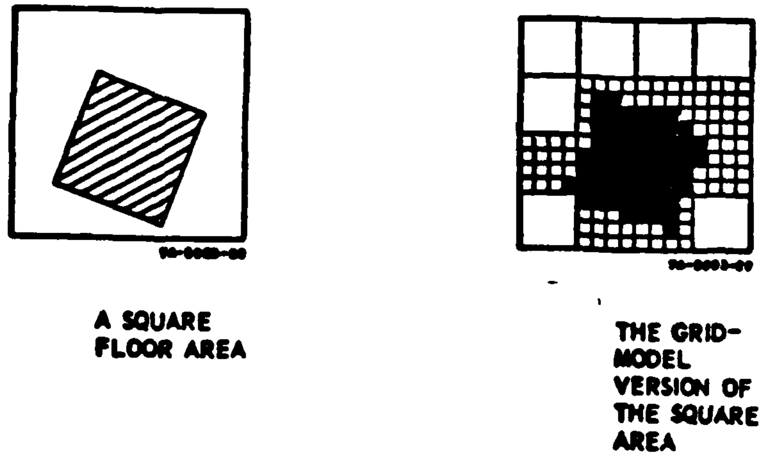


Figure 1-3:The Grid Model Cannot Clearly Represent the Obstacle as a Square

As seen in Figure 1-3, the jagged edges in the grid model's representation of the square made it hard for the robot to recognize it as such. To solve this problem, a line model was proposed in which visual images would be processed into line drawings and a straight-line representation of obstacles would be used for a model. This was not successful, however, due to the inability of vision systems at that time to provide the accuracy needed.

In addition to the grid and line models, a property list model was utilized. The property list model, later becoming the n-tuple model, represented objects in terms of their properties, using LISP type data constructs. Thus an object somewhere in the room might be denoted as an ordered list of such features as x-coordinate, y-coordinate, angle, size, shape, etc. The property list model was used for interpreting commands such as "GO TO A BOX". The coordinates would be looked up under an object named "BOX", then the grid model would be accessed by FORTRAN routines to determine collision free paths and to carry out the task.

The integration of the hierarchical levels of software gave Shakey the sophistication to remain the state of the art robot for many years. What is odd, is that Shakey, at the time, was considered a failure or at least an example of something the AI community had promised but couldn't deliver - namely, a completely autonomous robot. Shakey's environment had to be very simple for all his systems to work, and he was very slow, and well, "shakey". Funding on mobile robot research diminished and sponsors became disenchanted with AI in general for various reasons [Dreyfus 79]. This was mainly due to a change of heart at the Defense Advanced Research Projects Administration and not for scientific reasons.

The main lesson learned was that the instinctive skills which are easy for humans, such as seeing, moving, etc., are very hard to program into a robot, whereas higher level functions that are hard for humans, such as calculating, are much easier for a robot.

One of the contributors to the Shakey project was once asked if all the work that went into Shakey could have been done in software as a simulation. His answer was negative, because they wouldn't have known what to simulate. The difficulty lay in designing algorithms for poor data, not for perfect data, and they would not have known in which ways the data would have been poor [Raphael 68].

After Shakey, funding was continued in the areas of vision, natural language processing and planning, as serious problems in and of themselves, and not necessarily as subproblems of a mobile robot system.

1.2 The Jet Propulsion Laboratory Mars Rover 1970-1973

In the early seventies NASA began a project to develop a rover to be used in planetary exploration [Lewis 73, Dobrotin 77, Miller 77, Lewis 77, Thompson 79]. It had been noted in previous Viking missions, that due to long telecommunication delays it had taken several days to move a rock. Advantages sought in an autonomous robot would be reduced cost in both time and money for future space missions.

Figure 1-4:The JPL Mars Rover

The JPL robot, Figure 1-4, consisted of a mobile vehicle equipped with a six-degree-of-freedom manipulator (a modified Stanford arm) and an assortment of sensors (laser range-finder, stereo TV cameras, tactile sensors and proximity sensors). The navigation system used a gyrocompass and optical encoders on the wheels for dead-reckoning. An on-board mini-computer (General Automation SPC-16 with 32K memory) for real-time control of motors communicated with a remote PDP-10 on the Arpanet. The remote system was used to process TV and laser pictures, to construct the "world model" and to do planning and decision making. The robot, however, never advanced beyond the stage of being tethered with a 50-100 foot cable.

The rover's objective was to analyze a scene for traversability, plan a path to the goal and follow that path without bumping into anything. These objectives were achieved only in a simplified environment consisting of a laboratory with a flat surface, a limited number

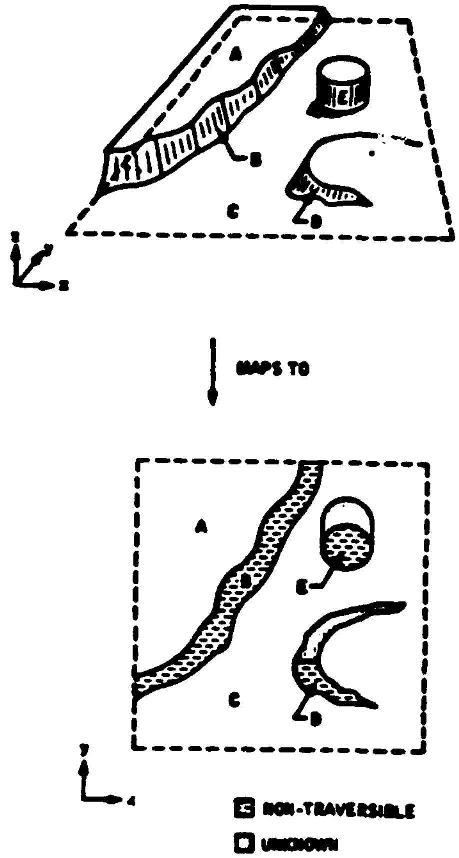


Figure 1-5:The JPL Rover's Map

of obstacles and constant illumination.

The model of the world held by the JPL Rover was a segmented terrain model derived by inputs from the vision system. Since the area explored by the robot was large, the terrain model was partitioned into map sectors of a convenient size and stored as separate files. Each sector was a fixed lattice of grid lines drawn parallel to the Rover's absolute coordinate system. The resultant collection of map sectors was similar to a catalog of charts. Each map sector represented areas that were either not traversable or unknown, as shown in Figure 1-5. All other areas were assumed traversable. Non- traversable regions were described as boundaries of polygons and these regions were then represented as lists of the vertices of those polygons.

This map had to be continually updated while the robot moved around performing its assigned task, and errors frequently got incorporated into the model. The first source of error was the uncertainty in vehicle position due to dead reckoning. This error increased with the distance from a known location. The second source of error was the limitation of the vision system to accurately determine relative positions of obstacles. Once an internal model was built the rover could refer to that model and using various search algorithms, plan an optimum route to its goal.

Although the JPL Rover project was able to produce several useful robotic subsystems such as the manipulator, the laser range finder and the navigation system, putting them together did not result in a completely autonomous robot as desired. The tether still remained and improvements were still needed to reduce errors in the respective subsystems so that the final system would be able to act intelligently and with a higher level of coordination. It was the classic case of an attempt at system building before the technology for the components was available.

1.3 The Stanford Cart 1973-1981

From 1973 to 1981, work was done at the Stanford University Artificial Intelligence Lab by Hans Moravec on developing a remotely controlled TV equipped mobile robot [Moravec 81, Moravec 83]. A crude cart was used as the mobile platform, but a sophisticated vision system and appropriate navigation and obstacle avoidance software enabled the Cart to move through cluttered spaces.

The Cart with its camera system is shown in Figure 1-6. The Cart used stereo imaging to locate objects and to deduce its own motion. A TV link connected the Cart to a remote KL-10, which sent control commands to the Cart and also did all image processing. The camera on top of the Cart was mounted on rails and slid by remote control to nine different positions to get nine pictures of the view before it. These pictures were then digitized and processed to extract 3D information from the scene.

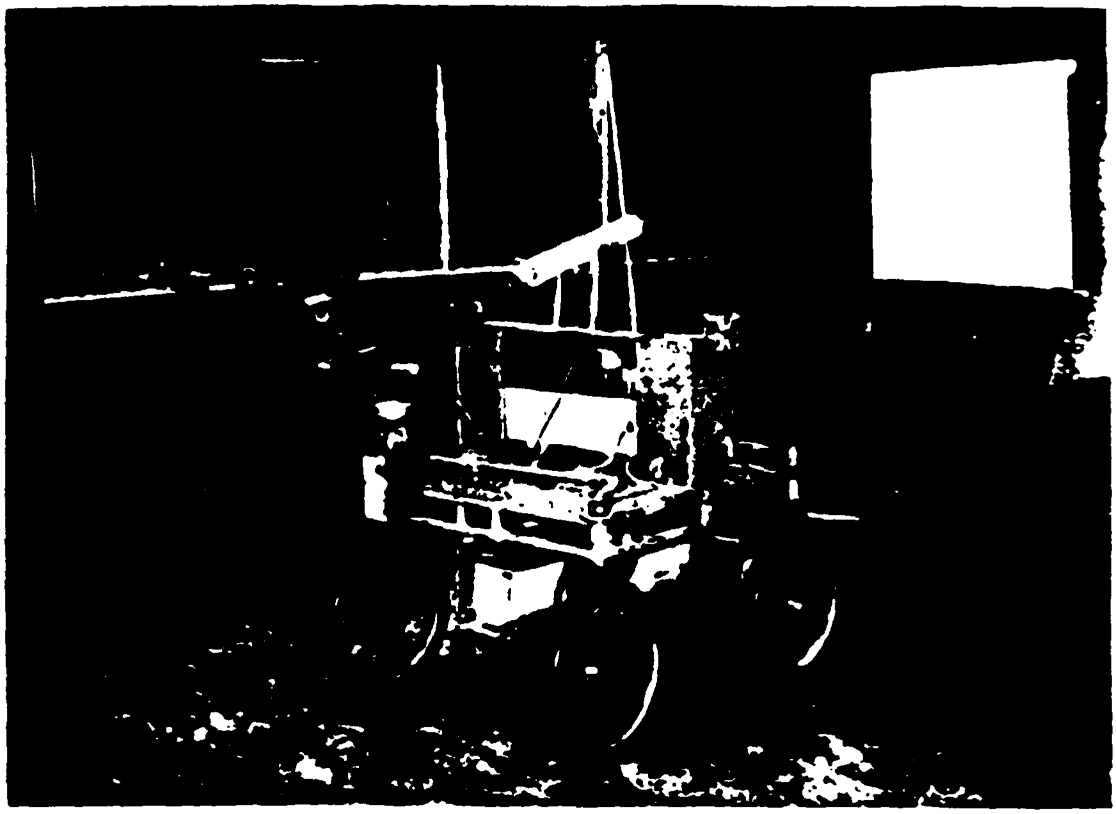


Figure 1-6:The Stanford Cart

Processing of the pictures amounted to extracting features from each picture and then correlating those feature points between any two images. Features were extracted by running an "interest operator" over each digitized picture, which would pick out areas in the picture which had the maximum gradient of grey scale. Thus points such as the corner of a table would be picked out because the top of the table might be well lit while the side was dimmed by shadow. This feature point would be marked in all nine pictures and then a correlator routine would compare that feature point's change in pixel position between any two pictures. Knowing that information and the distance that the camera had moved gave distance to the object. Nine pictures were used to increase reliability. The digitized image with its feature points marked is shown in Figure 1-7. Also shown is the path which the Cart has planned to reach its goal.

