SRI International

THE SRI MOBILE ROBOT TESTBED
A Preliminary Report

Technical Note 413

February 1987

By: Stanley Reifel
Research Engineer

Representation and Reasoning Program
Artificial Intelligence Center
Computer and Information Sciences Division

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This research has been made possible in part by FMC Corporation under contract 147466 (SRI Project 7390), and in part by General Motors Research Laboratories under contract 50-13 (SRI Project 8662).

SRI International

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

ABSTRACT

This paper describes a mobile robot designed for experimentation in artificial intelligence (AI). Presented here are details of the robot's hardware and software architecture. The robot is driven by two electrically powered wheels. On-board computers control the motors and visual and ultrasonic sensors. A library of low-level software provides primitive functions for high-level programs to interface with the robot's sensors and effectors.

Introduction

One of the goals of research in artificial intelligence is to develop information-processing techniques that can be applied to systems that interact with their physical environment. Sometimes it is possible to investigate this problem using simulation and modeling methods. Often, however, it is desirable to experiment with a fully integrated, functioning system that has perceptual and motor capabilities. In order to study this area better, the AI Center of SRI International is currently engaged in a program of basic research on intelligent mobile robots, including the design and construction of a mobile robot that will serve as a testbed for experimentation. In this paper, I describe the hardware and low-level software of the robot, and its interface to higher-level programs.

Mechanical Architecture

The robot stands 35 inches tall and weighs 300 lb. The cross section of its body is octagonal. A cylindrical shape would allow the robot to rotate about its center without the risk of colliding into nearby objects; however, it is difficult to package rectangular equipment efficiently in a cylindrical volume, and a lack of planar faces would make mounting sensors and controls accessible from the vehicle's outside inconvenient. The octagonal cross section is practical from a packaging point of view and retains much of the maneuverability offered by a cylindrical form. Its width and depth are 24 inches, large enough to house the necessary components and narrow enough to navigate through doorways.

The robot's body is built in two modular levels or tiers, with additional equipment mounted on the upper surface. The multi-level architecture maximizes the area for mounting the internal parts and provides a flexible environment for changes and additions. The height of a tier can be easily increased or new tiers can be installed in a modular fashion. A single "spinal cord" of cables connects the tiers together electrically. The eight faces around the perimeter of each tier perform several functions. Structurally, they serve to support the layer. The exoskeleton design is advantageous because none of the inner volume is obstructed with support members. The rear face of each tier is used to attach all the connectors, controls, and indicators that need external access by the equipment housed in that layer. Each rear panel is hinged to an adjacent face. The cables that connect to the panel components are grouped along the hinge, making access behind the rear panel simple. The side faces also help to give the robot a clean appearance. Four views of the robot are shown in Figure 1.

The bottom tier of the robot is the locomotion platform. It contains the two motors, gear-train assemblies, optical encoders, and the servo amplifiers. Two large batteries and a pair of DC-to-DC power supplies are also mounted in this area. All of the heavy items are on the bottom tier, so the robot is very stable because its center of gravity is near the ground.

