Buzz: An instantiation of a schema-based reactive robotic system

Ronald C. Arkin, Tucker Balch, Thomas R. Collins, Andrew M. Henshaw,
Douglas C. MacKenzie, Elizabeth Nitz, David Rodriguez, Keith Ward
Mobile Robot Laboratory, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA 30332-0280

Abstract

The Georgia Tech entry to the AAAI Mobile Robot Competition, a schema-based reactive robotic system, is described. New developments are presented including the introduction of two novel behaviors *probe* and *avoid-past*, specialized planning and sensing strategies, and a transputer implementation of the reactive control system.

1. Introduction

In July of 1992, a robot competition was sponsored by the American Association of Artificial Intelligence in San Jose, California [4]. A robot was required to perform three tasks. Phase 1 of the competition required a demonstration that the robot was able to effectively maneuver among both obstacles and humans in a highly cluttered and unmodeled environment. This environment consisted of a ring approximately 80 feet in diameter littered with a large number of cardboard boxes of various sizes. An enclosing wall roughly 3 feet high defined the arena. Phase 2 of the competition required the robot to locate and move to 10 target poles that were placed within this environment. These target poles were permitted to have markings or beacons placed upon them by the contestants. Phase 3 was to have the robot navigate to three specific target poles (from the original 10) which the judges selected at their discretion.

This paper describes the hardware and software aspects of the Georgia Tech entry, Buzz (named after the school mascot), a Denning MRV-3 robot. The robot incorporates ultrasonic sensing, computer vision, and IR beacons, while the control strategy is largely reactive. Two new motor behaviors, *probe* and *avoid-past* were developed and used for the first time in the context of the competition. A specialized transputer architecture, ANIMA, was also fielded for the first time in Phase 1 of the competition.

2. System Design

Figure 1 shows the overall system used during the competition. The competition pole was a 3.5" PVC pole which stands vertically and labeled by each team as needed. Our pole was identified by retro-reflective tape strips and IR beacons. The retro-reflective tape strips were used to provide both distance to the pole and the individual pole identification (see Sec. 4). A second system for recognition of the pole was the signal transmitted by two rotating infrared (IR) beacons located at the top of each pole. The IR signal provided a unique identifier in addition to the location and distance of the robot to the pole (Sec. 4).

A Denning MRV-3 mobile robot was used and equipped with shaft encoders and a ring of ultrasonic sensors around the circumference of the robot. Additional equipment, (IR camera, high-intensity light source, CCD camera, UHF imaging communication, and LAWN communication), was added. The light source illuminated the retro-reflective tape for the CCD camera to send to the off-board host system through the UHF video communication link. The IR camera sensed the beacon signal and transmitted the resulting ID and distance through the

Figure 1: Components of Overall System

LAWN radio communication link. The shaft encoder and ultrasonic data were transmitted via the LAWN as were commands from the off-board host to the robot.

The off-board base system consisted of a Sun SparcStation which receives input from the LAWN and UHF imaging communication. The host generated the appropriate robot commands and displays the poles which have been successfully recognized. For Phase 1 of the competition, an on-board transputer system was used instead of the off-board host eliminating the need for the UHF/LAWN links (Section 5).

3. Planning and Motor Control

Schema-based navigation is a form of reactive control that decomposes actions into a set of multiple concurrent processes called motor schemas. Each of these active behaviors generates a velocity vector based upon immediate sensory data which is combined with the outputs of the other behaviors. The result is transmitted to the robot for execution. For more details on schema-based navigation see [1].

Buzz's overall "plan" is depicted in Figure 2. For Phase 1, safe wandering, control remained in the *wander* state. Four motor schemas were active in this state: *Noise*, a random vector which acts like "grease" for this type of navigation; *Avoid-static-obstacle*, which pushes the robot away from detected objects; *Probe*, an open space attractor behavior used to encourage the robot to explore; and *Avoid-past*, a behavior which repulses the robot away from places it has recently been.