This information was used to build a model, and from this model it would plan an

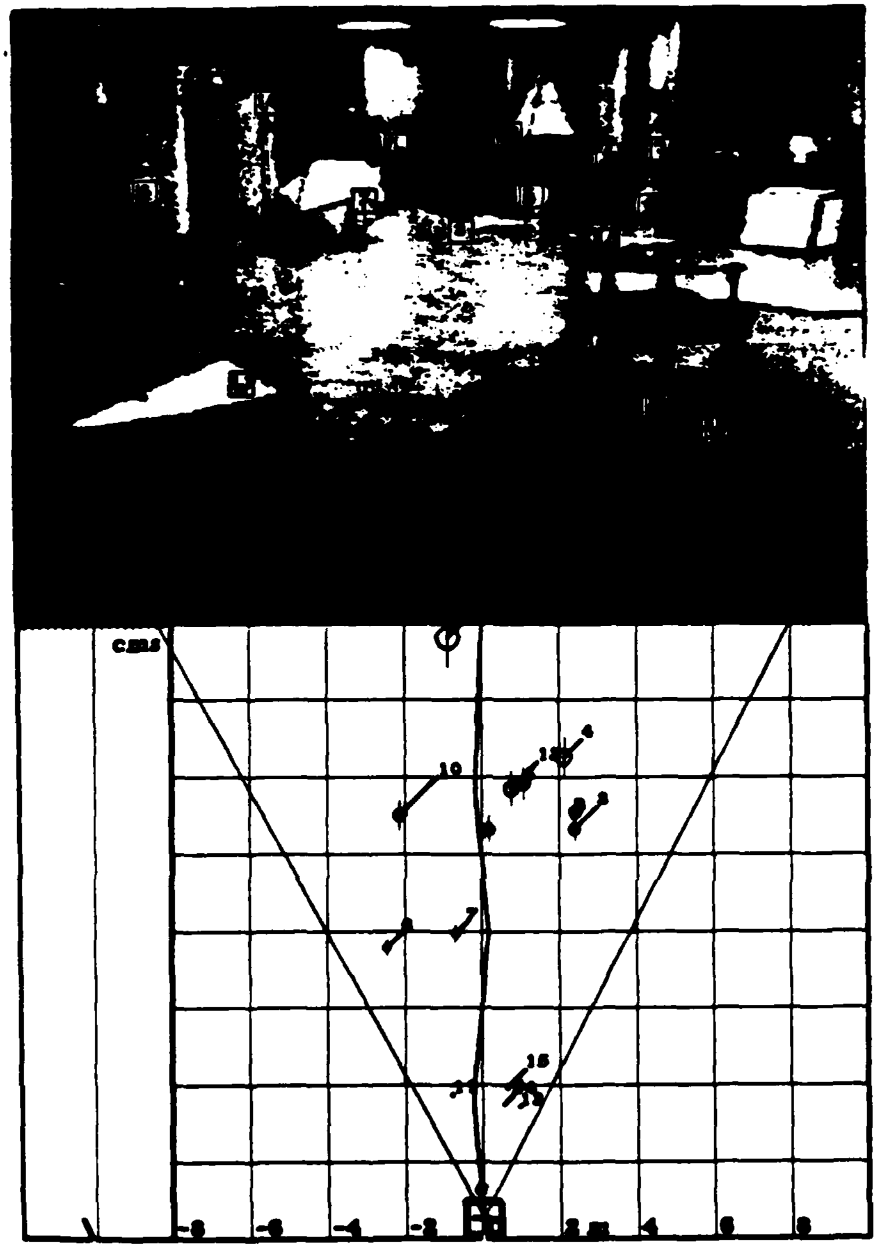


Figure 1-7:The Cart's View of the World

obstacle avoiding path to its destination. The system worked but was slow due to many factors. These ranged from the many computations needed to deduce the cart's own motion since its own dead reckoning system was so weak, to the fact that the system utilized interpreted LISP running on pre-LSI technology. The Cart would move one meter, stop, take pictures, think for fifteen minutes, and then move forward another meter. The Cart successfully maneuvered through several 20 meter courses (each taking about five hours) but failed in other runs.

Some problems in these runs were that featureless objects were hard to see, and also that shadows often moved considerably during the course of the run, throwing off the correlator since shadows produced new feature points due to their high contrast. Another problem involved weaknesses with the vision system's ability to maintain an accurate self-position model. Although the model was updated after each lurch, small errors in the measured feature positions sometimes caused the solver to converge to a position with an error beyond the expected uncertainty. Any features incorporated into the model after the Cart lost its correct sense of self-position were inserted wrongly. These errors were cumulative and caused the same object to seem to be in another place. The combination of old and new positions of these objects made it appear to the Cart that the path was blocked when in actuality it was open.

1.4 MELDOG 1977-1981

Beginning in 1977, the Japanese began a five year project to build a robot which would act as a seeing eye dog for a blind person. [Tachi 81] MELDOG (Mechanical Engineering Laboratory DOG) walks its master along the streets, stopping at intersections, and avoiding obstacles. Intersections are marked by landmarks which the robot can sense. The blind person keeps a mental map of which intersections to make a turn and the robot guides the person safely between intersections, as depicted in Figure 1-8. Ultrasonic transducers are used for obstacle avoidance and for tracking the landmarks.

Communication between man and dog is over a flexible wire link. Control commands

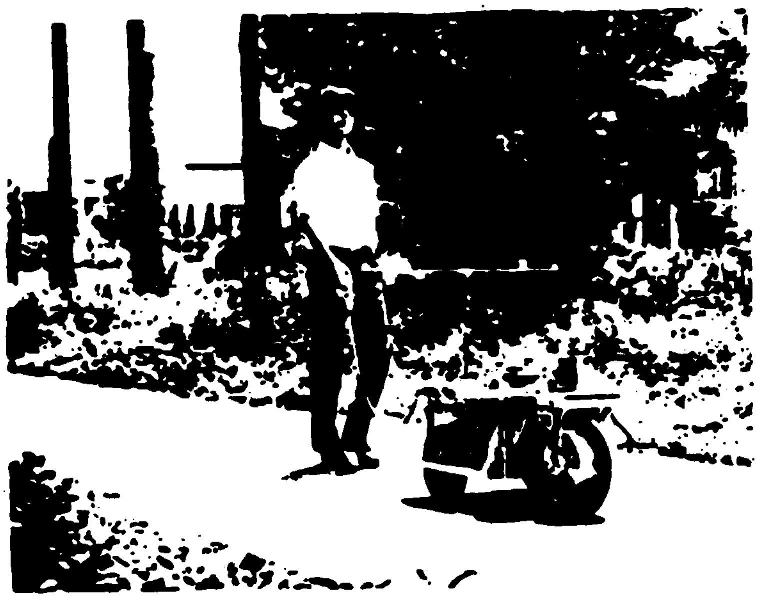


Figure 1-8:MELDOG - Guide Dog Robot

such as LEFT, RIGHT, STRAIGHT and STOP, are transmitted by control switches on the harness. Alarms from dog to man signalling danger are transmitter over the link also, but are in the form of mild shocks to the blind person's hand. Ultrasonic transducers are also used in a feedback system between man and dog, so they can walk fast or slow, but a distance of one meter between the two is always maintained. MELDOG has been successfully tested and may one day truly help the handicapped.

1.5 Hilare 1977-

Work began in 1977 in France at the Laboratoire d'Automatique et d'Analyse des Systemes to develop an autonomous robot that was not specialized for any given task or environment, utilized multiple sensors and was equipped with a multi-level computer and decision system [Briot 81, Bauzil 81, Ferrer 81].

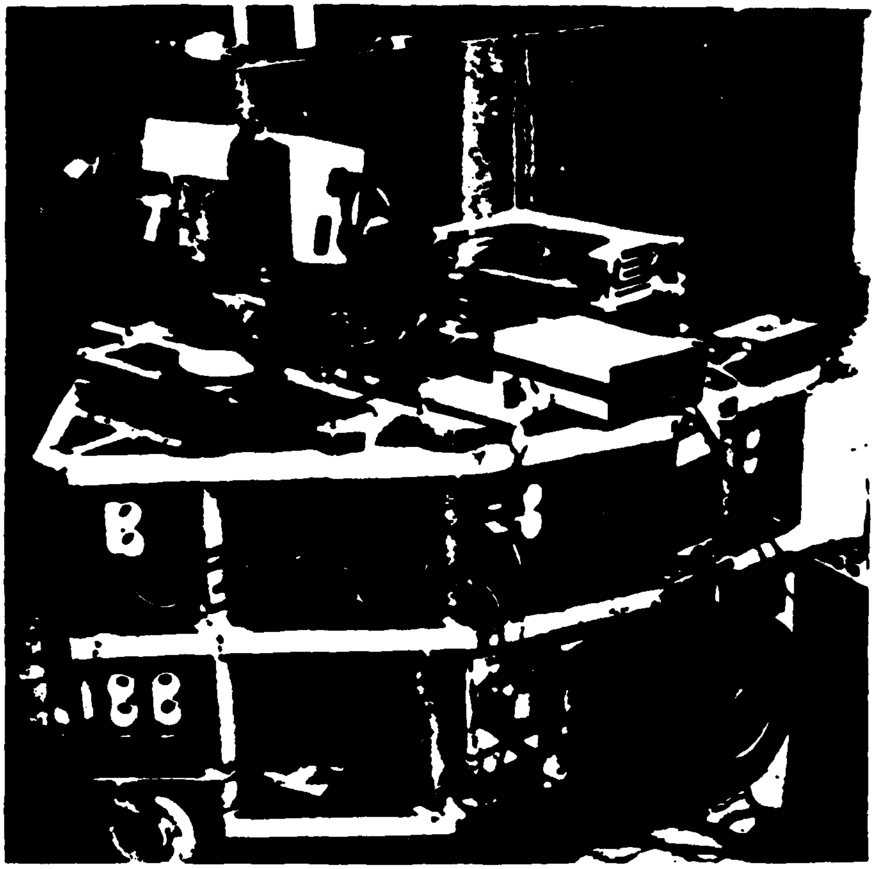


Figure 1-9:Hilare

Hilare, Figure 1-9, has a 3D vision system which uses a laser range-finder in conjunction with a video camera. Its sensor system also incorporates ultrasonic devices as proximity detectors for close-in obstacle detection and for paralleling a wall. It uses a system of infrared beacons mounted on the walls in the corners of its room to give it absolute positioning information. This works by using two infrared emitters and detectors on the robot. Measurements of angles are made by counting control pulses. The multi-level computer system consists of three 8085 on-board microprocessors for sensory data processing, an off-board MITRA-15 minicomputer for navigation and communication tasks, and a remote IBM-370 used as a peripheral to the minicomputer for complex tasks.