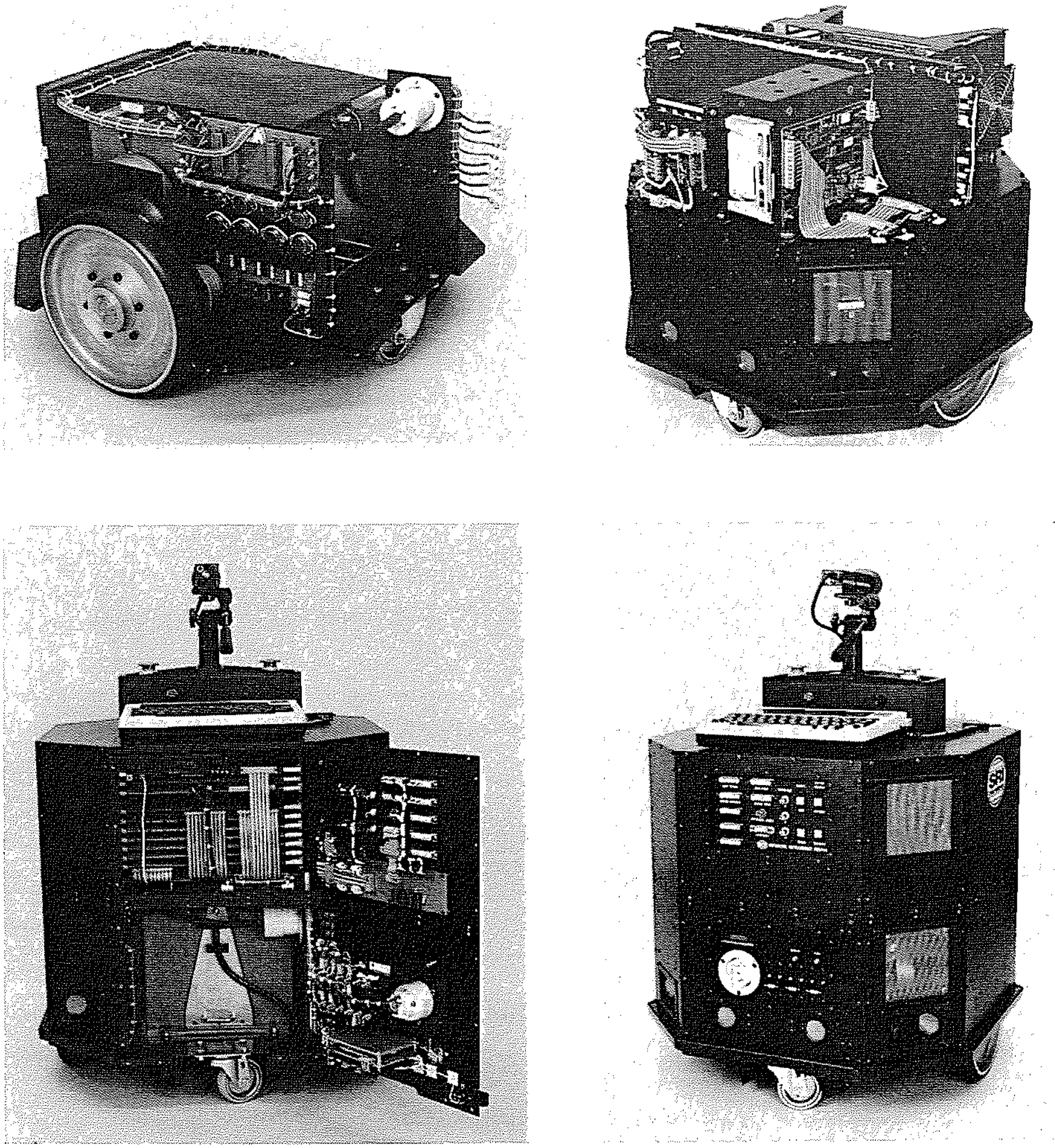

Figure 1: Four views of the robot

Attached to the outer faces of the bottom tier are 12 ultrasonic transducers and 8 bumper bars linked to microswitches.

Enclosed in the upper tier are the computers. Two computers working at different levels divide the task of controlling the robot. A small Z-80 based computer performs the low-level sensor/effector functions, including controlling the motors that drive the vehicle, coordinating the ultrasonic devices, and monitoring the bumpers and battery voltage. The second computer is a Sun Workstation that has been repackaged into a form suitable to fit in the robot. It is divided into two subcomponents: a 12-slot multibus card cage housing ten PC boards and a Winchester disk drive. Special precautions have been taken to ensure that the hard disk will operate reliably and avoid damage from vibrations and collisions; the drive has been mechanically isolated from the robot using rubber shock mounts manufactured by Barry Controls. The disk orientation was selected so that the head is least likely to collide with the platter. It is positioned with the plane of the platter aligned front to back and up and down with respect to the robot. Thus only a side-to-side motion can cause the head to crash into the disk. These precautions seem effective, because the disk drive functions reliably. Mounted on the upper surface of the second tier is a LCD terminal, a television camera, the speaker for the speech synthesizer, and two emergency stop buttons.