Phase 2, moving to discernible target poles, incorporated the *move-to-goal* behavior which attracts the robot to the recognized target. The system started by choosing a goal. The robot completed a full sweep with the camera, remembering the locations of all visible poles. The nearest unvisited one was then chosen as the goal. If no unvisited poles were visible, the robot would choose one it had previously seen. If there were no unvisited poles in memory, the robot went into a wander mode for a short period of time to gain a different vantage point and then conducted another visual search.

Once a pole had been selected as the goal, visual servoing was used to center the pole in the camera's field of view. This provided accurate heading information without calibrated optics. The system then initialized *move-to-pole* (*move-to-pole* is short for the *move-to-goal*

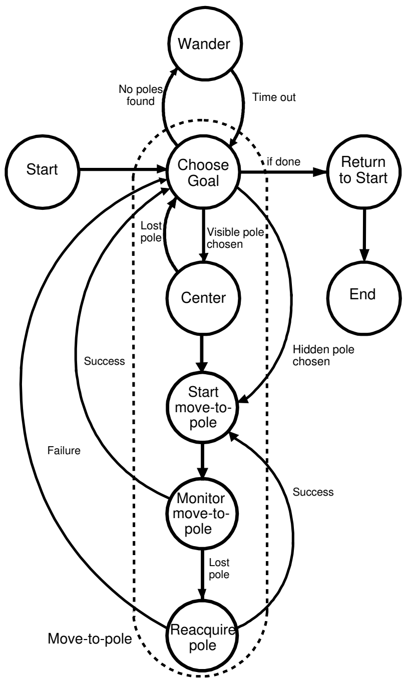


Figure 2: Schema Assemblage FSA

motor schema instantiated with the *find-pole* perceptual schema). If during execution *move-to-pole* lost the pole, a re-acquisition state was entered. The robot made a sweep with the camera, searching for the pole. If it was unable to locate it, the pole was abandoned and a transition to the *choose-goal* state was made.

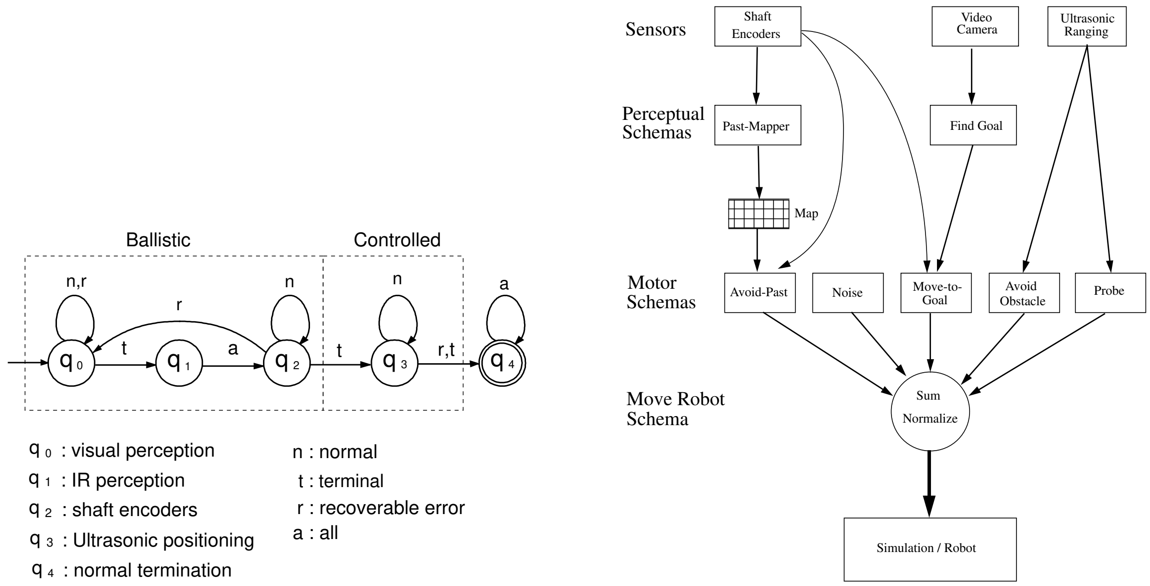
The operation of *move-to-pole* is depicted in Figure 3a. This diagram shows the different perceptual states that move-to-pole operated in as it executed. In state q_0, a visual search for poles was conducted. In state q_1, a confirmation search using the IR beacons executed. During state q_2, the robot moved towards the selected pole. State q_3 denotes final positioning in *move-to-goal* using the ultrasonics. State q_4 is the success state, which signaled completion. Figure 3b shows a data-flow diagram of the system moving towards the pole.