A distributed decision-making capability is provided through a system of cooperating expert modules which have expertise in the areas of object identification, navigation, exploration and planning. These modules consist of specialized knowledge bases, algorithms and heuristics, error processing capabilities, and communication procedures. This system enables Hilare to carry out navigation tasks which involve universe modeling, building a plan, and supervising the development and execution of that plan [Giralt 77, Laumond 83]. Hilare's world model defines obstacles as polyhedrons whose projections on the floor determine the navigation problem. This model can either be determined by the robot's perception system or provided as initial information.

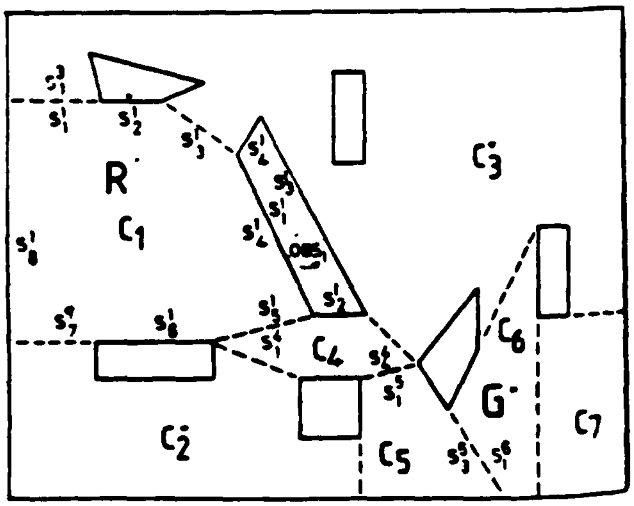


Figure 1-10: Hilare's World Model

The obstacles are represented as an ordered list of segments where each segment is represented by the Cartesian coordinates of its leftmost point, an angle with respect to some reference axis, and its length. As seen in Figure 1-10, empty areas are partitioned and represented as convex polygonal cells which include obstacle segments. Trajectories within cells are straight line paths between entry and exit segments so that adjacent cells have common segments which are traversable by the robot. This pattern of connexity can then be represented as a graph, which provides the structure necessary for path finding.

Optimum paths are determined by making a search over the resulting graph while minimizing costs in terms of distance and energy requirements. The minimization function is a linearly weighted combination of path length, angle of planned direction change, and

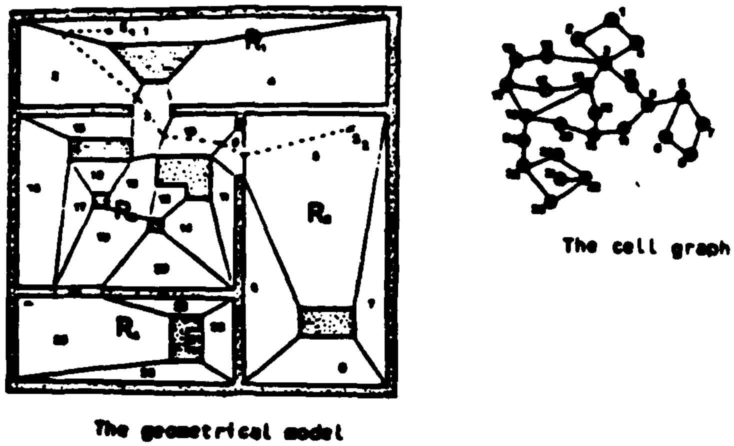


Figure 1-11:Choosing a Path Through Midpoints of Edges of Adjacent Cells

the number of predicted stops, together with a term which accounts for the uncertainty of information obtained by the robot and also the path viability due to estimated obstacle clusterings. Figure 1-11 shows a route chosen by the robot to navigate through several rooms from point S1 to point S2. By further structuring the graph representation, more efficient algorithms can be attained. In traversing from room R1 to room R2, the robot must always pass through room R3. Consequently, this path can be memorized, or more formally represented as a subgraph, as is also shown in Figure 1-11. Thus, even higher abstractions are attained and the robot is able to learn about the concepts of rooms and passages between rooms.

1.6 Robart I 1980-1981

Robart I was probably one of the first robots to be totally autonomous and yet still exhibit a high level of sophistication. Rapidly changing technology, including both the advent of the home computer and improvements in sensor technology, created the possibility of developing an autonomous machine that could perceive things about the environment, process that information, and then redirect that machine's actions accordingly.

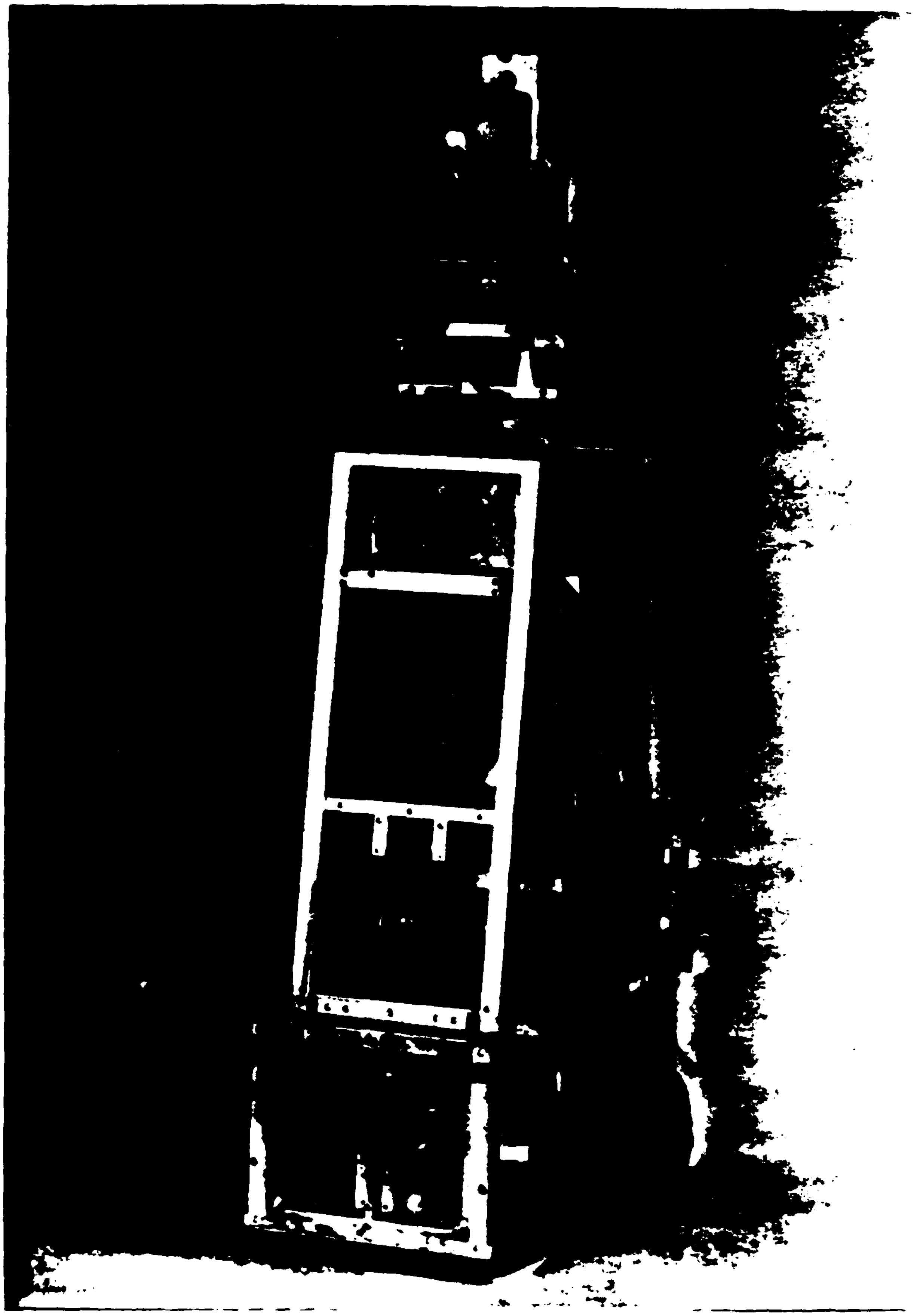


Figure 1-12: Robert I

Robart I was built at the Naval Postgraduate School by LCDR Bart Everett to serve as a feasibility demonstration for an autonomous robot and is the first known mobile sentry robot ever constructed [Everett 82a. Everett 82b]. Robart. Figure 1-12. would randomly patrol a house sensing for fire. smoke. flooding. toxic gas. intrusion. etc.. and take appropriate warning action if any of these conditions was found. The goal of the project was to show that certain applications could indeed be handled by autonomous mobile robots. using current technology. under the right conditions. The particular application of a sentry was chosen because it did not require any end effectors. or a vision system. The project was done on an extremely limited budget. using simplified approaches. the philosophy being that if successful under those conditions. an extrapolation should show the tremendous potential if later addressed with sufficient funding.

The robot had a single forward looking ultrasonic ranging unit. a long range near-infrared proximity detector that could be positioned by a rotating head. ten short range near-infrared proximity detectors, and tactile feelers and bumper switches for collision avoidance. The battery voltage was constantly monitored and when it fell below a certain adjustable threshold. the robot would activate. via a radio link. a homing beacon placed on top of its recharging station. For simplicity. an ordinary 75 watt light bulb was used as the beacon. tracked by an optical photocell array located on the robot's head. Thus the head position represented the relative bearing to the beacon, and the robot could home in on the battery recharger. The software provided verification of the correct beacon acquisition, the ability to maneuver around obstructions enroute, and the correction of any misalignment that occurred as a result of collision avoidance.

Other sensors onboard included a true-infrared body heat sensor which could detect a person out to a distance of fifty feet. This sensor was fairly directional, and mounted on the head so as to be positionable under software control. Also mounted on the head was a near-infrared long range proximity sensor with a parabolic reflecting collector. able to detect the edges of an open doorway to within an inch at a distance of six feet. This angular resolution allowed the robot to steer toward the center of the doorway while still some distance away.

