Design and Implementation of a Teleautonomous Hummer

Darrin C. Bentivegna, Khaled S. Ali, Ronald C. Arkin and Tucker Balch

Mobile Robot Laboratory
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30325-0280 USA

ABSTRACT

Autonomous and semi-autonomous full-sized ground vehicles are becoming increasingly important, particularly in military applications. Here we describe the instrumentation of one such vehicle, a 4-wheel drive Hummer, for autonomous robotic operation. Actuators for steering, brake, and throttle have been implemented on a commercially available Hummer. Control is provided by on-board and remote computation. On-board computation includes a PC-based control computer coupled to feedback sensors for the steering wheel, brake, and forward speed; and a Unix workstation for high-level control. A radio link connects the on-board computers to an operator's remote workstation running the Georgia Tech *MissionLab* system. The paper describes the design and implementation of this integrated hardware/software system that translates a remote human operator's commands into directed motion of the vehicle. Telerobotic control of the hummer has been demonstrated in outdoor experiments.

Keywords: mobile robotics, teleautonomy, feedback control

1. INTRODUCTION

This paper describes the mechanical instrumentation of a commercially available four-wheel-drive vehicle for autonomous operation and reports details of the software control system integration. Autonomous and tele-autonomous control of large vehicles is becoming increasingly important in civilian and military robotic applications.[1,2] Such applications require single vehicle or multiple vehicle control from a remote location. The user might be a soldier operating reconnaissance vehicles in hostile territory[3,4] or a technician operating a bulldozer robot in a dangerously radioactive area.

Figure 1 shows the overall system configuration. Actuators control the steering wheel, brake pedal and throttle. A control computer performs low-level computation for actuator control. The control computer also provides a simple ASCII serial-line interface for high-level control. An on-board Unix workstation processes input from a gyrocompass/inclinometer and Differential Global Positioning System (DGPS) and communicates with a remote control computer via a wireless link.

The following sections describe the equipment installed in the Hummer for actuation, sensing and control, the software for control of the installed equipment, and challenges that arose during the vehicle automating process. Finally, a description of how the system is controlled remotely using the *MissionLab* software system[5] is provided*.

2. VEHICLE ACTUATION AND SENSING

Three actuators control the Hummer, one connected to the brake pedal, one mounted on the throttle, and one to the steering wheel. Each actuator is unique and provided its own set of challenges during development and installation. All three actuators utilize a reversible DC motor that turns at varying speeds depending on the input voltage. Each motor is equipped with a clutch so it can be quickly decoupled from the actuated mechanism.

---

* *MissionLab* is freely available from http://www.cc.gatech.edu/aimosaic/robot-lab/research/MissionLab/.

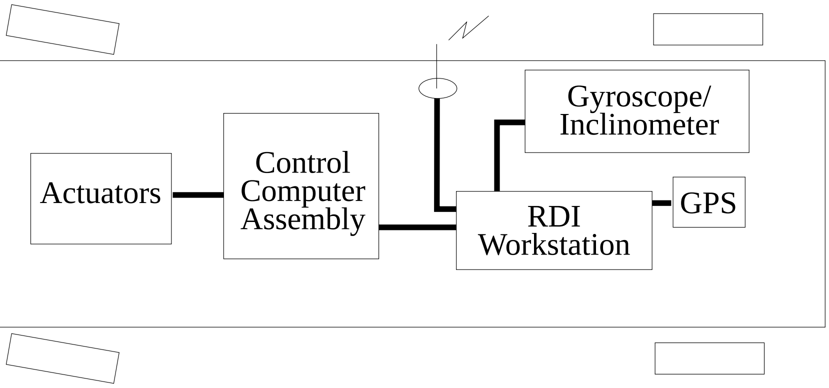

Figure 1. Block diagram of the vehicle system.

2.1. Brake actuator

The brake actuator was purchased as a complete unit from RedZone Robotics and was installed by Georgia Tech personnel. This unit includes a positioning motor, motor position encoder, brake position potentiometer, reduction gears, and a clutch.

The brake actuator is mounted on the floor of the vehicle directly behind the brake pedal (Figure 2). It is connected to the pedal by a chain. Care was taken to ensure that the unit did not interfere with the normal operation of the brake. Two separate indicators of brake position are available: a motor encoder and a potentiometer connected to the actuator chain. When the brake clutch is engaged, the encoder provides a direct indication of the brake pedal position. The potentiometer encodes the position of the brake whether the clutch is engaged or not. Its output is not currently used but could it be used to detect abnormal conditions such as when the operator has stepped on the brake or the brake clutch has failed.