The *avoid-past* motor schema repelled the robot away from areas where it had already been (Sec. 3.1.2). It was used to remove the robot from local minima. It was configured with a gain of 0.1, and a grid size of 0.5 feet. The floor and ceiling on the strength of the output vector were set at 0.5 and 1.3 respectively. During each cycle, 10 grid squares were marked in the map as having been visited. To generate the output vector, the 20 grid squares surrounding the robot were evaluated for the number of visitation markings.

The *noise* motor schema provided some randomness to the behavior. This was used to overcome local minima. It was configured with a gain of 0.2 and a persistence of 3 cycles.

The *move-to-goal* motor schema attracted the robot to the goal location based on the robot's position encoders and a stored goal location. At a threshold distance from the goal location, a transition was made to a final positioning motion. The final positioning was controlled based on distance readings from the ultrasonic sensors. For the competition, *move-to-goal* was configured with a gain of 1.4, a transition to final positioning at a distance of 5.0 feet from the goal, and a final parking distance of 2.5 feet from the pole.

The *avoid-static-obstacle* motor schema repelled the robot away from obstacles detected with the ultrasonics sensors. It was configured with a gain of 1.7, a sphere of influence of 3.0 feet, and a minimum distance (within which the output was virtually infinite) of 0.9 feet. The *probe* motor schema was not used in Phases 2 or 3 due to software problems during porting



(a) *move-to-pole* control FSA (b) Data-flow diagram for *move-to-pole*

Figure 3: *move-to-pole*

from the transputer version.

Phase 3 required moving to several poles in a sequence directed by the referees. The *choose-goal* state in Figure 2 was modified to search only for the desired pole. The *wander* state was also modified to conduct a directed wander. This added a directional tendency to the wander behavior based on positional information acquired during Phase 2. If the desired pole was not visible, this directed wander tended to move the robot closer to the position the pole was found in Phase 2.