In addition to its multitude of sensors, the five-foot-tall Robart could also speak. Voice synthesis was not only used to warn of the presence of intruders or other alarm conditions, but could also report on the internal status of its circuits, system configuration errors, time-of-day, temperature, etc.

Robart's behavior appeared arbitrary, or at least not preprogrammed. An operating system provided for the selection of various behavior primitives, each designed to meet a specific goal, based on the output of specific sensors, via interrupt software. When no specific actions were called for, a routine was randomly chosen from a preprogrammed set of sixteen routines that filled in the gaps. Some of these routines would move the robot more or less randomly to a new vantage point, where it might elect to stop and re-enter the surveillance mode. Motion under these circumstances usually involved moving straight ahead, unless it saw an object, in which case it would swerve to one side or the other as appropriate. It would then continue moving in the new direction until it encountered another obstacle.

Robart could also be put in either the "Hostile" or "Friendly" mode. In the "Friendly" mode it would greet a person with an amiable "Hi" or "Hello", while in the "Hostile" mode it would announce "Intruder, Intruder", and then advise the intruder to leave the room.

All sensors were interfaced to one 6502 based SYM-1 computer on an interrupt basis. A triangular wheelbase was utilized, with the one front wheel providing power and steering. Optical encoders were not used so dead reckoning was not performed, but an A/D converter gave four bits of information on steering wheel angle. The rotating head had similar resolution. In the worst case, wheel and head together could have as much as 22 degrees of error when looking for the recharging station. This was done on purpose, however, to demonstrate the feasibility of software compensation. In over 200 dockings, Robart only failed once to hit his recharging station half an inch from the centerline of its front bumper. The entire robot was powered by one 12V 20 amp-hour battery, providing roughly ten hours of service, with fourteen hours needed for full recharge.

Robart I has been built, tested, and run. It has demonstrated the feasibility of supplying perceived data to a completely autonomous robot and having that machine react appropriately. Robart doesn't have a vision system, so it doesn't recognize obstacles or even remember where they were. As seen from previous robot projects, vision systems take lots of computer power, and that usually means off-board processing. Nevertheless, constantly improving computer technology promises to bring about much more powerful and smarter robots in the future.

1.7 SCIMR 1981

SCIMR (Self Contained Independent Mobile Robot) was a robot built at the University of Pennsylvania Moore School of Electrical Engineering. [Andersson 81, Andersson 82]

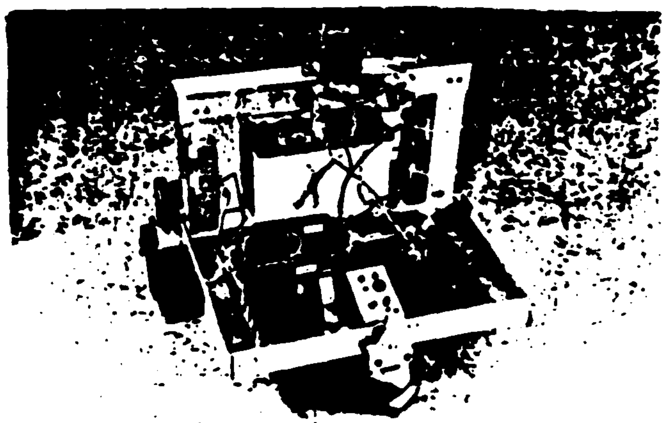


Figure 1-13:SCIMR

It was totally autonomous and used three interconnected 6808 microprocessors with 4K bytes of RAM each for control. One processor controlled the two rear wheels which powered and steered SCIMR. The second processor controlled the sole sensor, a rotatable Polaroid ultrasonic transducer. A third processor managed coordination, planning and communication with the other two. SCIMR is shown in Figure 1-13.

A multitasking operating system allows each of the processors to run up to eight tasks at a time, which gives SCIMR the capability to implement real time processing in its feedback systems. Such a feedback system is used to get SCIMR to move parallel to a wall. In order for the robot to maintain a set distance and orientation with respect to the wall, three measurements are taken by the sonar transducer. One measurement is taken perpendicular to the robot, which theoretically is also perpendicular to the wall. The other measurements are taken both fore and aft of the perpendicular. From this data SCIMR can deduce how much it should turn so that it will remain moving parallel with the wall.

SCIMR's environment was the hallways of the Moore School. It dealt only with passages and intersections. It would move down a hallway at some set distance from the wall, and could tell when it got to an intersection because the wall it had been following would disappear. SCIMR then built a map of the hallways he explored. The map was represented as a graph, where intersections were the nodes and the hallways were the edges that connected the nodes. The graph representation was chosen instead of a bit map representation, in which each bit would represent a location in space. A bit would be on if an object was at that position and off if there were only empty space. A bit map representation was deemed hard to generate, memory inefficient, and slow to process. Furthermore the sonars would be too coarse to supply reliable information in cluttered environments.

Two of SCIMR's major problems had to do with the sonars. First of all the sonar beamwidth is about 20 degrees. No focusing method was used to attempt to narrow the beam. A more difficult problem was that many surfaces can appear to a sonar beam, which has a wavelength of about 1/8 inch, as a mirror. Unless the beam is pointed exactly perpendicular to the surface, the reflected beam will bounce off the object at an angle equal to the angle of incidence. Consequently, some erroneous distance measurement is returned.

Other problems include a slow scan rate. SCIMR moves at one foot per second and scans at the same time. A stepper motor was used to rotate the sonar, and this limited the speed at which scanning could be done. Furthermore, a certain amount of time was needed between excitations of the transducer for timing out the maximum distance. SCIMR's world is very limited. By curtailing his environment to hallways, the navigation problem is simplified. He only has to make 90 degree turns whenever he reaches an intersection. Going down the next hallway, the wall follower routine assures that his orientation remains some multiple of 90 degrees from his starting orientation. Another problem SCIMR had was number representation in the 6808s. Care had to be taken to detect overflows and floating point numbers were out of the question. In addition, trigonometric functions always had to be approximated.

1.8. Australian Robot 1980-

The Australian National University has built a mobile robot for use as a research tool for computer vision. [Jarvis 80] The idea is that the relationship between computer vision and robotic action is not unlike the process by which humans learn to see. The robot is provided with two manipulator arms, a mobile base and visual, audio and ultrasonic sensors. Processing is done off-board by a NOVA 2/10 computer which is tethered to the robot. Two Z80 processors connected as slaves to the NOVA are used for real time image acquisition and for speech recognition and synthesis. A Genisco system provides color image and graphics output. The onboard TV camera's zoom, aperture and focus are controlled by individual stepper motors and the entire camera, which is mounted in a gimbal frame, can be swivelled to see a 60 degree solid cone in front of the vehicle. Ultrasonic sensors rotate with the camera and contribute 3D information about the scene. Further research is planned in using the vehicle to help gain an understanding of how humans see, and how that knowledge can be applied to computer vision.

1.9 Autonomous Free Swimmer 1981-

The Naval Ocean Systems Center has developed an unmanned free swimming submersible designed for underwater pipeline search and inspection. [Harmon 81] A multi-tasking operating system provides functions for pattern recognition, sensor and effector coordination, communication with the surface and representation of knowledge acquired from the environment. The software is split between two computers, an 8080 for the device interfaces and control processing, and an LSI 11/23 for sensor processing and knowledge representation functions. The swimmer could work either completely autonomously or tethered with a fiber optic link. The swimmer locates a pipe by searching for a magnetic signature similar to that of a pipe. An acoustic altimeter is also used to survery the ocean bottom near the pipe.

Representing this sensory data in a useful way, however, is not an easy problem. Harmon notes that while there are several prevalent schemes there is little or no experience in applying these schemes to the domain of mobile robots. Furthermore, there is no satisfactory way of representing temporal information which is critical to an autonomous robot. Semantic networks were first tried as a knowledge representation scheme but failed due to the problem with representing timing of events. A symbol net type of representation was next attempted but was supplanted by a proposal for a more advanced scheme called the Temporal Representation Inference Network Abstraction (TRINA). TRINA is a weighted directed graph where nodes represent objects, states, events or couplings. Activation of a sensor causes a node to propagate its influence to related nodes in space and time. This is facilitated by giving each node information regarding the neighboring nodes' expectations of the future regarding that node. A desired action of the robot can be chosen by designating certain nodes as action nodes and connecting them to the influence of other nodes which might represent conditional goals or situations. Whenever a sensor detects a certain situation, the desired action will then be generated. This scheme differs from conventional algorithmic programming in that each situation can be closely coupled to conditions apparent at that time.

1.10 Trent Robot - UK 1983-

Researchers at Trent Polytechnic have implemented an obstacle avoidance system for mobile robots that are presently used on some factory floors. [Cooke 83] These types of robots typically follow a guidewire placed under the floor, but should an obstacle appear in the path, they usually crash into it. The obstacle avoidance system implemented here uses a rotatable sonar transducer to detect an object in the path.

If an obstacle is detected, the cart then turns right or left and goes around it and picks up the guidewire again on the other side. Whether to turn right or left depends on which side has more clear space. The sonar looks to the left and then to the right and turns to the side with maximum free distance. The cart moves in that direction for five seconds, then rotates its sonar so that it is still pointing at the obstacle, and takes another reading to see if it has reached the corner yet. The algorithm assumes rectangular obstacles. When it has reached the corner, it records how far it has travelled through dead reckoning, and makes a turn. It travels up the side of the obstacle parallel to the direction it was originally headed, checking every five seconds for the next corner. It then makes the next turn and travels a distance equal to the distance previously recorded, and finally turns back onto the path it was previously on, with the obstacle behind it.

An improved version of this algorithm which has been proposed but not yet implemented is to use the sonars to scan the obstacles to determine distances to the corners and build a map which the robot can follow. This alleviates the need for stopping every five seconds to see if the obstacle is still there. Instead, upon first detecting the obstacle, the sonars are rotated, angles to the corners noted, and distance needed to travel calculated.

1.11 Ground Surveillance Robot 1983-

The Naval Ocean Systems Center has recently undertaken an ambitious project to build a completely autonomous vehicle to traverse rough terrain. [Harmon 83a, Harmon 83b] The vehicle used is an armored personnel carrier and it is designed to navigate from some starting location to a goal, where the initial and final positions are obtained from a satellite system, but where the area between consists of unknown terrain.