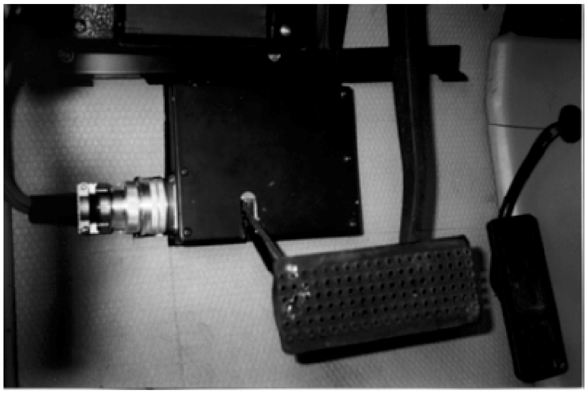

Figure 2. The brake pedal actuator connected to the pedal by a chain.

2.2. Steering actuator

A steering actuator was assembled from off-the-shelf components. A DC motor, reduction gears, position encoder and a clutch are used for the driving mechanism. A bracket was fabricated that allows the driving mechanism to be mounted directly under the steering wheel shaft and as close to it as possible (Figure 3). A sprocket is installed on the factory steering wheel and connected to the driving mechanism with a nylon belt. When the steering actuator clutch is engaged, the motor encoder provides a direct indication of the position of the front wheels.

As a safety measure, the drive belt is designed to slip if the operator attempts to override automatic steering commands. This slippage could become a problem if the vehicle is operated in rough terrain and excessive force is required to turn the wheels. Slippage will result in a mismatch in the actual and measured position of the front wheels. If slippage occurs, the vehicle will oscillate about the desired heading.

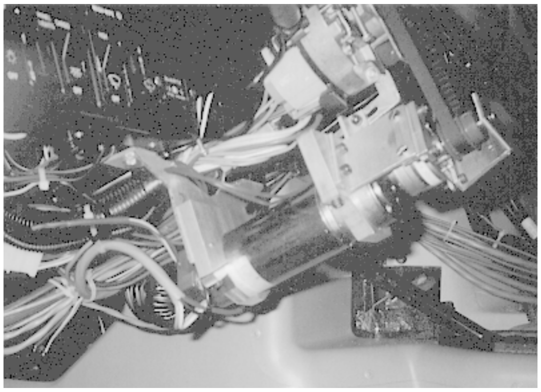

Figure 3. The steering actuator mounted beneath the Hummer's steering wheel column.

2.3. Throttle actuator

The factory installed cruise control servo is used to operate the throttle. Modifications were made to the control unit to allow direct computer access. The actuator unit consists of a positioning servo and a clutch. Use of this servo made physical installation easy but software control difficult. This servo does not have a positioning encoder which led to several implementation difficulties that are covered in Section 3.1.

The throttle actuator, unlike the brake and steering actuators, does not need a constant voltage applied to it in order to maintain the desired throttle position. Once the throttle position has been set and the voltage is removed from the servo motor, the throttle will remain in that position as long as the throttle servo clutch is engaged. When power is removed from the clutch, the throttle returns to the idle position.

2.4. Gyroscope/compass

The vehicle is equipped with a Digital Gyro Stabilized Sensor System from KVH Industries. It is mounted over the rear passenger side wheelwell, as this location was found to provide minimal magnetic interference. It provides stabilized pitch, roll, and azimuth readings for the vehicle. Data from the system is input directly to the on-board RDI Unix workstation (not the low-level control unit).

2.5. Low-level control unit

The low-level controller was designed at Georgia Tech specifically for this system. At the heart of the actuator control is a PC104 80486-based computer. This board has support for a VGA monitor, four serial lines, a parallel line, an ethernet connection, and an LCD monitor. For development purposes, a hard drive is connected to the PC104, but the computer can also be booted with a floppy disk containing the controller software. Figure 4 shows how the PC104 computer is situated relative to the other electronic components.

PC104 compliant controller boards are used to control the brake, steering and throttle actuators. The controller boards handle much of the low level feedback control necessary for the positioning devices. The controller boards are initialized with parameters for proportional gain, integration, integration limit, differentiation, velocity and acceleration. Once the parameters are set, a user can give a desired position to the controller board and it will execute the required trajectory without intervention.

