Human Control of Cooperative Robots Working on Mars

Geb Thomas*, Mark Fienup**, Paul Gray**, Nick Changnon*, Paul Debbins*, Ryan Dixon**, Tom Ferris*, Luke Hirshy*, Holly Jongedyk**, Aaron Locke*, Dan Perry**, Michael Symonds**, and Rizwan Tahir**

*The University of Iowa **The University of Northern Iowa

In a distributed, behavior-based system, multiple robots work together to accomplish complex tasks using a control logic similar to that used by a colony of termites or ants gathering food and building complex shelters. The flexibility and robustness of this type of system is suited for robotic exploration and resource identification on Mars. The principle question considered in this research is: how can people interact with a distributed, behavior-based system.

This is a new field of robotic research, heralded by NASA leaders as the direction to head in robot exploration of space and as with all initiatives, the problems inherent in making this happen are being revealed. The purpose of this research is to understand how human management can be integrated into a distributed robotic system without undermining the positive control characteristics that make distributed robotics desirable.

Because people have many skills that are difficult to artificially replicate, such as interpreting unusual or unexpected events, drawing on experience, reasoning inductively, and developing novel solutions, human participating typically makes a robot system more reliable, intelligent, and flexible. However, the current research with distributed systems emphasizes autonomous operation, which would provide scientists on Earth with little influence or control over the system on Mars. The purpose of this research is to continue to look for ways to effectively bridge this gap between human operator and robots, thereby creating a communication interface that uses the strengths of both systems.

Our novel system design, based loosely on JPL's CAMPOUT architecture, allows an operator to influence the particular behaviors of each of five the robots within a robot swarm. Three of the robots have sonar devices and two have cameras. Communication cycles are sent through the World Wide Web so that the robots can be controlled from any networked computer. The initial target task was to explore a remote room without relying on absolute position sensors on the robots – one of the most demanding design challenges in a Mars exploration mission.

During the project's first year we developed a preliminary distributed robotic system -- a communication infrastructure between five robots and one human operator -- and ran one test experiment. The tasks necessary to do this included: developing a wireless communication architecture, a communication protocol, creating new hardware for relative position signaling using infrared LEDs, integrating a wireless camera system with the robots, designing and building a graphical user interface for controlling the robots, and designing and programming a set of behaviors to facilitate exploration of an environment. During the second year we intend: to demonstrate a new mathematical algorithm that will provide better relative position estimates and accuracy measures, to advance the architecture to more closely match the CAMPOUT architecture used at JPL, and to demonstrate the effectiveness of the design in order to interest program managers at NASA.

Introduction

Dan Goldin, Administrator of NASA, recently described his vision for space robotics:

> Ultimately, though, we want "herds" of machines to function as a cohesive, productive team to explore large areas of planets. We do not want them too big and clumsy. And we do not want armies of people on the Earth controlling them. Such an approach is very expensive and scientifically not very efficient.
>
> We want our robots to be small, agile, energy efficient, smart and adaptable…
>
> We need to move toward the next era of robotic operations in space, which places the human into the role of a systems manager - not a real time controller as they are today. This is a crucial difference. (Goldin, 2000)

The Mars Reference Mission (Hoffman and Kaplan, 1997) describing the steps to land humans on Mars includes a series of activities such as collecting natural resources and setting up the power station that must be done by robot teams. The Mars Surveyor Mission team recently (12/01) reported evidence of water within 1 meter of the Mars' surface. Mining this water could provide both material for life support and fuel for a return mission from Mars. While mining and extracting this water is many steps removed from where the field of robotics stands today, the work proposed here represents the initial move to begin a systematic analysis of how a distributed robotic system could one day accomplish these tasks.

Swarm robotics is one avenue robotic engineers are exploring as a means to meet the need for developing robust, distributed intelligent systems. It is an approach inspired by natural phenomena. (Brooks, 1989; Arkin et al. 1998). The hive building behavior of termites is an example of coordinated group behavior (Bonabeau, et al. 1987). Each termite possesses very limited cognitive capabilities and lives by simple instinctual reactions to its immediate local environment. There is no leader that coordinates the behavior of a termite colony. Yet, the combined efforts of these simple creatures create hives thousands of times the size of an individual termite. While the individual termite is very simple, the *emergent behaviors*, or the coordinated actions of the collective, satisfy the needs of the group.

Decentralized systems offer several advantages over centralized systems. First, decentralized systems do not have a single point of failure; the loss of a single agent will not disable the system. Second, decentralized systems tend to be easier to design and build, because the designer need only create simple, low-level behaviors, instead of a single, complex control system for all possible situations. Finally, decentralized systems are inherently parallel, which allows for scalable systems and faster task completion. These advantages make decentralized systems an attractive alternative for space exploration.

Both Dan Goldin's comments and the Mars reference mission suggest that scientists and engineers on Earth will supervise the robot teams in order to help the robots overcome unexpected obstacles, recover from unexpected system error, and make decisions that are difficult or impossible to automate. The question of how to accomplish this effective human-robot interaction, however, is a largely undeveloped research area, and one the proposed research argues to be the initiation of a new field in the study of robotics and human-machine systems. Of the over 1000 books, journals and technical reports at the University of Iowa with the keyword robot, less than 5% also mention people, person, man or human. Most of this small subset of human-robot interaction research emphasizes telemanipulation -- the direct control of an individual robot joint as opposed to task-level control -- or examines societal relationships and philosophical issues regarding people and robots. The human control of a swarm of robots is largely without precedent. The proposed work assumes the task of defining how to effectively integrate a human operator with a distributed robot system, using an approach that is relevant to NASA's current plans.