At the present stage, all controls such as steering, clutch, brake, etc., have been replaced with actuators which can be activated from a computer keyboard. Sensors are being added, such as sonar transducers for obstacle avoidance. It is planned to be able to

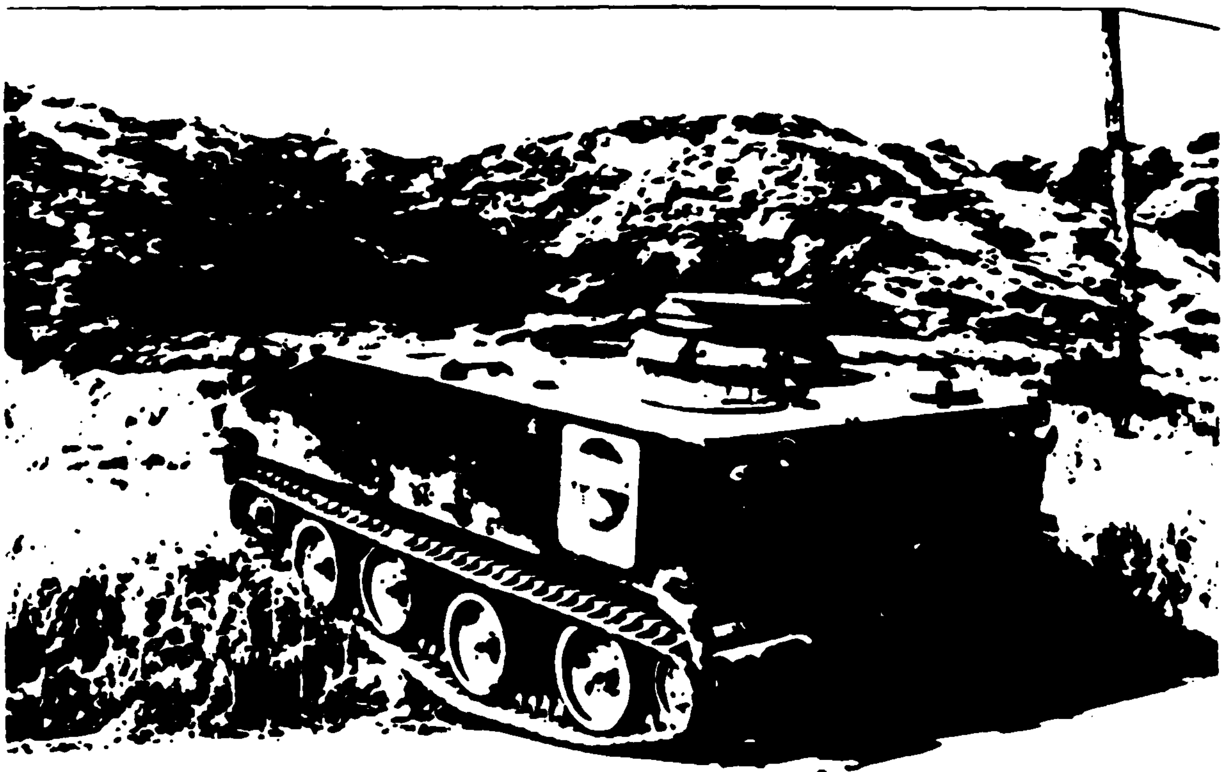


Figure 1-14: Ground Surveillance Robot

discriminate various textures from the returned sonar signals in order to distinguish soft obstacles from hard ones. That would aid in deciding whether or not to run over a bush or swerve around a tree. Also to be added are sensors for imaging (grey level vision systems and laser range finders). and senors for navigation and vehicle attitude.

It's one thing, though, to have lots of sensor or actuator subsystems working very nicely, and quite another to have them working together as an intelligent system. For this reason, care in designing the architecture for the software system is of prime importance. The main goal is flexibility, due to the fact that precise requirements for an autonomous robot are unknown. Harmon states that very little experience exists for such robots and no such system has been successfully demonstrated in a practical application as yet.

In order to maintain as much flexibility as possible, standard microprocessors and standard buses are used to build a hierarchial control system. A series of Intel 8088 microprocessors connected together with an RS232 bus are used for sensory processing, navigation, control of actuators, etc. Much of this software will be written in Pascal while speed critical code for control of actuators will be written in PL/M, an Intel version of PL-1. Above these processors in the hierarchy, will be 16 bit or 32 bit processors (such as the MC68000 or the National 16032) running some LISP-like language to implement a knowledge based expert system and to do the image processing.

Processing tasks fall in three groups, sensor processing, control, and knowledge based tasks, which intercommunicate by passing messages within a broadcast topology. Sensory input will be used to build a world model which will be represented as a relational network, where nodes represent various obstacles with various properties and links between nodes represent relationships between obstacles, such as distance.

1.12 The Unimation Rover 1983

Unimation has recently attempted to build a robot which incorporates a manipulator on a mobile base. The project has recently been passed onto the Mechanical Engineering Department of Stanford University. The base uses the omnidirectional wheel system which was originally designed at the Veterans Administration Rehabilitation Research Center. The omnidirectional base works by using three wheels which oppose each other, but which have rollers on their rims, thus allowing one wheel to roll freely while the other two are propelled. Proper combinations of wheel control allow steering in any direction without having to reorient the vehicle.

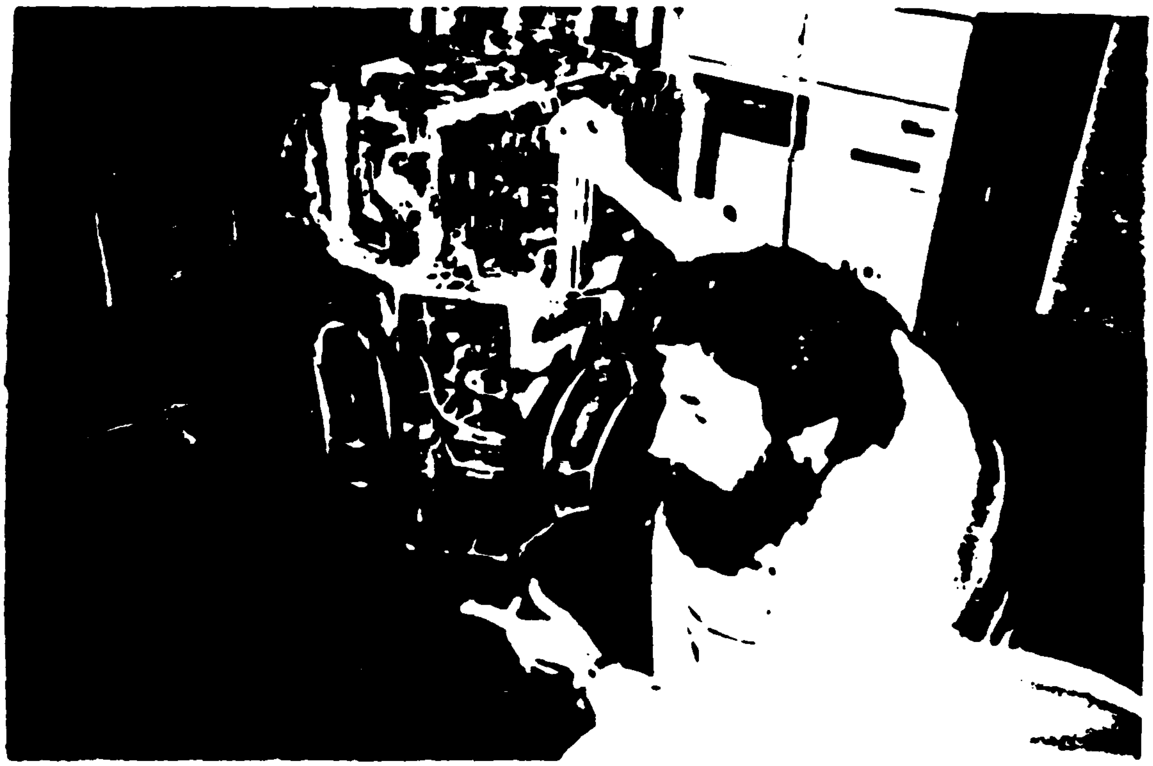


Figure 1-15:Omnidirectional Base

The base with electronics for the motor drivers is shown in Figure 1-15.

A major problem with this design is that power consumption is enormous. The PUMA motor controllers were not designed for mobility and are extremely power hungry. Six twelve volt batteries are needed for the six motors of the arm and the three of the base.

Even with six batteries, charge runs out after about an hour. The power constraint is always a very serious one for any mobile robot, but especially for one using a six degree-of-freedom manipulator which isn't energy efficient. This points out once again that robotics is really a systems problem, and that so much more goes into making an intelligent mobile robot than a colection of subsystems which work well individually.

1.13 Topo 1983

One robot which is getting a lot of press these days is Androbot's Topo (1-16). Topo's design philosophy is to provide, as cheaply as possible, a robot consisting of just its basic building blocks, which can be interfaced to a home computer, so that the customer can write the higher level software. A programming environment written in FORTH is provided in order to supply the customer with basic robot control primitives such as MOVE, STOP, RIGHT, LEFT, etc. An infrared link is used to send commands from the computer to the robot.

Topo's architecture is based on the Intel 8031. Two boards are provided, a communications board and a motor control board, with expandability to other boards available. Communications is effected by sending messages to the communications port on all the boards having only the appropriate board acknowledge its message. An improved version of Topo, called Bob, is supposed to be ready by the first quarter of 1984. It will be 8088 bases and have more memory, but more details are not available at this time.

Topo's sensors consist of sonar transducers for obstacle avoidance. These were used in a demonstration of Topo's ability to follow a person. Encoders on the wheels are provided for dead reckoning, but the canted wheel base doesn't seem well suited for this task, as it appears that slippage of wheels will be significant. Topo makes no attempt to build any internal representation of his world or to implement any other types of intelligent behavior. Rather, it leaves it up to the customer to write that software, while providing a mobile base and sensor platform.