3.1 New Behaviors

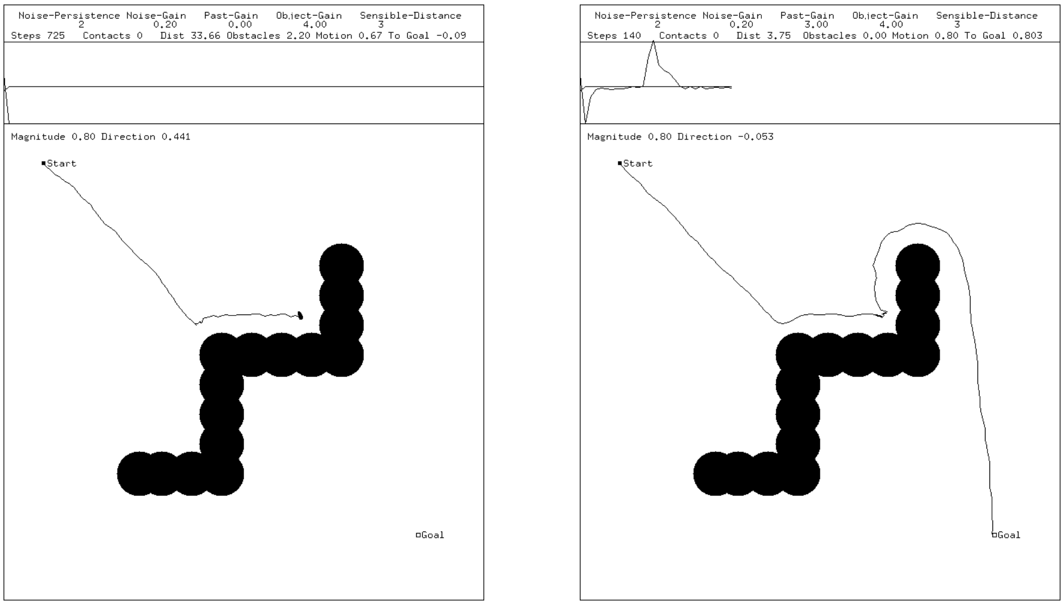
3.1.1 Probe

A new schema, *Probe*, was developed to enhance the performance of the robot in cluttered environments. Probe encourages motion toward open areas by sampling sets of adjacent ultrasonic sensors and discarding directions that are blocked. The sampling starts with the sensors most closely aligned with the current direction of motion and continues until an open path is detected. If an open corridor is found, a field that is proportional to the angle of the corridor with respect to current motion is returned. This schema augments wandering behavior while providing escape route searching when the robot becomes blocked.

3.1.2 Avoid-Past

A basic navigational behavior may be generated by instantiating the *Move-to-goal*, *Avoid-static-obstacle* and *Noise* motor schemas. Unfortunately, this behavior is not robust enough to solve some of the navigational problems posed in the AAAI competition. Figure 4a shows how a simple box canyon can foil the strategy. Buzz's navigational behavior was improved with the addition of an *Avoid-past* motor schema. This improved system is able to navigate more efficiently and escape from box canyons (Fig. 4b) [2].

Avoid-past is a motor schema which, combined with *Move-to-goal*, *Avoid-static-obstacle* and *Noise*, forms a robust schema-based navigation system (Figure 3). *Avoid-past* uses a



(a) Without *Avoid-past* (b) With *Avoid-past*

Figure 4: (a). Navigation in a box canyon without *avoid-past*. The goal is in the lower right corner. The robot's path is indicated by the black line. In this case, the robot becomes trapped. (b). Navigational behavior improved with *Avoid-past*.

simple spatial memory to compute a vector away from areas that have already been visited. The more often an area has been visited, the stronger the repulsive force generated.

Avoid-past operates in conjunction with the perceptual schema *Past-mapper*. At each time step *Past-mapper* updates a spatial map which **Avoid-past** references to compute a vector away from the "most-visited" areas. This vector is combined with vectors generated by *Move-to-goal*, *Avoid-static-obstacle* and *Noise* to determine the robot's heading and speed for the next time interval.

In Phase 2 of the AAAI Mobile Robot Competition Buzz encountered a box canyon. It had chosen a pole which was blocked by two boxes forming a "V." The *Avoid-past* schema successfully forced Buzz out of the canyon and around the obstacles.

4. Perception

Four different sensor sources were used: ultrasound, shaft encoders, vision, and infrared beacons. The ultrasonic sensors were used for obstacle avoidance as in our previous work [1]. Obstacle location was detected resulting in a repulsive field. Ultrasound was also used for final positioning of the robot relative to the pole in Phases 2 and 3. Positional information regarding the location of poles was stored using shaft encoder data during Phase 2 for subsequent use during directed wandering in Phase 3. The vision and IR sensing are described below.

4.1 Vision for Pole Identification and Location Determination

Our objectives were to be able to locate poles from a distance, judge how far away they were, and identify each pole as distinct from the other ten in real-time. Since the ultrasonic sensors do not have a long range, vision was decided to be more practical for accomplishing these goals.

Having made the decision to use a black and white CCD video camera for image processing, a simple way of finding pole location, distance, and identity was needed without having to extract a copious amount of information from each scene. Since, image processing tends to be slower than other sensor processing, a goal was established to minimize the amount of time spent analyzing each picture. The idea to use retro-reflective tape markings on each pole came from the ease and speed of bright spot detection in video images, and the range that the tape is visible with even a small light source. Lab test results indicated that, with a three-inch halogen lamp placed directly above the camera, the retro-reflective tape could be seen up to 60 feet away. This covered the first objective by making poles easy to locate from almost anywhere in the competition arena.