Related Work

Sheridan and Rasmussen on Human-Robot Interaction

Thomas Sheridan, a pioneer in human-robot interaction, describes the interaction of a person with a telerobotic system in terms of four tasks: planning, teaching, monitoring, and intervening. Sheridan's model (Sheridan, 1992) illustrates how the operator must allocate his or her attention among the different sources of incoming information, estimate the state of the robot, and eventually select a control action from among a list of potential actions for the remote system. This is a powerful subjective model. However the model is operator-centric. It focuses on the tasks and strategies exercised by the operator before sending the commands to the robot, rather than illustrating how human and robot can be designed to achieve optimal cooperative harmony.

Rasmussen's (1986) abstraction-hierarchy approaches the problem of human-robot interaction from a different perspective. With this approach the control goals and system implementation are laid out in a hierarchy that moves from the physical instantiation of each component to the general purpose of each subsystem to the goal of the system as a whole. This hierarchy provides the understanding and framework from which the communication between the operator and the system may be designed. The abstraction hierarchy is centered on developing the appropriate mechanism for communication between the people and the machines, provided that the machine is already designed. Consequently, the technique becomes intractable when the robot design is unconstrained.

Robot Control Work at NASA Ames

This work is a natural outgrowth of a series of experiments carried out by NASA Ames Research Center with mobile robots in challenging remote environments. Trials have been performed at the Kilauea volcano (Stoker and Hine, 1996); Mt. Spur, Alaska (Bares and Wettergreen, 1997); under the Arctic glacial ice (Hine et al., 1994); Dinosaur Tracks, Arizona (Christian et al., 1997); the Atacama Desert in Chile (Cabrol et al., 1998, Wettergreen et al. 1999); and the Silver Lake region of the Mojave Desert (Stoker et al., in press, Thomas et al., in press). In these tests, a field robot, directed by a team of geologists and engineers, conducted specific scientific observations of the remote terrain. Typically, the centerpiece of the operator interface has been the generation and presentation of photorealistic virtual reality environments representing the robot's immediate surroundings (Christian et al., 1997; Nguyen 2000). With each iteration, the team gained experience, developed new systems, procedures, tools, and technologies, fueling the growth of the field of robotic geology. This proposal represents a departure from the previous work because it emphasizes multiple robots and it does not pursue the development of a photorealistic virtual environment. Instead, it emphasizes the development of a scientific foundation for human interaction with multiple robots.

Previous work in the GROK Lab

Jerry Steele, a recent alumnus of the GROK Lab, developed a novel system to control a robot swarm. His technique included a set of behaviors to ensure a continuous communication relay to distant members of the swarm. The operator indicated a desired direction for exploration. The robots in the swarm would more or less move in this direction depending on the strength of the operator's command and their other behaviors, such as obstacle avoidance and the need to stay in communication with the other robots. This approach allowed the operator to influence system behaviors without overriding the individual behaviors of each member of the swarm. The approach was successfully simulated, but has not yet been demonstrated with physical robots.

Method

The main components of the physical architecture are pictured in Figure 1. The system includes 5 robots. All the robots have a visual sensor, infrared detectors and transmitters, a speaker and microphone, and three bump sensors. In addition, three of the robots have sonar transmitters and two have cameras. The robots in Figure 1 both have sonar sensors (the round disk at the top of the mast). Each robot has a microprocessor that can communicate through a serial connection to an IPAQ handheld computer. The IPAQ has a wireless modem that allows the IPAQs to communicate with each other with standard Internet protocols. Information collected from the robot and commands sent to the robot are exchanged with one or more operator interface terminals. An operator interface running on the network allows sonar and image information collected from the robots to be graphically displayed.

Hardware

The robots are Rugwarrior Pro robots. The robots use a Motorola MC68HC11A microcontroller and 23K of random access memory. The microcontroller has access to the states of all the sensors and controls two 6-volt gear motors, which drive two 2 1/2 inch wheels. If the clear plastic body collides with an object in the robot's path, the body shifts and depresses one of the three bump sensors. Each robot was constructed from individual components (resistors and capacitors) by GROK Lab students.

The sonar sensors also come unassembled in a kit. The resolution of the sonar device was better than 1/32" absolute and is reported to 5 decimal places of precision, although we have not been able to confirm this resolution. The effective range of the sonar device is 1-1/2 to 17 feet, but the precision degrades dramatically for target more distant than 8'. The sonar is mounted on a servomotor, which was tested with the following experiment:

Servo Motor Experiment

The Rugwarrior Pro robot mounted with the Rugbat sonar kit board and servo was placed in a position facing a long wall with a perpendicular distance of 9 feet (2.75 meters) between the front of the servo and the wall. A laser pointer affixed to the top of the servo shone at a distinct point on the facing wall. Using preprogrammed software commands to position the servo, different angle positions were commanded with integer arguments such as set servo (0); or set servo (25). The integer arguments range from –100 to +100 and each increment represents approximately a one-degree angle change. For each angle position commanded, the exact spot of the laser on the facing wall was marked with tape and pencil. Different combinations of successive servo position settings were observed, and the accuracy with respect to previous calls to the same setting could be estimated by comparing the laser point to the pencil-marked point on the wall that represented where it was supposed to be.