The controller boards work well for brake and steering actuation. These actuators have encoders that provide the position of their respective motors. The encoders enable the controller boards to operate with only low-bandwidth communication from the control computer. The motor and clutch for the throttle actuator are also connected to a controller board, but since there is not an encoder on this actuator, the encoder inputs on the embedded controller are left unconnected.

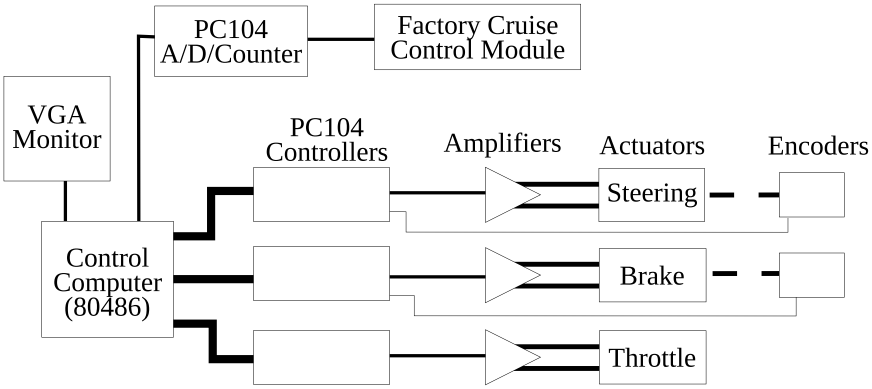

Figure 4. Low-level control unit. Position feedback for steering and brake are provided by motor-mounted shaft encoders, while throttle feedback is through the vehicle speed monitored at the cruise control module.

A PC104 compliant board with A/D converters and counters is installed at the top of the PC104 stack. This board is used to measure the speed of the vehicle. Distance pulses are generated by a magnetic pickup coil located on the transmission output shaft. This is the factory-installed sensor whose output is used in various places throughout the vehicle to compute speed. The raw signal is very noisy and could not be used reliably when connected directly to the PC104 counter. The same signal, however, is used by the factory installed cruise control module. Fortunately, the conditioned signal from the cruise control module produces a clean square wave whose frequency is proportional to the speed. This conditioned signal was used for speed control.

All of the low-level control hardware is installed in the Motor Control Box located between the two rear seats, (Figure 5) permitting a person to easily operate the system from the right back seat. It is cooled using re-routed air from the vehicle's environmental control system.

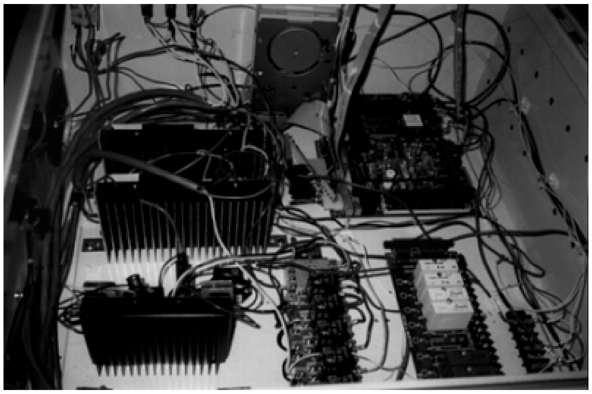

Figure 5. The Motor Control Box: low-level control computer and power supplies.

2.6. Power distribution

The installed control equipment requires electrical power of several different voltages and currents. Figure 6 is a block diagram of the power distribution subsystem. Primary electrical power during autonomous operation is obtained from the vehicle's battery.

The actuator amplifiers drive servos at up to +/-24 volts and require a source voltage of 28VDC. Additionally 5VDC and 12VDC voltages are required for operation of the low-level computer and 110VAC for the standard VGA monitor. In the future, the computer monitor will be replaced by a low power LCD display.

An inverter/charger was installed to provide 110VAC to operate the computer monitor and any other devices that may require AC power. When operated in invert mode, the inverter/charger uses the vehicle's battery power to produce 110VAC. When the inverter/charger is plugged into a 110VAC outlet it supplies 110VAC and 13.6VDC to the vehicle. The 13.6VDC is used to charge the vehicle's battery. This charging capability has come in handy on numerous occasions. The DC voltages are produced using two DC/DC converters. The converters produce reliable outputs for input voltages between approximately 11 and 14VDC.