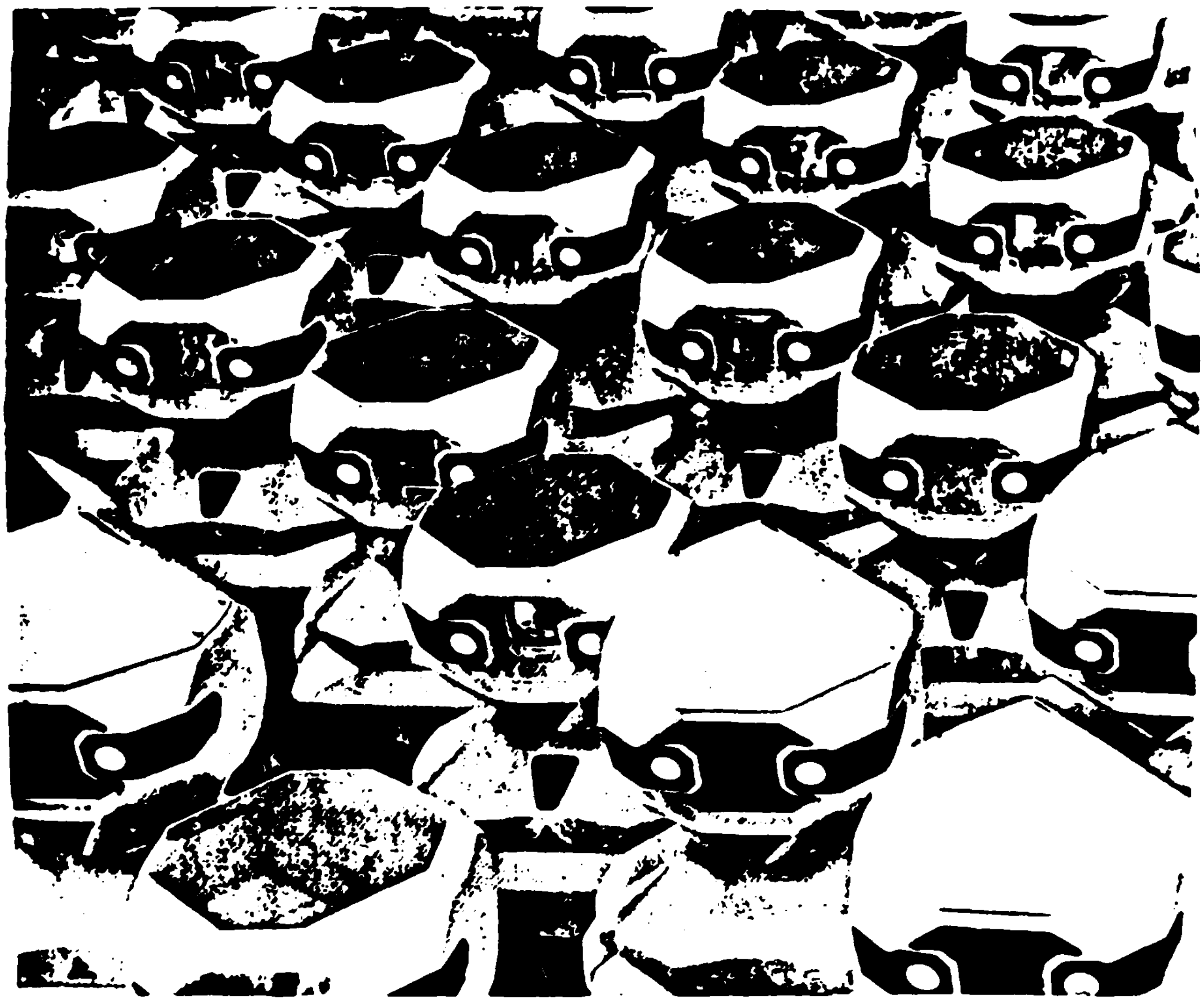


Figure 1-16: Androbot's Topo

1.14 The CMU Rover 1982-

Since the Stanford Cart project, Moravec has gone on to Carnegie Mellon University and has begun a follow on robot, which attempts to overcome many deficiencies of the Cart [Moravec 83]. The CMU Rover is first of all mechanically well designed so that many of the Cart's problems, such as breaking down or losing track of its position, are avoided.

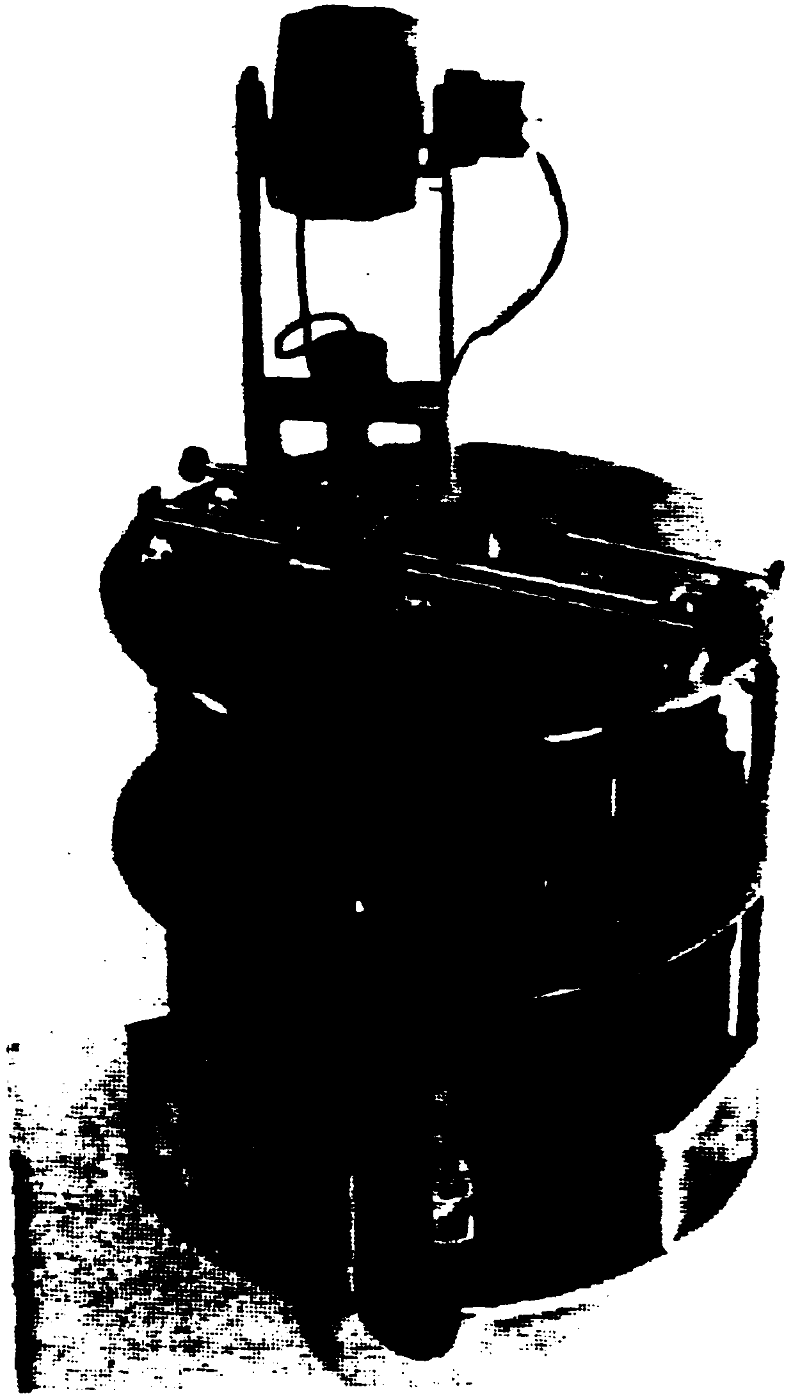


Figure 1-17:The CMU Rover

The Rover is much smaller than the Cart being about a meter tall and a half meter in diameter, as can be seen in Figure 1-17.

It has three independently steerable wheels which enable it to have a full three degrees of mobility in the plane. Optical encoders on the motor shafts give it dead reckoning capability. The Cart's only sensor was its TV camera. The Rover will have the same type of system but will also have infrared proximity detectors and ultrasonic rangefinders for obstacle avoidance.

All of the Cart's processing was done off-board on a remote mainframe. The Rover will still have an off-board computer, a VAX 11/780 with an attached array processor, to speed up the vision system, but will also have a dozen on-board processors (half of them being 16 bit MC68000s) for local decision making and control.

The control system at first was planned to be written in a language similar to presently used manipulator languages such as AL, VAL or AML, but attempts at defining the structures and primitives required for a mobile application pointed out that these essentially linear control systems would be inadequate for a mobile robot. The problem is that a roving machine is regularly faced with emergency situations (falling down stairs, running over a person, etc.) which it can't anticipate, but with which it must deal. The solution is to use independent processes that communicate via messages posted on a data structure called a blackboard. Processes can change their priority based on relevant messages posted on the blackboard. The Rover isn't finished yet, but it promises to be a powerful tool for studying the problems associated with an autonomous machine and and for determining what types of problems need to be solved to give such an entity intelligence.

1.15 Robart II 1982-

Robart II is being built by LCDR Bart Everett, Robotics Coordinator for the Naval Sea Systems Command, Washington D.C. Robart II is a second generation prototype sentry robot, built to improve some of the capabilities of its predecessor, Robart I.

Robart II is a battery operated autonomous mobile robot which stands four feet tall, and measures 17 inches across at the base. The system employs a control hierarchy of six onboard 6502 based micro-processors, and the platform houses a multitude of sensors for navigational planning, collision avoidance, and environmental awareness. These include six ultrasonic rangefinders, fifty near-infrared proximity detectors, a long range near-infrared rangefinder, plus various sensors used to detect special alarm conditions, such as fire, smoke, toxic gas, flooding, vibration, and intrusion. Four true infrared motion detectors are employed for detecting the presence of an intruder up to fifty feet away, reacting to the thermal radiation emitted by the human body. Special internal circuitry checkpoints are analyzed by self-diagnostic software, and operator assistance is requested if necessary through speech synthesis.

A front view of Robart II (Figure 1-18) shows the five sonar transducers on the body and one on the rotatable head. The long-range near-infrared sensor with parabolic reflecting dish sits on top of the head while three of the true infrared motion detectors can be seen mounted just below the head. A rear view (Figure 1-19) exposes the card cage which houses the six computers and all the driver circuits.

The entire system is also a vastly improved mechanical design, taking advantage of lessons learned on the earlier version. The propulsion system uses two individually controlled drive wheels on either side of the base, with casters in front and rear for stability (Figure 1-20). This configuration allows the robot to spin about its vertical axis for markedly improved maneuverability. The motors are each controlled through pulse width modulation, and synchronized by high resolution optical encoders attached to the armature shafts. A low level dedicated 6502-based controller handles all drive and steering functions upon command from the top level microprocessor. The optical encoders supply precise displacement and velocity information for use in dead reckoning during maneuvering. Conventional eight inch wheelchair tires and motors provide a quiet, powerful propulsion system with minimal wheel slippage.