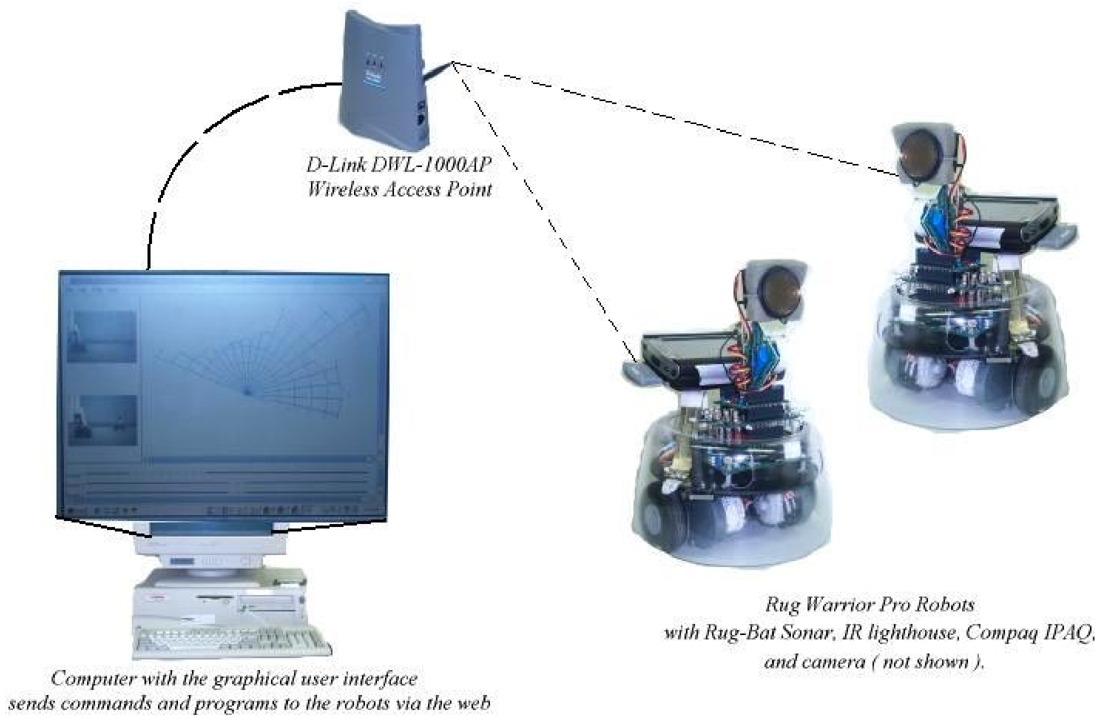


Figure 1: Main system components.