The body is supported by four wheels; two are driven by motors and two are free. The powered wheels are located on the right and left sides of the robot. One motor on each side drives a 10 inch wheel through a 25:1 gear reduction. An optical encoder is geared to each wheel, providing position and velocity feedback to the motor controller. It would have been easier to implement an open-loop system driven by stepper motors, but servoed DC motors offer several advantages. Stepper motors with sufficient torque to propel the robot would have been too energy-inefficient. A closed-loop control system provides better accuracy and allows the computer to detect whether or not the vehicle is moving as expected. The two free wheels are located in the front and rear of the robot. These casters are mounted on a spring-loaded suspension system. The suspension guarantees that all four wheels support the robot when it moves on uneven surfaces such as bumps and ramps.

This wheel configuration allows the robot to move forward and backward and along curved paths and to rotate about its center. Unlike other designs, however, the robot cannot move from side to side. The most common method used by omnidirectional vehicles is a three-wheel configuration, in which the steering mechanism couples the wheels together so they always point in the direction of travel. The robot is maneuvered by rotating the three wheels until they point along the desired trajectory and then driving them forward. This design is not truly omnidirectional because it has only two degrees of control and seems to offer little advantage over the differential drive method we use. Furthermore, because the vehicle cannot rotate about its center, a pivoting sensor platform with slip rings is needed to conduct electric signals to the base. Another approach that does achieve three degrees of freedom uses three omnidirectional wheels equipped with rollers mounted along the perimeter of the wheels. While this design provides more freedom of motion, it does so at the expense of additional complexity and bulkiness. The wheel and roller assemblies are mechanically complicated and the control problem is more difficult as well. This type of robot with a triangular footprint uses its interior space inefficiently. No central area low-to-the-ground remains for mounting large batteries and power supplies. We believe that the benefits of the simplicity of our design seem to outweigh the advantages gained from omnidirectional motion.

Electrical and Computer Architecture

The robot is powered by two large electric vehicle batteries with a capacity of 85 Ah at 24 V. These old-technology wet cells are used because our experience has shown that they perform better than the more modern sealed gelcells. This single source powers both the motors and the computers. For efficiency, the motors run directly off the raw 24 V; power for the computers is conditioned by six Converter Concepts 100-W DC-to-DC switching power supplies. This system provides about two hours of continuous operation. For longer debugging sessions, an external battery charger can be connected that can power the robot indefinitely.

To propel the robot, two PMI U12M4LR flat armature DC motors are used. The servo amplifier drives the motors with pulse-width modulation at 15.6 kHz, a frequency high enough to be inaudible. As a safety precaution, the motor power is also controlled by a relay that must be active for the motors to function. The relay can only be activated by manually pressing the high-power-enable button. The relay can be disabled by several sources: any of three kill buttons, either of the two computers, or a watchdog timer that detects if the motor controller is inoperative.

The servo amplifier switches the motor power based on signals received from the motor controller. The motor controller consists of four cards in a STD card cage. The CPU board is an off-the-shelf product manufactured by Prolog that includes a Z-80 microprocessor, RAM, ROM, and serial I/O ports. Three custom I/O cards are used to interface to the robot's sensors and effectors. Two of the cards provide the circuitry that generates the pulse width modulated signals that drive the motors, decode the outputs of the shaft encoders and monitor the bumper switches. Also, a hardware floating point processor is included to speed up the servo control calculations, and an analog to digital converter is used to measures the battery voltage. The third card is an interface to the ultrasonic sensors.