To determine the distance, a second stripe of tape was added to each pole so that the difference in the position of the new stripe relative to the first within the image could be calculated in pixels. Taking advantage of the effect of perspective, a look-up table was created with the actual distance to the pole corresponding to the number of pixels separating these two stripes in the image. This was tabulated, maintaining an accuracy within a half a foot, out to a distance of 60 feet. A similar procedure was used to determine the angle of the pole relative to the front of the robot, i.e., how far the robot would have to turn in order to be directly facing the pole. This was done by measuring the distance in pixels from a pole to the center of the image, then measuring the actual angle that it represented, and placing them in another look-up table.

The final objective was to label each pole as a separate entity from the rest. Adding a third tape stripe to each pole accomplished this. The space between the top and bottom stripe was kept consistent over all poles so that distance could be calculated accurately, but the third stripe was placed between the two at different intervals on each pole. The unique identification was the ratio between the top and middle, and the middle and bottom tape stripes (See Figure 5). A distinct striping pattern was created for each of the 10 poles.

The integrated algorithm used in the competition started by searching in a band at the horizon for light spots of a certain brightness and size. When a candidate spot was detected, the search proceeded vertically in an effort to find two more high-intensity spots. Only when three spots were in a vertical line were they assumed to signify a pole. Once the robot recognized a pole, it calculated the distances between the tape stripes, and the pixel offset from the center of the image, then consulted the lookup tables for the correct values for range, angle, and identification. These values were then used to move the robot in the direction of the current pole. This scheme turned out to be a fast, robust way to identify ten poles in a large arena and then move to them.

4.2 Infrared Beacons for Pole Identification and Location Determination

The IR Beacon system supplied by Denning Robotics is made up of infrared beacons (using LED transmitters) and an infrared sensor. The system configuration devised for the competition robot consisted of 20 IR beacons and one IR sensor. The receiver was placed on top of the robots rotating disk. This provided an ability to turn the beacon with the direction of the robot motion. The output consisted of two angles (vertical and horizontal position) and the beacon code. Using basic geometry the distance was calculated.

Two beacons (facing in opposite directions) per pole were mounted onto a sturdy plexiglass square. This square was fastened to the top of a DC motor which was embedded into a PVC cap for the poles. The motor rotated the beacons at 4 rpm. The slow access time for the beacon forced us to wait at each heading longer than desired. Although the system tested successfully within our lab, when in the convention hall, the lighting produced too much IR noise for the sensor to recognize any beacon and so was not used at the competition.

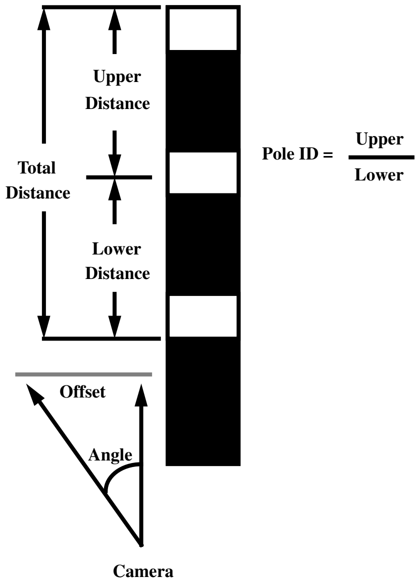


Figure 5: Pole marking for vision

5. Transputer Implementation for Phase 1

Approximately four months before the competition, a proposal was made to port the workstation-based implementation onto a parallel-processing system. The system, called ANIMA (Architecture for Natural Intelligence in Machine Applications) [3], is a flexible, real-time platform that had been recently developed at Georgia Tech for the study of robotic architectures. The goals were to provide both improved performance and on-board computation. The ANIMA structure, although never before used on an actual robot, had been prototyped and used in simulations of both hierarchical and reactive robot systems. Most of the required work was to package it for the Denning robot, add an appropriate interface to the robot, and port all of the schemas as they were developed.