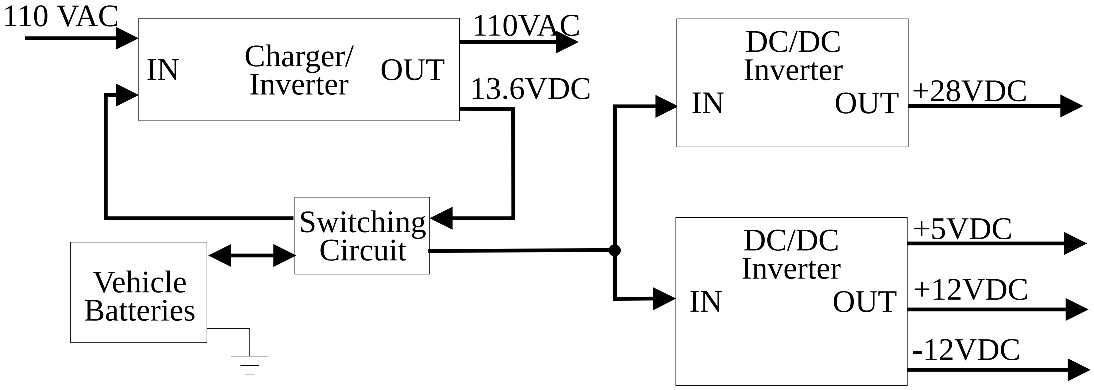

Figure 6. Power distribution block diagram.

3. LOW-LEVEL CONTROL SOFTWARE

The low-level control software runs on the control computer and provides the lowest level interface to the sensors and actuators equipment. This software was developed in C++ to take advantage of data abstraction. This method of software design allows it to be used with different hardware designs without changing the high-level code that uses the application programming interface. For example, if another vehicle is developed with the same hardware functionality but with different actuators and controllers, the class methods would be rewritten for the different controllers and all other code would function without modifications. Figure 7 shows the classes and the relationship between them. Each of these classes is described below.

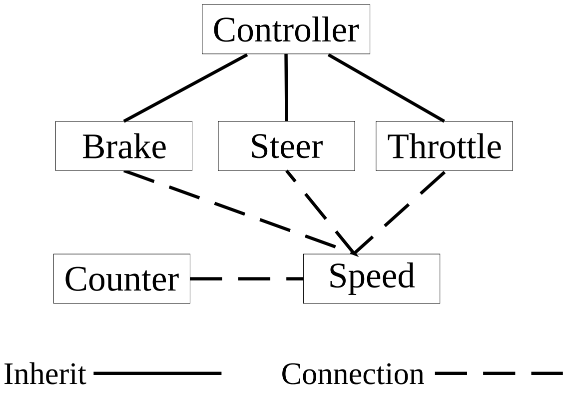

Figure 7. C++ software for low-level vehicle control: class hierarchy.

3.1. C++ classes for vehicle control

The Controller class provides the application programming interface to the controller boards including methods for initialization. Positioning devices connected to the embedded controllers are controlled via objects that inherit from the Controller class.

The Steer and Brake class inherit from the Controller class and add methods as needed for their unique operation. These classes control the position of the steering wheel and brake pedal. To ensure that excessive force is not applied to the steering wheel, the Steer class will not attempt to turn the wheel unless the vehicle's velocity is greater than 1 mile/hour (mph). In order for a Steer object to know the current ground speed, it is connected to a Speed object.

The Throttle class also inherits from the Controller class for control of the position of the throttle. Due to the throttle actuator's lack of an encoder, this class is programed differently than the Steer and Brake classes.

The positioning error abilities of the embedded controller are used to control the movement of the throttle with a single command. Since the throttle controller does not receive feedback from the actuator, there will be an increasing error as the controller tries to move the actuator to a commanded position. A maximum position error is specified in the controller that, when exceeded will cause power to be removed from the servo motor. The length of time power is applied to the actuator is proportional to the position error specified. This method works with the throttle because this actuator does not need to have a voltage applied to it to maintain its position. The use of the throttle to control the speed of the vehicle is covered in the Speed Class section below.