Twelve ultrasonic sensors provide range measurements to nearby objects. The transducers are mounted six inches up from the bottom of the robot. Because most common obstacles rest on the floor, placing the sensors low and pointing them in a plane parallel to the ground increases the ability to detect these objects and thus avoid collisions. The transducers are arranged with four pointing forward, equally spaced apart to cover the full width of the robot. Four are similarly mounted in the back. Each side of vehicle has two transducers aimed perpendicular to the direction that the robot moves, one forward of the wheels and one behind. These side-mounted sensors are useful for following walls and detecting doorways in halls. One reason for choosing this configuration is to reduce erroneous measurements cause by specular reflections. Because the robot often maneuvers in a rectilinear world and much of the time it is aligned with that world, this placement of the transducers points them at nearly right angles with the surfaces in the environment.

The ultrasonic electronics is organized in a network consisting of one host module and six slave modules. Each slave module has its own single-chip microprocessor, two Texas Instrument ultrasonic analog boards and a pair of Polaroid instrument-grade transducers. The slaves are responsible for controlling the analog electronics, measuring the time of flight of the sound waves, and communicating with the host over a common ribbon cable bus. The host module serves as an interface between the slaves and the motor controller. This networked architecture is used for several reasons. By mounting the analog electronics near the transducers, the coaxial cable that connects them together can be very short; this reduces the possibility of inducing motor noise into the ultrasonic signal. Having a single digital bus connecting the slaves with the host module simplifies the wiring and improves reliability, since an error correcting protocol is used. Because the host and slave modules each have their own microprocessor, the work of controlling the ultrasonics is distributed rather than performed by the motor controller.

After the initial installation of the ultrasonic system in the robot, we found that motor noise caused false echos and, therefore, faulty range measurements. This problem was solved by reducing the noise generated by the motors and shielding the ultrasonic analog electronics. The noise is attenuated by small capacitors across the motors and the servo amplifier, and by high-frequency ferrite shielding beads that are placed on the wires leading to the motors. Small aluminum plates are mounted near the circuit side of each pair of ultrasonic analog boards to provide additional shielding for the analog electronics from the remaining noise.

The ultrasonic modules receive instructions from the motor controller, and the motor controller is directed by the on-board Sun through a fast serial line. The Sun is a general-purpose off-the-shelf computer that runs the UNIX operating system. The hardware is designed around the Intel multibus, which is advantageous because a rich variety of peripherals are available for it. The CPU is a 68010 microprocessor with four megabytes of physical memory. The Sun includes three types of I/O devices: a 42-megabyte Micropolis hard disk, an Ethernet controller, and six RS-232 serial ports.

Augmenting the standard Sun hardware, we have added additional I/O devices. The video electronics consists of an Imaging Technology video digitizer, a Sony CCD camera, and a Sony Watchman monitor. The digitizer has a resolution of 512 by 512 with 8-bit pixels. Black and white images and graphics can be displayed using the on-board Watchman CRT or pseudo-color images can viewed with an off-board RGB monitor. A Prose 2000 speech board gives the robot the ability to synthesize speech. This a single Multibus board that accepts standard text from a serial line and uses a synthesis-by-rule algorithm to generate speech.

An LCD lap-top computer is mounted on top of the robot to serve as the console for the Sun Workstation.

Low-Level Software Architecture

The lowest-level software that drives the robot runs in the motor controller and is written in Z-80 assembly language. Implemented here are two PID servos that control the right and left wheels independently. Fifty times per second, a new power level is computed for each wheel, based on the values from the optical encoders and several parameters such as desired velocity and acceleration. Also during this cycle, a packet of status information is built and sent out over a serial line to the Sun. This continuous stream of data reports the robot's velocity, position, battery voltage, sonar data, and error codes. Because the Sun is constantly receiving information about the robot's state, little delay is introduced by acquiring this data.