An implementation of the ANIMA architecture had been developed based on the Inmos T800, a member of the transputer family of microprocessors developed for parallel processing. Each transputer provides four high-speed serial links for the required processor interconnection. The ANIMA architecture requires no parallel data busses or backplanes of any kind. Instead, it consists of modular components connected only by high-speed serial links. This allows the processors to be distributed to any convenient locations within the intelligent machine. The transputer architecture for this implementation of ANIMA used five processors and an RS-232 interface spread over six transputer modules, or TRAMs . The TRAMs were mounted on a PC-bus host board within a specially-packaged IBM PC AT compatible system, complete with an electroluminescent display, a floppy disk drive, and a ruggedized hard drive. Although the performance of ANIMA benefits from separate high-speed channels to each physical sensor and effector, the Denning MRV-3 platform provides a single RS-232 port. All communication with the sonar, infrared detectors, bumper switches, and motor controllers had to be multiplexed through this port. The utilization of the five processors is shown in Figure 6.

The motor schemas and much of its perceptual schemas were included in the "*Reasoner*" process (which can easily be split among additional processors as necessary). Some aspects of the perceptual schemas (sensor data processing, mostly) were included within the appropriate logical sensors. Because of the relatively low processing demands placed on the *Integrator* and *Coordinator*, these were combined onto a single processor, and messages to all logical devices were multiplexed on a single channel. These logical devices also ran as parallel tasks