A second low level dedicated 6502 controller is used to operate six ultrasonic ranging

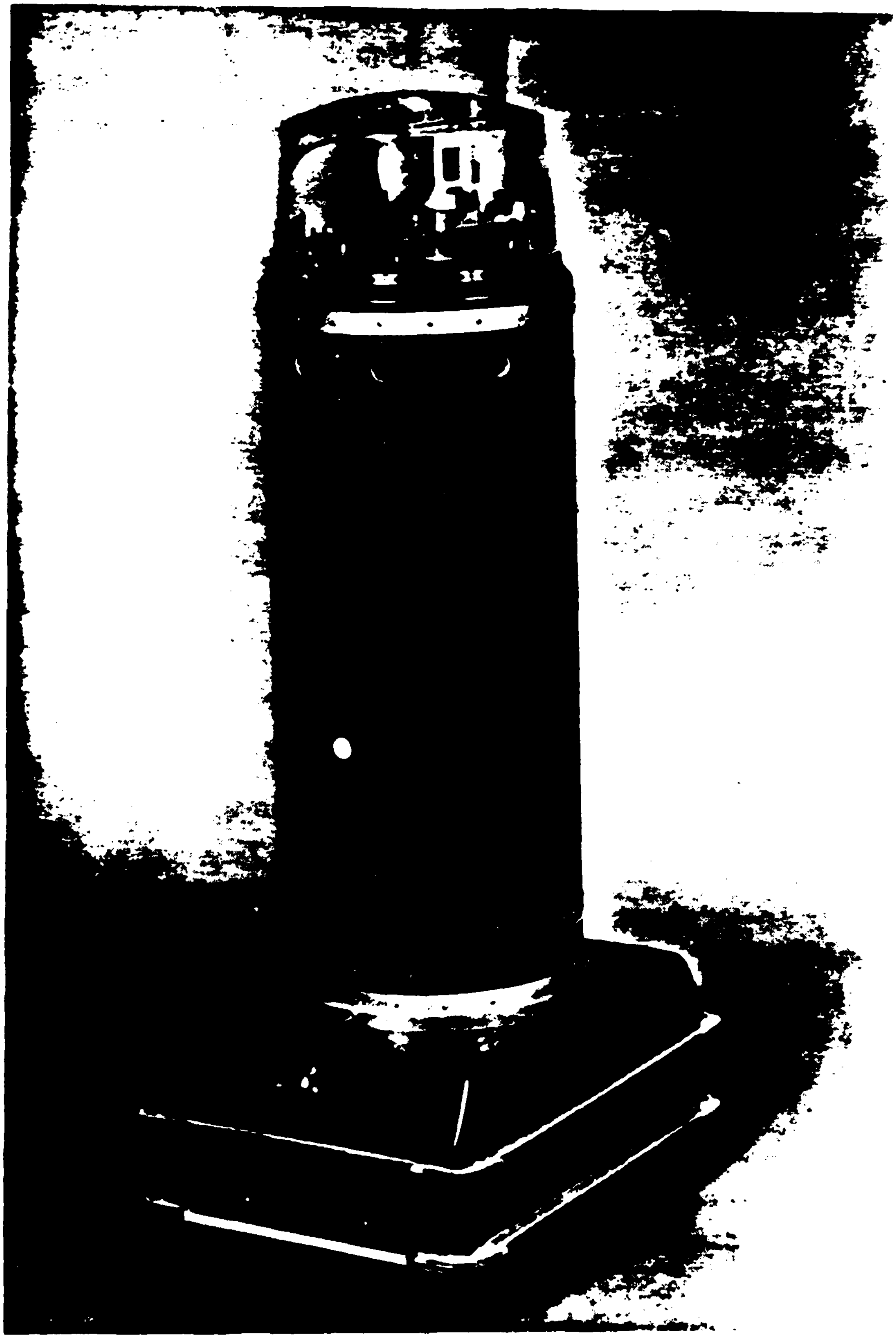


Figure 1-18: Robart II - A Front View

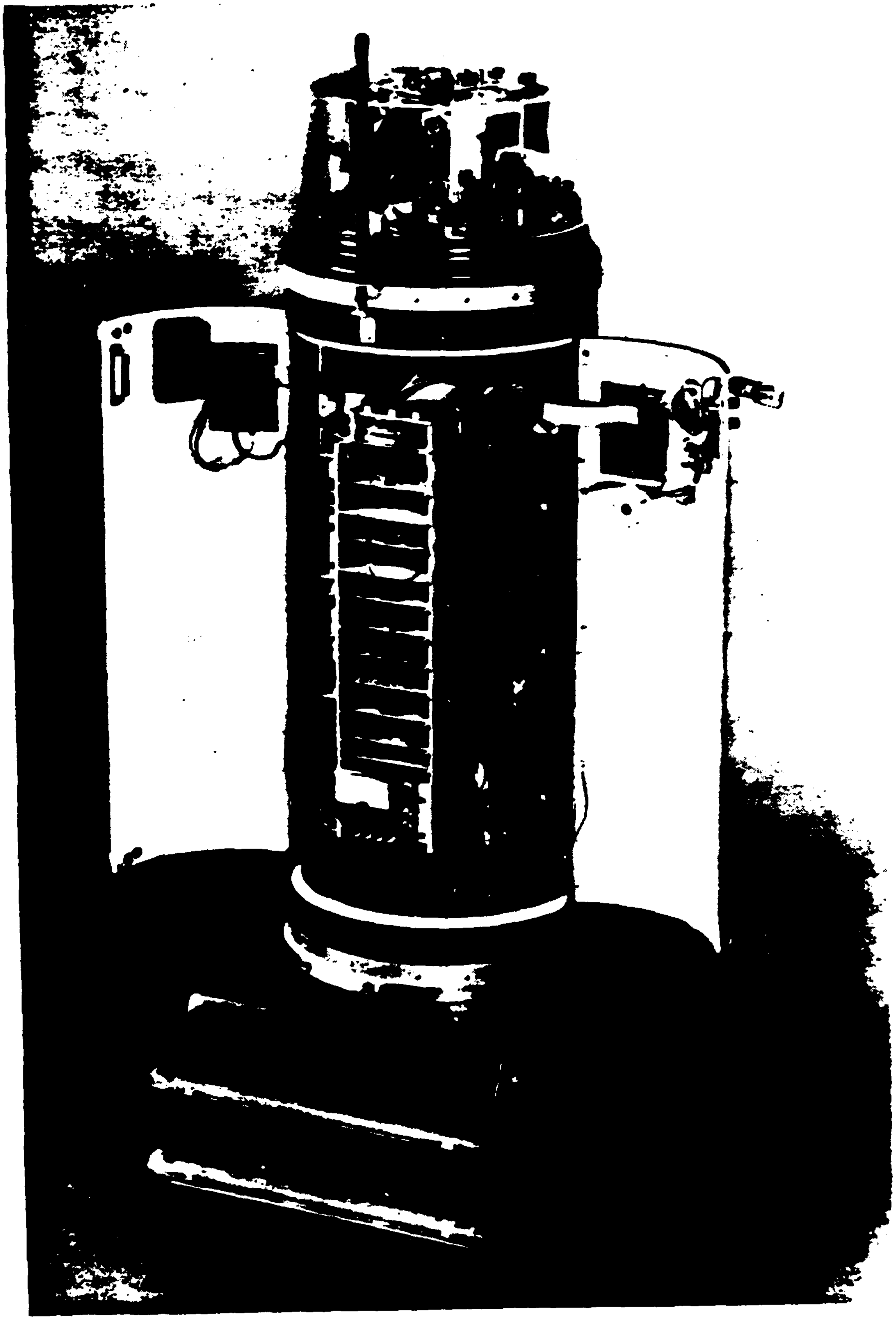


Figure 1-19:Rear View - Controlling Computers and Interface Electronics

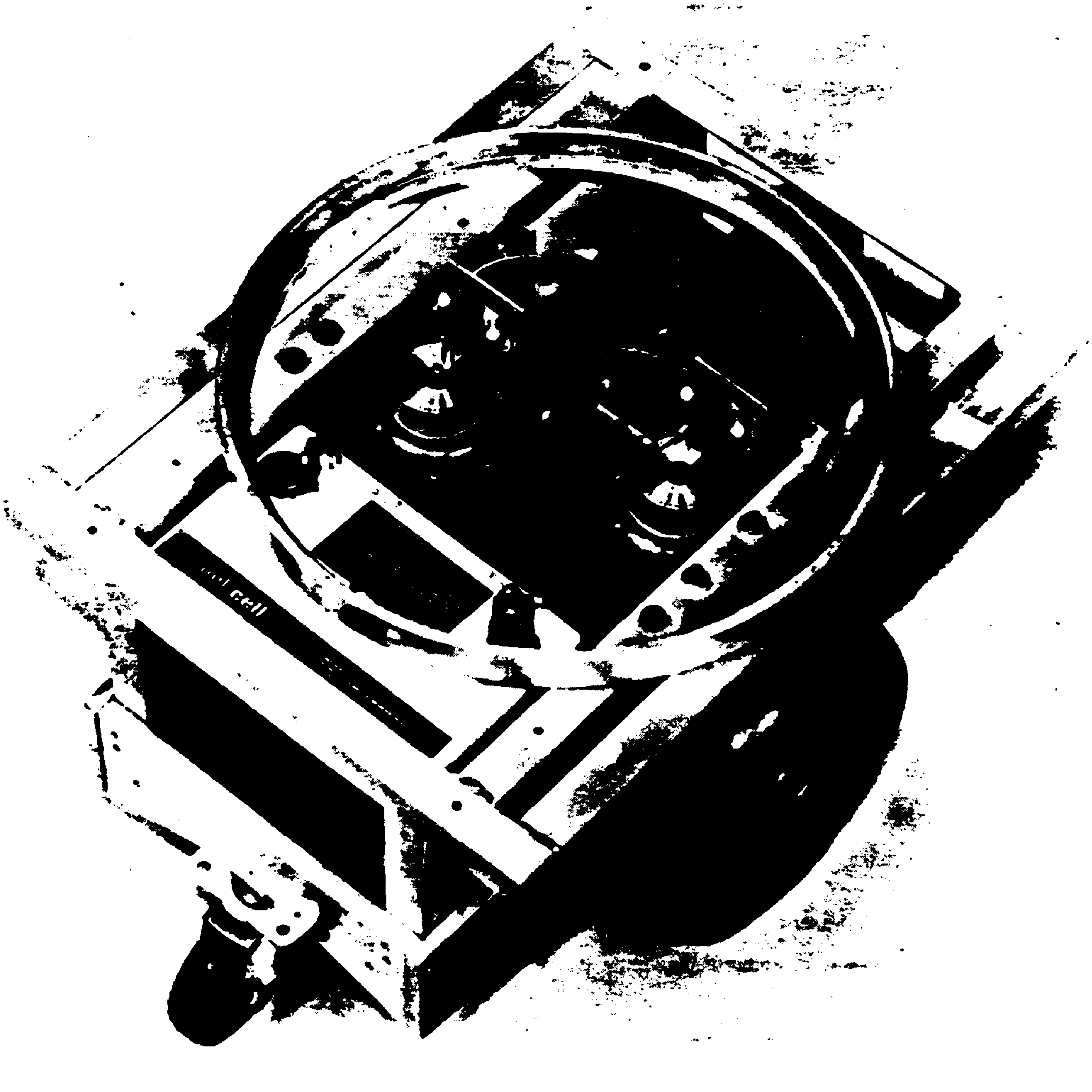


Figure 1-20:Drive Wheel Base

Figure 1-21:Low Level Computers and Sonar Transducer Driver Boards

modules through a special multiplexing circuit. Five of these units have their transducers arranged in a forward-looking array, with overlapping beam patterns. These transducers can be sequentially fired in any combination, as determined by the command from the top level controller. Associated ranges are then fed back up the hierarchy to the top level microprocessor. A sixth transducer is mounted on the rotatable head, positionable up to 100 degrees either side of centerline. The position and velocity of the head is controlled by another dedicated low level microprocessor. Figure 1-21 shows three of the low-level computers which control the head, sonars and drive motors, mounted on the right side of the card cage. On the outer fixture are the sonar transducer driver boards.