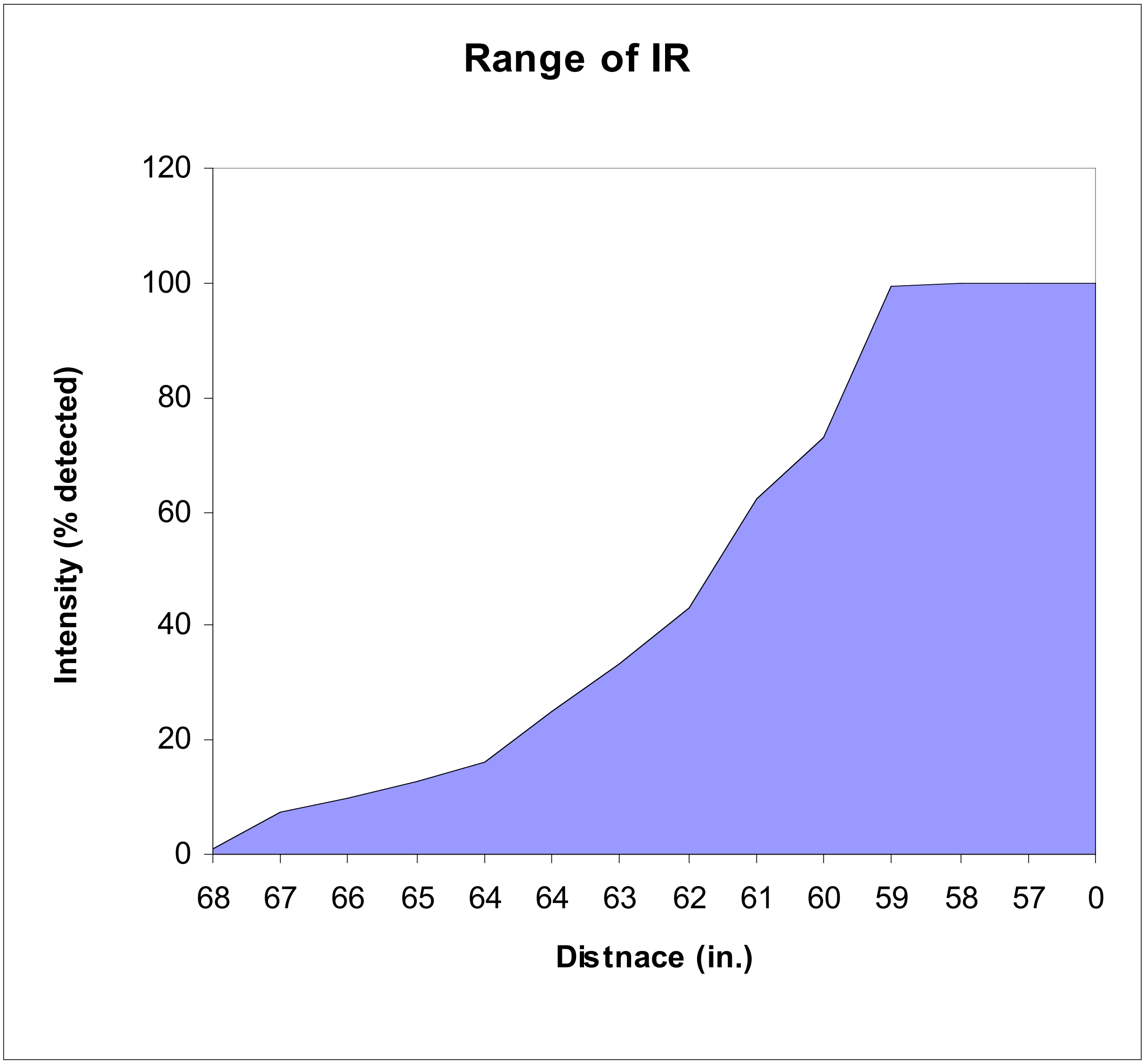


Figure 2: Lighthouse detectability as a function of position relative to the lighthouse.

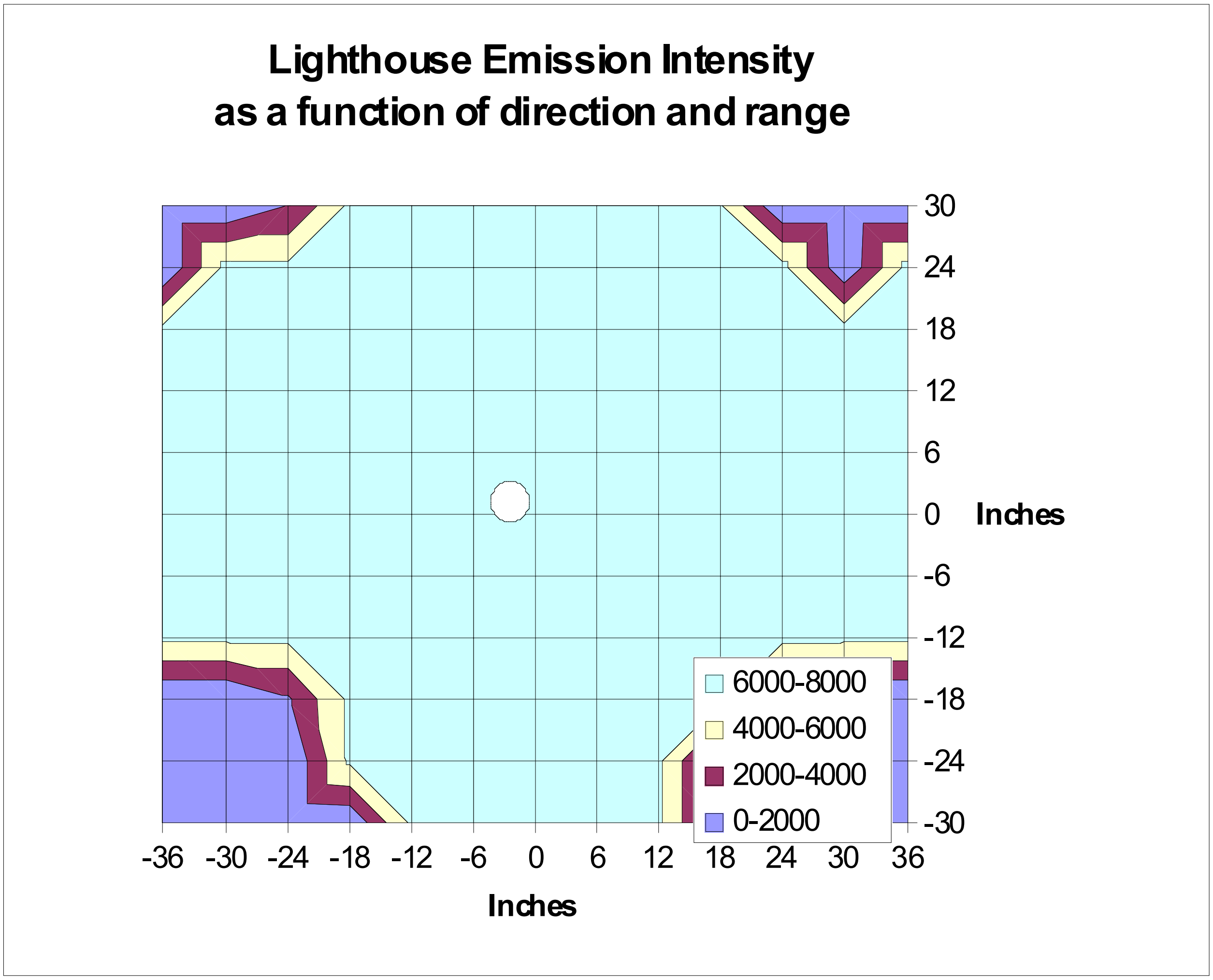


Figure 3: Detected lighthouse intensity as a function of distance.