The Counter class provides methods to interface to the installed A/D/counter board. Methods can be added to this class to control and access all items connected to the A/D board. Currently, the primary function of this board is to read the counts from the ground distance sensor. There is another counter available that may, in the future, be used to measure engine speed or other devices that provide a pulse. This board can measure voltages and therefore can provide the capability to add future functionality to the entire configuration such as a temperature sensor in the Motor Controller Box or reading the throttle position via the vehicle's Throttle Position Indicator.

The Speed class provides methods for controlling and reading the speed of the vehicle. The method for controlling the speed is designed to be placed in a loop and called at least four times per second. In order for the Speed object to operate, it must be connected to a Brake, Throttle, and Counter object. The Speed class will read the distance traveled via the Counter object and compute the speed. If the vehicle speed is less than the desired speed, the Speed class will ensure the brake is released and apply the throttle. If the vehicle's speed is greater than the desired speed, the Speed class will decrease the amount of throttle applied. If the vehicle does not begin to slow, the Speed class will slowly apply the brake until the desired speed is obtained. If a stop command or a desired velocity of zero is sent to the Speed class, it will apply the brake to bring the vehicle to a safe and steady stop.

3.2. Operating the installed actuators via the low-level control software

Two programs were written for the control computer: one to test the installed equipment and the other to allow the actuators to be controlled using a simple command language over the serial port. These programs load control parameters from a file. The file contains values for the proportionality gain, integration gain, differentiation gain, and integration limit parameters for each motor.

The testing program provides a graphical user interface which allows a user to move the actuators and change the parameters. This program also displays the brake and steering motor positions and the vehicle speed as read via the Speed object.

The serial interface program allows the actuators to be controlled via the computer's serial port. The program provides a simple set of commands that can be expanded as needed. The current commands are listed in Table 1. The commands provide a high level method for controlling the vehicle. The commands may be sent from another computer on-board or from a remote computer via a wireless transmitter.

4. HIGH LEVEL CONTROL SOFTWARE

A Unix laptop computer (an RDI SPARCstation) inside the vehicle serves two purposes. First, it is the communication point with the remote operator console external to the vehicle using FreeWave wireless radio modems. Second, it takes input from the remote operator console in the form of compass directions, and using the on-board compass, converts these directions to steering, brake, and throttle commands. The purpose of these commands is to orient the vehicle in the actual directed compass heading. The output commands are sent to the PC104 computer which controls the actuators.

Table 1. Serial-line control commands.

Command Format	Description
`home vel acc`	Set the wheels dead ahead with angular `vel` degrees/sec. Accelerate the angular change by `acc` degrees/sec/sec.
`drive vel acc`	Accelerate to vel mph, if vel is less than the current velocity, slow down.
`setsteerangle ang vel acc`	Steer the wheels to the new position `ang` at `vel` degrees/sec. Accelerate and decelerate the angular change by `acc` degrees/sec/sec. A wheel position left of center is indicated by a negative `ang`. Velocity is always positive.
`stopdrive`	Stop movement (release throttle and apply the brake).
`stopsteer`	Stop changes in steering. The controller will seek the present steer angle.

4.1. Operator Console

The operator console resides on another Unix RDI laptop computer external to the vehicle. It forms a part of the *MissionLab* software system.[5,6] When controlling the vehicle, the human operator uses an on-screen "joystick", as shown in Figure 8. This joystick is labeled with N, E, S, and W indicators, representing the compass directions north, east, south, and west. When the operator clicks with the left mouse button in this joystick, direction and speed commands are sent to the robot. The direction sent is a compass direction in degrees, based on the location given by the joystick. The speed value is based on the distance from the center of the joystick to the location clicked. The greater the distance, the greater the speed that is sent to the robot. The operator can order the vehicle to stop by clicking the middle mouse button anywhere inside the joystick. For more detail on the teleoperation software the reader is referred to.[7]

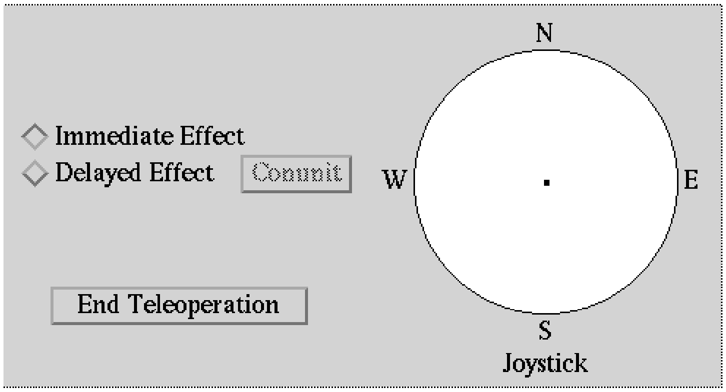

Figure 8. On-screen "joystick" used to control the robot.