A fourth dedicated controller is assigned the function of controlling a DT-1050 microprocessor based speech synthesizer, and a future speech recognition. All low level controllers receive commands from the top level controller via an eight line parallel bus, and communicate information back up via a common serial interface. Figure 1-22 shows the top level processor, a SYM-1, mounted above the 6502 based computer used for controlling speech synthesis. The actual synthesizer board is mounted on the outer fixture.

Approximately 256 internal checkpoints will constantly monitor circuit performance, system configuration, operator controlled switch options, cable connections, distribution bus voltages, etc., with speech output generated by the self diagnostics to advise of any difficulties. A 1200 bit per second serial RF link will be available for telemetry, or specific overriding of commands from an observer located at a remote terminal.

The multitude of sensors combined with multi-processor control will give Robart II the capability to perform very sophisticated tasks.

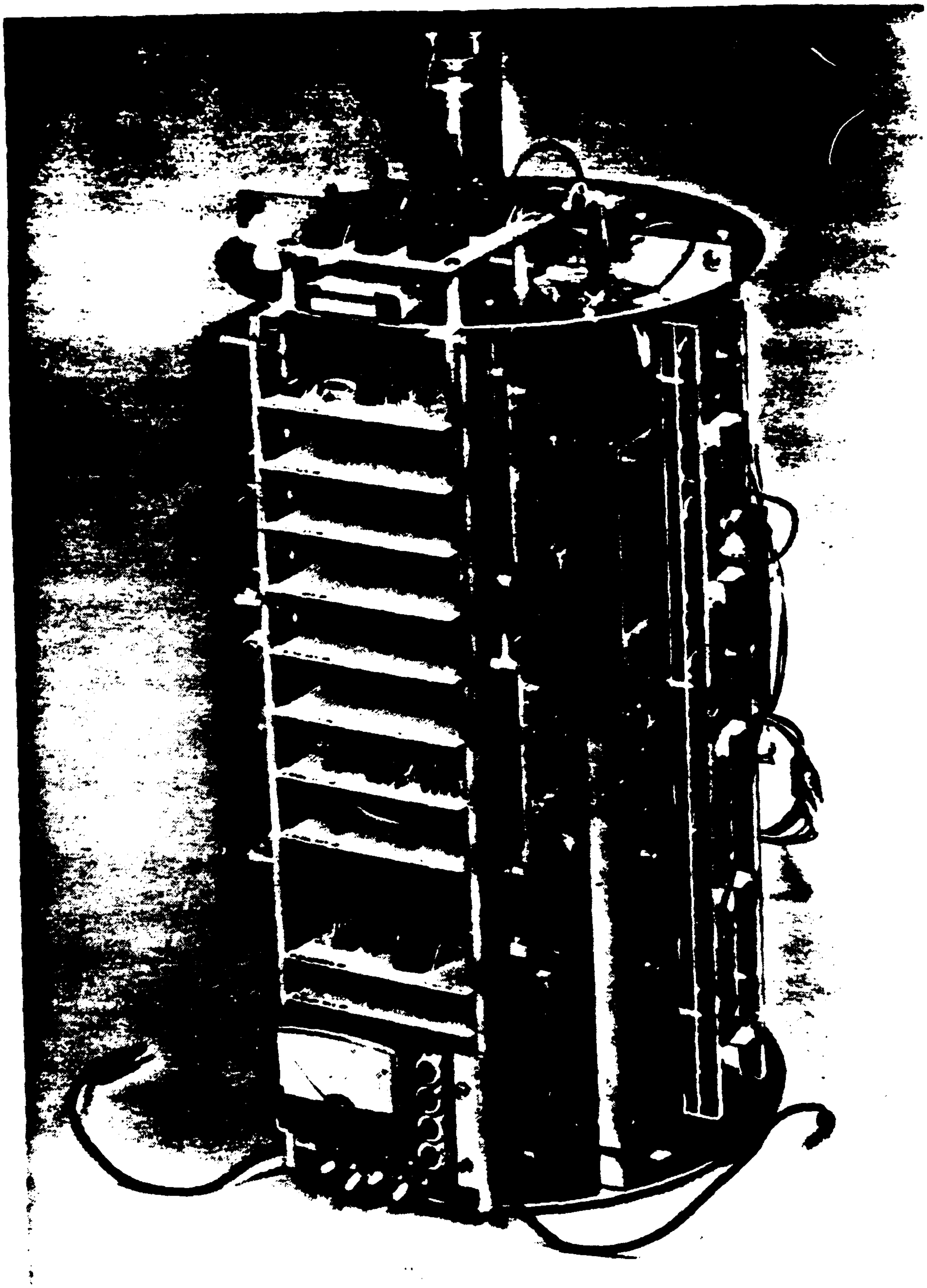


Figure 1-22:Top Level Processor and Speech Synthesis Computers

References

[Andersson 81]
Andersson, R. L.
Playing It By Ear: An Intelligent Sonar Guided Robot.
Master's thesis, University of Pennsylvania, May, 1981.

[Andersson 82]
Andersson, R. L. and Bajcsy, R. K.
SCIMR - A Test Bed For Real Time Processing of Sensory Data.
In *IEEE Computer Society International Conference on Pattern Recognition and Image Processing*. June, 1982.

[Bauzil 81]
Bauzil, G., Briot, M. and Ribes, R.
A Navigation Sub-System Using Ultrasonic Sensors for the Mobile Robot HILARE.
In *Proc. First Annual Conference on Robot Vision and Sensory Control*. Stratford-Upon-Avon, UK, 1981.

[Briot 81]
Briot, M., Talou, J. C. and Bauzil, G.
The Multi-Sensors Which Help a Mobile Robot Find Its Place.
Sensor Review :15-19, Jan, 1981.

[Coles 69]
Coles, S. L., Raphael, B., Duda, R. O., Rosen, C. A., Garvey, T. D., Yates, R. A. and Munson, J. H.
Application of Intelligent Automata to Reconnaissance.
Technical Report, Stanford Research Institute, Nov, 1969.

[Cooke 83]
Cooke, R. A.
Microcomputer Control of Free Ranging Robots.
In *Proc. 13th International Symposium on Industrial Robots and Robot 7*. Chicago, Ill., 1983.

[Dobrotin 77]
Dobrotin, B. and Lewis, R.
A Practical Manipulator System.
In *Proc. IJCAI-5*, pages 749-757. 1977.

[Dreyfus 79]
Dreyfus, H. L.
What Computers Can't Do.
Harper and Row, New York, 1979.

[Everett 82a]
Everett, H. R.
A Computer Controlled Sentry Robot: A Homebuilt Project Report.
Robotics Age, March/April, 1982.

[Everett 82b]
Everett, H. R.
A Microprocessor Controlled Autonomous Sentry Robot.
Master's thesis, Naval Postgraduate School, Oct. 1982.

[Ferrer 81]
Ferrer, M., Briot, M., and Talou, J. C.
Study of a Video Image Treatment System for the Mobile Robot Hilare.
In *Proc. First International Conference on Robot Vision and Sensory Controls*, pages 59-71. Stratford-Upon-Avon, UK, April, 1981.

[Giralt 77]
Giralt, G., Sobek, R. and Chatila, R.
A Muti-Level Planning and Navigation System for a Mobile Robot; A First Approach to HILARE.
In *Proc. IJCAI-6*. 1977.

[Harmon 81]
Harmon, S. Y.
Autonomous Free Swimming Submersible: A Demonstration of Autonomous Robotics Technology.
Technical Report, Naval Ocean Systems Center, Oct, 1981.

[Harmon 83a]
Harmon S. Y.
Coordination Between Control and Knowledge Based Systems for Autonomous Vehicle Guidance.
In *Proc. IEEE Trends and Applications Conference.* Gaithersburg, MD, May, 1983.

[Harmon 83b]
Harmon, S. Y.
Information Processing System Architecture for an Autonomous Robot System.
In *Oakland University Conference on Artificial Intelligence.* Rochester, MI, April, 1983.

[Jarvis 80]
Jarvis, R. A.
A Mobile Robot for Computer Vision Research.
In *Proc. Third Australian Computer Science Conference.* Canberra, Australia, Feb, 1980.

[Laumond 83]
Laumond, J.
Model Structuring and Concept Recognition: Two Aspects of Learning for a Mobile Robot.
In *Proc. of IJCAI-8*. 1983.

[Lewis 73]
Lewis, R. A. and Bejczy, A. K.
Planning Considerations For A Roving Robot With Arm.
In *Proc. IJCAI-3*, pages 308-315. 1973.

[Lewis 77]
Lewis, R. A. and Johnston, A. R.
A Scanning Laser Rangefinder For A Roving Robot With Arm.
In *Proc. IJCAI-5*, pages 762-768. 1977.

[Miller 77]
Miller, J. A.
Autonomous Guidance and Control of a Roving Robot.
In *Proc. IJCAI-5*. Cambridge, MA, 1977.

[Moravec 81]
Moravec, H. P.
Robot Rover Visual Navigation.
UMI Research Press, Ann Arbor, 1981.

[Moravec 83]
Moravec, H. P.
The Stanford Cart and CMU Rover.
Technical Report, Robotics Institute Carnegie-Mellon University, Feb, 1983.

[Nilsson 69]
Nilsson, N. J. and Rosen, C. A.
Application of Intelligent Automata to Reconnaissance.
Technical Report, Stanford Research Institute, Feb, 1969.

[Raphael 68]
Raphael, B.
Programming a Robot.
In *Proc. IFIP Congress 68*. Edinburgh, Scotland, Aug, 1968.

[Rosen 68]
Rosen, C. and Nilsson, N.
Application of Intelligent Automata to Reconnaissance.
Technical Report, Stanford Research Institute, Jan, 1968.

[Tachi 81]
Tachi. S.. Tanie. K.. Komoziya. K.. Hosoda. Y. and Abe. M.
Guide Dog Robot--Its Basic Plan and Some Experiments with MELDOG Mark I.
Mechanisms and Machine Theory 126:21-29, 1981.

[Thompson 79]
Thompson. A. M.
The Navigation System of the JPL Robot.
In *Proc. IJCAI-5*. pages 335-337. Tokyo, 1979.

END
9-87
DTIC