Observations:

- Compared to earlier positions of the same integer argument when other positions are called in between, the servos could be off by as much as one degree or more in either direction. For example, calling set servo (6), set servo (25), then set servo (6), resulted in more than a degree of inaccuracy.
- Turning on and off of the servo and reinitializing it, then calling the same positions, has a repeatability of better than one degree
- Over a period of 5 minutes, the servo will drift up to a degree to the left (counter-clockwise). .
- Some commanded increments of one degree show no change at all in position of the laser pointer, suggesting that the device has limited positional resolution (approximately 1 degree).

The Imaging System

Two of the robots were equipped with cameras. Ultimately we hope to integrate frame grabbers onboard the robot, but the camera signals are currently received through a separate 4 GHz communication path. The 60-degree FOV cameras are wireless video cameras manufactured by the X10 Company. The receiver is connected to a PC with a framegrabber. The framegrabber is manually engaged to capture the image, which can then be forwarded to the receiving machine. In the first image, the word distance is misspelled; don't know how to get into that area. The power supply for the cameras has been modified so that the robot can turn on or off its camera. This allows us to have several cameras on the robots although only one can communicate with the wireless receiver at a time. A separate 12V is mounted on the robot and the power between the source and the camera is controlled by one of the pins on the robot's I/O board.

The Infrared System

Several of the algorithms require one robot to be able to identify another robot, which is not possible through software alone. To address this concern we developed an infrared lighthouse system. Eight infrared light emitting diodes LEDs are mounted near the top of the robot on a specially manufactured circuit board. This figure demonstrates that robots can reliably detect one another at a distance of up to sixty-one inches, but reliability degrades to a distance of sixty-eight, beyond that the robots have to differently detect one another. The pattern is not radially symmetric. The robot can turn on and off these LEDs. Another robot can use its directionally sensitive infrared detector to turn until it sees the transmitting detector. Figure 2 shows the strength of the signal received by the infrared detector placed at various distances from the lighthouse. Figure 3 presents the range of detectable signals around a lighthouse placed at (0,0). This figure demonstrates that the 8 LEDs create an uneven illumination around the robot. Consequently, the reliable detection distance is only 25 inches.

Software Architecture

The robots are programmed using the Interactive C-programming language. Interactive C runs on a host machine and communicates with the robot over a 9600-baud serial connection. Programs can work in either or both of two modes: direct interaction or load-and-run. With direct interaction, the robot can immediately respond to any command typed by the operator. For example, the operator can type "bumper sensors ()" and the robot will immediately respond with an integer reflecting the state of the three bumper sensors. The load-and-run modality is similar to more traditional programming style, wherein the operator writes a program, then downloads it to the robot. After the program is loaded to the robot's memory, it starts to operate and the programmer can disconnect the serial cable between the robot and the computer. Generally the programs loop through the same set of actions again and again. For example, one program might check if the front bumper has been activated, then drive the robot in reverse. Such programs are called behaviors, because they cause the robot to respond to stimulation.

Currently we have created seven behaviors, listed in Table 1.

Table 1
Available Behaviors

Behavior	Description
spin ()	spins a specified number of encoder clicks, then returns.
drive ()	drives a specified number of encoder clicks, then returns.
face beacon ()	spins the robot 360 degrees, while sampling the infrared signal, then returns to the position with the strongest infrared signal.
check face ()	causes the robot to pan left, then right, then continue moving towards the side with the stronger signal until it can lock on a signal.
crowd beacon ()	finds and approaches an infrared beacon.
sweep ()	performs a 180-degree sonar sweep, returns a struct of a distance reading at every degree.
drive past ()	drives past an infrared beacon.

In order to control which robots are currently running on each robot and download new behaviors to the robot, the robots are each connected to the Internet through the IPAQ. In order for this to work, the IPAQs must communicate with the robots through a serial port and with the Internet through a wireless network card. We adopted a client-server model and developed a server program to run on each IPAQ. A parallel client program also developed in C++ could be run on any network-enabled PC. The client program makes a connection with the server and can: 1) download a new behavior to the IPAQ, 2) command the IPAQ to download a behavior to the robot, 3) execute or remove programs from the robot, 4) provide the sonar array returned by a sweep command performed on the robot.

Each message consists of two integers followed by a variable length message payload. The first integer represents the type of message; the second represents the length of the payload. The integer code for each message and its type are listed below.