The Sun instructs the motor controller by sending it command packets over the same serial line. These command packets are used to set the motor controller's internal parameters, to stop and start the robot motion, and to poll the ultrasonic sensors. The parameters that can be set include the rate of acceleration, the velocity, and the feedback constants used by the servo algorithm. Two methods can be used to control the robot for discrete movements and continuous motion. For discrete movements, a command instructs the right and left wheels to each move a given distance and then stop. This primitive is used for moving along a straight path or a circular path, or for rotating about the robot's center. A separate command is used for continuous motion, which takes signed velocities as arguments. This method is typically used when the robot is servoed from ultrasonic or visual data.

The Sun uses a special character handler that collects the status packets and sends new commands to the motor controller. Each character that is received from the motor controller is added to the packet currently being built. When a full packet is collected, its integrity is verified by comparing the checksum of the data with the expected value. After a valid packet is received, the robot's X, Y, theta location is computed by integrating the change in position of the right and left wheels. This information, along with the packet received, is then made available to the user process. Commands to the motor controller from the user process are queued by the character handler and transmitted in the background. After the motor controller receives a command, it acknowledges whether the packet was received incorrectly, causing the Sun to retransmit if necessary.

All of the robot's I/O devices are accessed by user programs through calls to three libraries. One library interfaces to the motor controller through the character handler, supporting all the motion control and ultrasonic functions. Ultrasonic measurements can be requested two in different ways: the Sun can poll each sensor individually, thus allowing random access, but the propagation delays of this method, limit the total number of ultrasonics readings per second to five. Alternatively, the Sun can provide the motor controller with a sensor polling pattern. The motor controller then continuously samples the defined transducers and includes the results along with a time and robot position stamp in the outgoing status packet. This system increases the sampling rate to 16 readings per second. A combination of both methods may also be used.

A second library has a set of procedures that are used to interface with the speech server. The speech server is organized with three queues for speech output. The queues are prioritized, with the highest priority being for emergency error messages, the middle for normal messages, and the lowest for debugging information. The speech server runs as an independent process that polls the three queues and drives the Prose 2000 synthesizer. The third library provides primitive functions to control the video hardware, capture images, and display graphics.

Programming Environment

The robot's programming environment centers around the UNIX operating system used by the on-board computer. UNIX offers several advantages over a simple kernel. Properties such as virtual memory and multiple processes are useful when developing large, complex systems such as this. UNIX provides debugging facilities and supports numerous programming languages. Having an on-board disk and a file system allows the robot to be self-contained by keeping all of its programs and data files resident. Sun UNIX includes Ethernet support, making communication with other computers simple and fast.

Programs written to control the robot can execute on its internal computer or on larger off-board computers. Programs targeted to run on the robot directly are edited and compiled on a Lisp machine or Sun Workstation. Executable code is downloaded to a file on the robot using the Ethernet. The net is then disconnected, and the robot can load and run programs saved on its disk. Debugging these as well as most real-time programs is often very difficult. We have found three of our I/O devices to be useful as debugging aids. Having a fast disk allows logging large amounts of data on-the-fly that can then be analyzed after completing an experiment. The graphics hardware has proven to be very useful for displaying sensor data in real-time using a format that is easy to visualize, such as points, line segments, and processed images. When the robot is running, reading the console output in addition to watching the graphics display and the robot itself is difficult. By directing console messages to the speech synthesizer, programs can explain what they are doing in a manner that is not distracting.

Projects that require faster computers or different programming environments can make use of the robot by controlling it remotely from an off-board computer. With an Ethernet tether to the robot, other computers can make remote procedure calls to all of the primitive functions defined in the robot's I/O libraries. From a Symbolics Lisp machine, about 40 remote calls can be made to the robot per second, fast enough to do simple servoing. This method of controlling the robot offers its advantages at the expense of slower access to the sensors and effectors and sacrifices in autonomy.

Acknowledgments

Stanley Rosenschein has provided considerable high-level guidance throughout the development of this project. The contributions of Leslie Kaelbling and William Wells to the overall design of the robot and specifically to the robot interface libraries have proven invaluable. The insights of Arthur Crittenden and Ernest Lantz aided in solving many of the special mechanical problems.