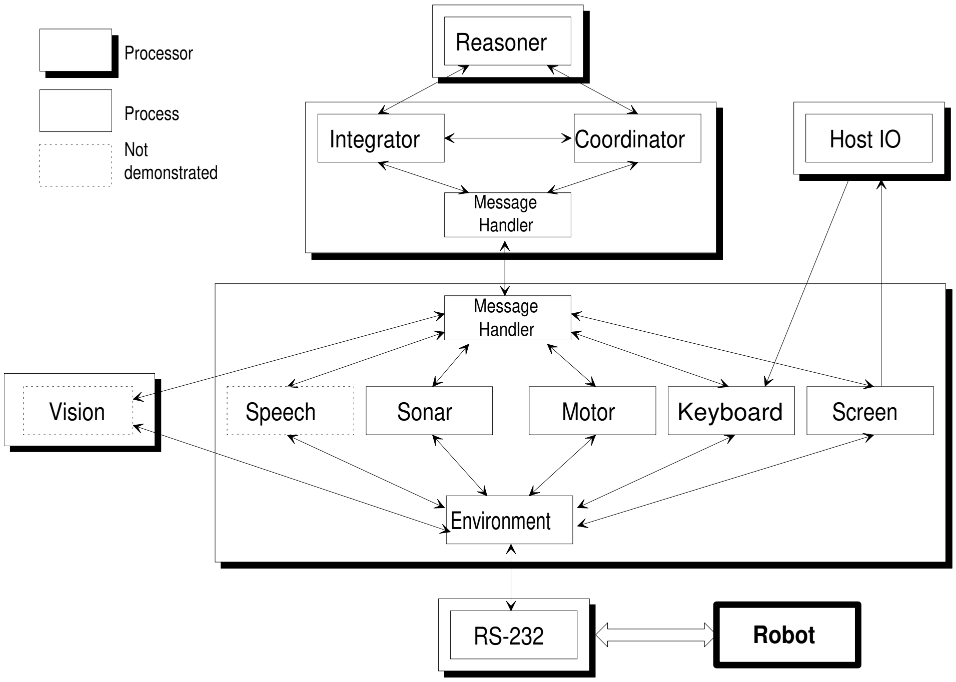


Figure 6: Transputer Implementation

on a single processor, since no especially sophisticated processing was done at this level. This essentially reduced the *Reasoner* to a reactive execution system. No hierarchical control in the traditional sense was used. Provision was made for inclusion of a separate processor (actually, a group of processors) to perform vision, using the remaining link from the logical devices processor. The *Environment* process actually serves a dual role. In normal operation, it passes messages along to the RS-232 handler process. In simulation mode, it intercepts commands to the robot and emulates the behavior of the robot in a grid-based environment, passing back sonar and bumper data when requested. An additional processor (not shown) is used in simulation mode just to provide a graphical display of the simulation status. The impact of this is that simulation capability is built into the real code - no porting is required to keep the simulation current relative to the actual robot software. Of course, the usefulness of any simulation depends on its fidelity. This organization allows the simulator, as a separate parallel process, to be enhanced at any time. We found that the simulation provided good qualitative results with regard to new robot behaviors which were subsequently tested on the actual robot.

As expected, the use of parallel processing in ANIMA significantly extended the capabilities of the robot. For example, the acquisition of all 24 sonar readings through the RS-232 channel takes about 160 msec, and we were able to overlap this with other processing. It was also possible to continuously keep track of the time between robot responses, providing the basis for a dead-man switch if the robot ceased communicating for any reason. The ANIMA-controlled system was able to negotiate obstacle-strewn areas about 50% faster than the Sun-controlled system, mainly because of the decreased latency of sonar data and the managed use of the RS-232 channel.

6. Results

All in all we were quite pleased with the results. The judges for Phase 1 stressed the robot considerably, pushing it into a corner and holding it there for close to 10 minutes (which was different than what was anticipated and specified in the rules). The entire run was 20 minutes

Figure 7: Buzz in Phase 1 of competition

Figure 8: Buzz during Phase 2 of competition

long (Figure 7). Nonetheless we came in fourth that day out of a field of 11.

Phase 2 was much better. The robot was able to find all 10 poles and successfully navigate to 8 of them in the allotted 20 minutes (Figure 8). At the end of the day we were in second place.

Phase 3 however, held on the third day, was plagued by radio/video interference. After successfully finding and moving to the first pole which the judges specified, the video signal was corrupted (unbeknownst to us at first) by another robot operating closer to our receiver than our own robot. Buzz could not visually acquire the remaining two poles, although it did proceed with directed wandering and reacquisition attempts as the plan would indicate. This inability to perceive the last two poles, however, resulted in a poor performance which yielded an overall fifth place finish.

7. Summary

Many lessons were learned during this experience:

- The importance of on-board computation.
  In Phase 2 of the AAAI competition, the offboard Sun computer and radio link performed well, and Buzz ended up in second place, but radio problems in Phase 3 limited us to a fifth-place finish overall. This seemed to indicate that a full port of the vision schemas to the ANIMA hardware would have improved our overall standing, since the machine would have been immune to radio interference. Since visual data could have been processed at much higher frame rates, it would have been possible to perform nearly continuous tracking, also improving Buzz's performance.
- The use of software engineering techniques to support team programming.
  The quality of the software we have been using improved dramatically due to the necessity of sharing code over a multi-person team. This is a lasting contribution to our environment.
- The value of sensor redundancy and backup systems, especially in regards to sensing and communication links.

All in all, the competition and Buzz provided an extremely useful testbed for many of the ideas we have been advocating and was well worth the investment of time and energy.

Acknowledgments

The authors would like to thank Denning Mobile Robotics for providing Buzz for the competition, AAAI for providing financial support for student travel, and Bill Wester and Dave Sonnier for their help with Buzz.

References

[1] Arkin, R.C., "Motor Schema-Based Mobile Robot Navigation", *International Journal of Robotics Research*, Vol. 8, No. 4, August 1989, pp. 92-112.

[2] Balch, T., Arkin, R., "Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation", *submitted to 1993 IEEE Conf on Robotics and Automation.*

[3] Collins, T.R., Arkin, R.C., and Henshaw, A.M., "Integration of Reactive Navigation with a Flexible Parallel Hardware Architecture", submitted to *1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993.

[4] Dean, T.L. and Bonasso, R.P., "The 1992 AAAI Robotics Exhibition and Competition", to appear in *AI Magazine*, Winter 1992, Vol. 13 No. 4, AAAI, Menlo Park, CA.