Messages Sent From the IPAQ Server to the Robot Manager Client

1. Function_Execute_Success
2. Function_Execute_Failure
3. Save_Behavior_Information_Success
4. Save_Behavior_Information_Failure
5. Load_Behavior_Success
6. Load_Behavior_Failure
7. Available_Behaviors
8. Loaded_Behaviors
9. Robot_Name
10. Delete_Behavior_Success
11. Delete_Behavior_Failure

Messages Sent From the Robot Manager Client to the IPAQ Server

1. Execute_Function
2. Save_Behavior_Information
3. Load_Behavior
4. Get_Available_Behaviors
5. Get_Loaded_Behaviors
6. Get_Robot_Name
7. Delete_Behavior

In order to make the system work, the behaviors must be pre-compiled before they are downloaded to the robot. Unfortunately, interactive-C does not have a provision for this need, because it is designed with a just-in-time compiling scheme. Consequently a set of tools were devised to run on the IPAQ and effectively capture compiled behaviors by mimicking the responses a robot normally provides when downloading a behavior. This set of tools allows the compiled files to be collected and stored and moved normally.

The Operator Interface

The final component of the project is a graphical user interface. One interface with full functionality was developed as part of the Client/Server system. This interface provides all the functionality provided by the server. The second interface is designed to assist the operator in understanding the relationships among the robots without resorting to the position estimation system. The interface provides three windows. The window on the left displays one or more images, although currently this functionality is restricted to two pre-selected images. The window on the right displays one or more overlapping sonar maps. The maps are designed to indicate the distance from the robot origin of each of the sonar reflections. The images and sonar grids may be used by the operator to form a cohesive understanding of the remote environment.

Access to the data collected by the robots is rendered by corresponding horizontal lines in the bottom window. Each line represents a robot. Time increases to the right. If a robot collects sonar data or an image, an icon appears on the line. If the robot moves, the solid line is replaced with a dashed line. Eventually the operator will be able to click on icons or drag a selection line around the icons to select which information is displayed in the top two windows. By noting the position of the solid lines, the operator will be able to determine when an image might present the robot's position at the time a sonar map was created. By comparing the different data sources, the operators may eventually be able to construct a map of the entire room.