5. TEST RUNS

The operation of the robot system was tested by driving it through the lanes of a crowded parking lot and in an empty parking lot, making several turns, circles, and figure-eights, as shown in Figure 9.

For these tests, the operator was located in another parked vehicle with the operator console, as shown in Figure 10. A second person was located in the driver's seat of the robot for safety reasons. Most of the time, the robot remained within view of the operator, although during one of the tests the robot was briefly steered out of sight of the operator, with the on-board personnel relaying via radio the status of the vehicle back to the operator.

Figure 9. The robot is making a U-turn at the command of the operator.

Figure 10. The operator controlling the robot from the operator console.

The human operator controlled the movement of the robot using the on-screen "joystick" on the operator console. The remote operator gave simultaneous direction and speed commands. The compass direction commands are converted into egocentric coordinates and the speed into commands to the robot for control of steering, brake, and throttle to achieve and maintain the specified velocity.

This form of telerobotic control proved to be adequate for controlling the vehicle in the test scenarios. The human operator was able to move the robot where intended, although the robot had to be kept moving slowly when there were obstacles present. With this system, the operator has the responsibility of directing the robot to avoid obstacles in the environment. The telerobotic control of the vehicle will be greatly improved by incorporation of obstacle avoidance. Future work will extend the principles of shared control to combine the operator commands with obstacle-avoidance commands generated by the robot as described in.[7]

6. CONCLUSION

The Georgia Tech instrumented Hummer provides a versatile platform for autonomous and tele-autonomous robotic research. The vehicle is outfitted with hardware and software to control the steering wheel, brake pedal and throttle using an on-board computer or from a remote operator's console. The system has been integrated with *MissionLab* which provides for the design and execution of high-level behavior-based robotic control. Both the software and hardware are configured to provide for future expansion.

The ability to control large vehicles autonomously is important in an increasing number of domains. Georgia Tech's experience in developing a tele-autonomous hummer demonstrates the feasibility of such a vehicle and can serve as a guideline to others interested in actuating their own stock four-wheel drive vehicle.

Acknowledgments

This research has be funded by DARPA/ONR under grant #N00014-94-1-0215. The authors would like to thank Dr. Douglas MacKenzie and Dr. Jonathan Cameron for their involvement in the design of this system.

REFERENCES

1. P. Kenny, C. Bidlack, K. Kluge, J. Lee, M. Huber, E. Durfee, and T. Weymouth, "Implementation of a reactive autonomous navigation system on an outdoor mobile robot," tech. rep., Artificial Intelligence Laboratory, University of Michigan, 1994. http://krusty.eecs.umich.edu/ugv/homepage.html.
2. W. Aviles, et al., "Issues in mobile robotics: The unmanned ground vehicle program teleoperated vehicle (tov)," in *Mobile Robots V*, pp. 587–597, SPIE, (Boston, MA), November 1990.
3. D. J. Cook, P. Gmytrasiewicz, and L. Holder, "Decision-theoretic cooperative sensor planning," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(10), pp. 1013–23, 1996.
4. T. Balch and R. Arkin, "Motor schema-based formation control for multiagent robot teams," in *First International Conference on Multiagent Systems*, pp. 10–16, June 1995. San Francisco.
5. D. MacKenzie, J. Cameron, and K. Ali, "MissionLab: User manual for MissionLab version 0.85," tech. rep., College of Computing, Georgia Institute of Technology, 1996. http://www.cc.gatech.edu/aimosaic/robot-lab/.
6. D. MacKenzie, R. Arkin, and J. Cameron, "Multiagent mission specification and execution," *Autonomous Robots* 4(1), pp. 29–52, 1997.
7. R. Arkin and K. Ali, "Integration of reactive and telerobotic control in multi-agent robotic systems," in *From Animals to Animats: Proc. Third International Conference on the Simulation of Adaptive Behavior*, pp. 473–478, (Brighton, England), August 1994.