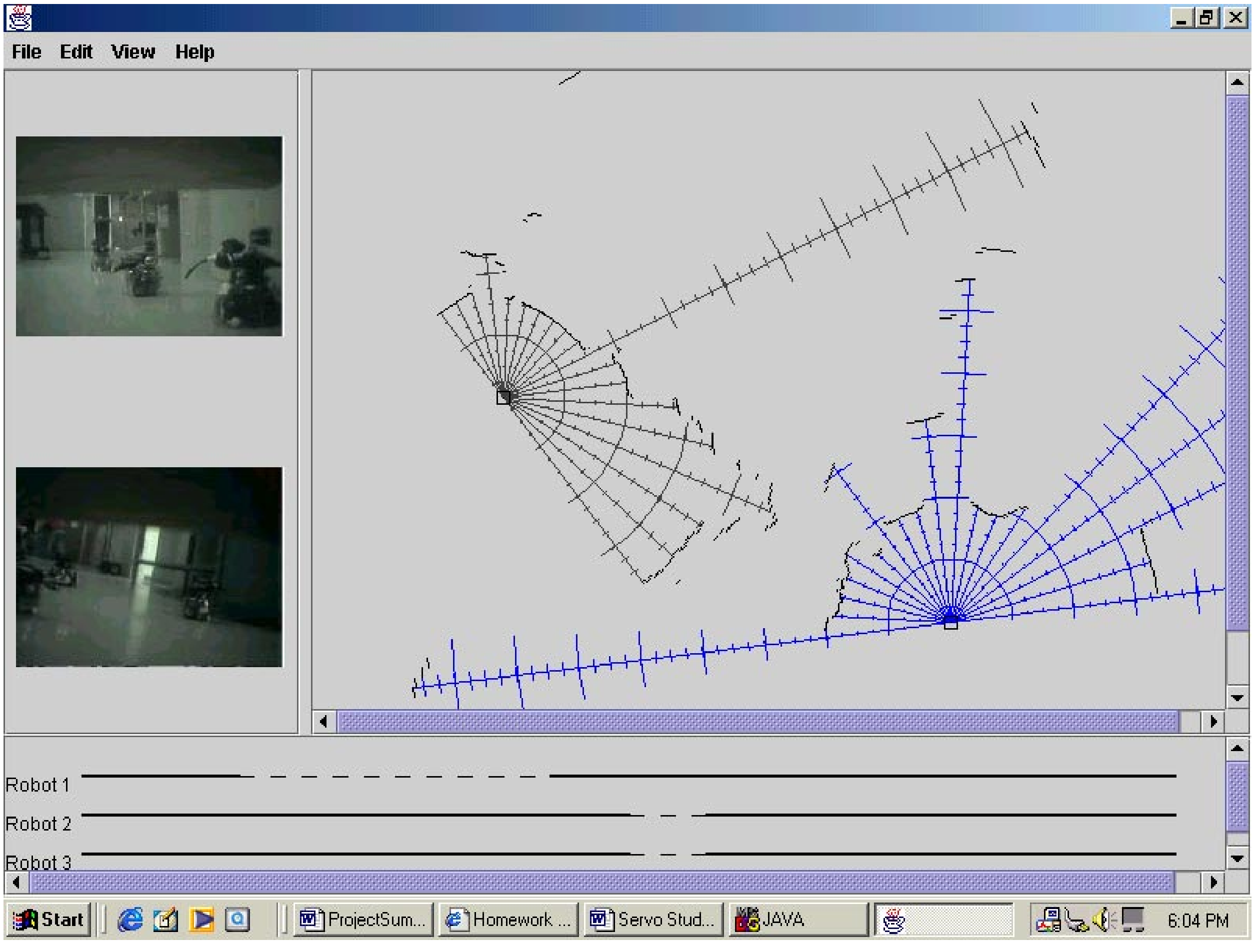


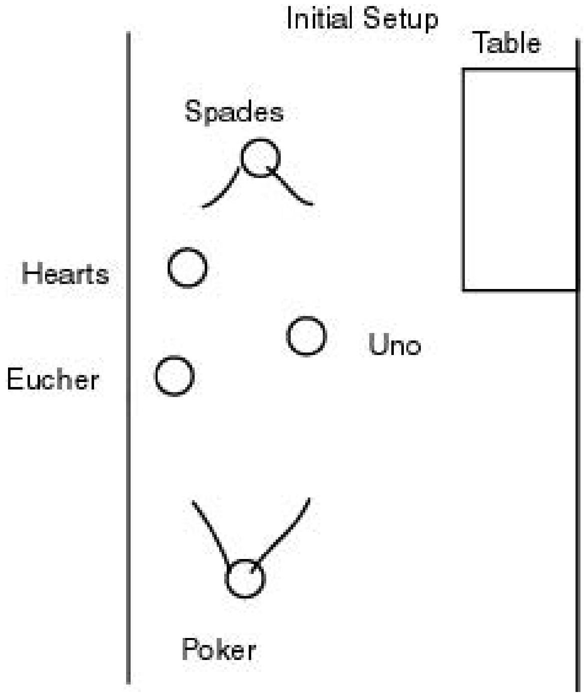
Figure 4: Screenshot from the graphical user interface.

Results

On December 1, 2001, the system was set up in the Center for Computer Aided Design building at the University of Iowa. Five robots were placed in a hallway outside the control room. Not all of the hardware was operational on all of the robots, but at least one example of each technology was working. Images were taken from the two robots with cameras, and sonar data was collected from each of the other three robots twice. First an initial data set was collected, then the sonar robots moved and collected a second set of data.

Figure 5 presents the initial and final position of the robots. Figure 6-9 present the 4 images collected.

Setup for the Robot Test on December 1, 2001



Spades and Poker have cameras, Hearts, Eucher and Uno have sonar devices.
The orientation of the sonar is not clear.

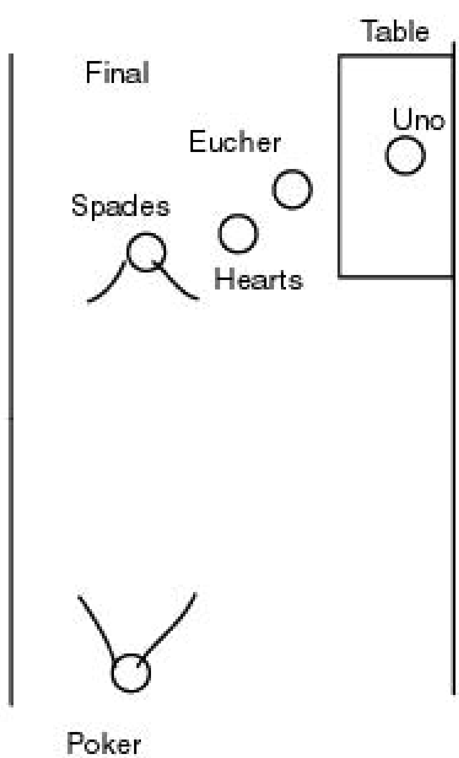


Figure 5: Test setup.

Figure 6: Image from Spades before sonar move.

Figure 7: Image from Spades after move.

Figure 8: Image from Poker before sonar move.

Figure 9: Image from Poker after sonar move.

Conclusions

The test successfully demonstrated the hardware and our ability to collect the data through a wireless network. The utility of the graphical user interface to allow the operator to understand the position of the robots and to map the room without an absolute or relative reference frame aside from the sonar data was not confirmed, however. In the next year we will begin the development of hardware that will us to simulate the use of pseudo-lights. Pseudo lights are placed within an environment and provide a reference position for the robots. We will simulate this capability with sonar beacons. To complete this task we will need to build a lighthouse for each beacon and fix the lighthouse to the top of an object easily detectable with sonar. The lighthouse will be turned on and off over the network with a connection to the parallel port of a PC.

The user interface is currently undergoing a revision now that a relevant data set is available; we hope to refine this to become a novel and productive system. We are also considering various modifications that will provide relative position estimates among the robots.

The operator interface was studied as part of a graduate course in human-computer interaction. The graduate students identified a series of large and small improvements to the user interface including such issues as:

The design of appropriate error messages
Techniques by which control signals may be sent to the robots
The arrangement of buttons on the display
The techniques for direct interaction
The need for hot keys and mnemonics to speed task performance.

All of these issues will be addressed in the next revision of the software. Also, the availability of position information with each data set will allow the display to show data in its correct position as it arrives to the interface.

Bibliography

Arkin, R.A., Khaled Weitzenfeld, Alfredo Cervantes-Perez, Francisco, *Behavior Models of the Praying Mantis as a Basis for Robotic Behavior*, . 1998, Georgia Institute of Technology: Atlanta, Georgia. ftp://ftp.cc.gatech.edu/pub/people/arkin/web-papers/mantis.ps.Z.

Bares, J.E., and D.S. Wettergreen, Lessons from the development and deployment of Dante II, *Proc. 1997 Field and Service Robotics Conference*, Canberra, Australia, 1998.

Bonabeau, E., *et al.*, *Self-organizaiton in Social Insects.* Trends in Ecology & Evolution, 1987. **12**(5): p. 188-193.

Brooks, R.F., Anita, *Fast, Cheap and out of Control*, . 1989, MIT: Cambridge, Mass.

Cabrol, N. A. and E. A. Bettis III , B. Glenister**,** G. Chong, C. Herrera,, A. Jensen & M. Pereira, C. R. Stoker, E. A. Grin, V. C. Gulick, R. Landheim, P. Lee, T. L. Roush, A. P. Zent**,** J. M. Dohm**,** R. Keaten**,** G. C. Dunfield**,** G. Thomas, J. Golden, K. Saville, G. Ludvigson, and B. Witzke**,** D. Wettergreen**,** D. Christian**,** M. H. Sims, K. Schwher, M. G. Bualat, H. J. Thomas, and E. Zbinden 1999**.** Nomad Rover Field Experiment, Atacama Desert, (Chile) II. Identification of Paleolife Evidence Using A Robotic Vehicle: Lessons and Recommendations for a Mars Sample Return Mission. *J. Geophys. Res.*, in press.

Christian, D., D. Wettergreen, M. Bualat, K. Schwehr, D. Tucker, and E. Zbinden, Field experiments with the Ames Marsokhod rover. *Proc. 1997 Field and Service Robotics Conf.*, Canberra, Australia, 1997.

Goldin, D. (2000). Remarks to the Cal Tech Alumni Association, May 20,2000. ftp://ftp.hq.nasa.gov/pub/pao/Goldin/2000/CalTechFinaldoc.pdf

Green, D.M. and Swets, J.A. (1966) *Signal Detection Theory and Psychophysics*, John Wiley and Sons, Inc. New York, NY.

Hine, B.P., Stoker, C., Sims, M., Rasmussen, D., Hontalas, P., Fong, T., Steele, J., Barch, D., Andersen, D., Miles, E., and Nygren, E., The appliction of telepresence and virtual reality to subsea exploration. The 2nd Workshop on Mobile Robots for Subsea Environments, in Proc. ROV '94, Monterey, CA, 1994.

Huntsberger, T. (2001), "Biologically Inspired Autonomous Rover Control," *Autonomous Robots*, 11: 341-346.

Nguyen, L., Bualat, M., Edwards, L., Flueckiger, L., Neveu, C., Schwehr, K., Wagner, M. D. and Zbinden, E. (2000), "Virtual Reality Interfaces for Visualization and Control of Remote Vehicles." *IEEE International Conference on Robotics and Automation*.

Pirjanian, P., Huntsberger, T.L., Trebi-Ollennu, A., Aghazarian, H., Das, H., Joshi, S. and Schenker, P.S. (2000), "CAMPOUT: A control architecture for multi-robot planetary outposts," to appear in Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III, Vol. 4196, Boston, MA, Nov. 2000.

Schenker, P.S., Huntsberger, T.L., Pirjanian, P., Trebi-Ollennu, A., Das, H., Joshi, S., Aghazarian, H., Ganino, A.J., Kennedy, B.A., and Garrett, M.S. (2000), "Robot work crews for planetary outposts: Close cooperation and coordination of multiple robots," to appear in Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III, Vol. 4196, Boston, MA, Nov. 2000.

Rasmussen, J. 1986. *Information Processing and Human-Machine Interaction*, Elsevier North Holland.

Sheridan, T.B., *Telerobotics, Automation, and Human Supervisory Control*, MIT Press, Cambridge, MA, 1992.

Stephen J. Hoffman, S.J and Kaplan, D.L, eds., 1997, *Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team*, Johnson Space Center, http://www-sn.jsc.nasa.gov/marsref/contents.html

Stoker, C and B. Hine, Telepresence control of mobile robots: Kilauea Marsokhod experiment, AIAA, Reno, NV, 1996.

Stoker, C., Cabrol, N., Roush, T., Hovde, G., Thomas, H., Schreiner, J., Thomas, G., Sims, M., et al. Submitted, Journal of Geophysical Research-Planets as part of the Special Issue on the 1999 Marsokhod Rover Silver Lake Field Experiment), "The 1999 Silver Lake Marsokhod Field Test: Mission Overview, Data Sets, and Summary of Results." In press.

Huntsberger, T.L. Pirjanain, P. and Schenker, P.S. (2001), "Robotic outposts as precursors to a manned Mars habitat," to appear in Proc.Space Technology and Applications International Forum (STAIF-2001), Albuquerque, NM.

Thomas, G., Reagan, M., Bettis, A., Cabrol., N., Rathe, A., Analysis of Science Team Activities During the 1999 Marsokhod Rover Silver Lake Experiment: Implications for Automated Planetary Surface Exploration. J. Geophysical Research, in press. Available on-line at http://www.agu.org/pubs/pip/.

Wettergreen, D., D. Bapna, M. Maimone, and G. Thomas, G., Developing Nomad for robotic exploration of the Atacama Desert, *Robotics and Autonomous Systems*, 26: 127-148, 1999.