Intelligent behaviors for a convoy of indoor mobile robots operating in unknown environments

Nathan M. Farrington,[*] Hoa G. Nguyen, Narek Pezeshkian

SPAWAR Systems Center, San Diego, USA

ABSTRACT

Intelligent behaviors allow a convoy of small indoor robots to perform high-level mission tasking. These behaviors include various implementations of map building, localization, obstacle avoidance, object recognition, and navigation. Several behaviors have been developed by SSC San Diego, with integration of other behaviors developed by open-source projects and a technology transfer effort funded by DARPA.

The test system, developed by SSC San Diego, consists of ROBART III (a prototype security robot), serving as the master platform, and a convoy of four ActivMedia® Pioneer 2-DX robots. Each robot, including ROBART III, is equipped with a SICK® LMS 200 laser rangefinder. Using integrated wireless network repeaters, the Pioneer 2-DX robots maintain an ad hoc communication link between the operator and ROBART III. The Pioneer 2-DX robots can also act as rear guards to detect intruders in areas that ROBART III has previously explored. These intelligent behaviors allow a single operator to command the entire convoy of robots during a mission in an unknown environment.

Keywords: indoor mobile robots, intelligent behaviors, unknown environments

1. INTRODUCTION

The goal of the Autonomous Mobile Communication Relay[1] (AMCR) project is to develop a team of intelligent, indoor mobile robots with the purpose of maintaining a wireless network link among themselves and a human operator. The human operator commands the robots through a computer called the operator control unit (OCU). We have developed and integrated several intelligent behaviors for this project in order to provide higher degrees of autonomous operation. We have successfully demonstrated several of these behaviors, and several more in simulation.

This work is important for two reasons. First, the military is moving towards one human operator for many robots (1:M). Currently, the ratio is either one operator to one robot (1:1) for smaller systems, or many operators to one robot (M:1) for larger and more complicated systems such as the Predator unmanned aerial vehicle (UAV). This change will not be possible without each robot having a minimum level of sophistication to efficiently and safely carry out its orders.

Second, mobile robots are plagued with communication problems. Tethered robots are physically hampered when going around corners or running over their own tether, which causes entanglement, stretching, or breaking. Wireless communications between a robot and an operator suffer from multipath interference, signal loss, and non-Line of Sight (NLOS) as a robot penetrates deeper into an unknown environment. A solution to the high-bandwidth wireless communications problem is to maintain near LOS between the lead robot and the operator outside by the use of relays. Each relay will maintain near LOS with neighboring relays, forming a communications chain from the lead robot to the operator.

The layout of this paper is as follows. Section 2 outlines the approach of the AMCR project. Section 3 introduces our laboratory robots and the software tools used during development. Section 4 describes the major behaviors that we have developed or adapted into the overall architecture. Behaviors include 1) communications capability that allows robots to exchange information about the world, 2) robot recognition capability, 3) simultaneous mapping and localization, 4) localization in a convoy without an existing map, 5) localization after map building, and 6) navigatation within the map. We also introduce our future goal of a mixed reality interface, which will provide a very powerful and

[*] Further author information: E-mail: nfarring@spawar.navy.mil, Telephone: 1 619 553 9196

intuitive interface to the operator while becoming an important component in SPAWAR's groundbreaking Composable FORCEnet architecture.

2. APPROACH

We are using a highly capable robot having sophisticated sensors and computing power as the lead robot, with smaller, less capable robots as mobile relays. The communications link between the lead robot and the operator is transparent so that the operator can focus on commanding the lead robot, while still being able to supervise the relay robots. The scenario we are addressing involves the mobile relay nodes forming a convoy following the lead robot at the start of a mission. The rear-most mobile relay stops and becomes a stationary relay when it detects that the signal strength of the link between it and the node behind has dropped to a preset level.[1] However, when a communication shortcut becomes available, and a relay node detects that it is no longer needed, it will rejoin the convoy in order to be reused. This paper addresses the intelligent behaviors that are needed to carry out this last step.

3. HARDWARE & SOFTWARE

The test system consists of one lead robot and four relay robots. ROBART III is an advanced prototype security robot and serves as the lead platform.[2] ROBART III is equipped with a rich sensor suite, multiple computers, and a non-lethal weapon. The sensor suite includes several proximity sensors such as a SICK LMS 200 laser rangefinder, Polaroid® SONAR modules, Sharp® infrared ranging modules, MINI-BEAM® photoelectric proximity sensors, and bumpers. Also included are optical encoders on each of the seven motors, a compass, and a gyroscope.

The other four robots used in the test system are ActivMedia Pioneer 2-DX robots. They are configured identically and each one is equipped with a SICK LMS 200 laser rangefinder, 16 Polaroid SONAR modules, and an optical encoder on each of the two wheel motors. We have extended the basic Pioneer platform by installing a Pentium-based single board computer (SBC) in each robot to implement the various behaviors. Fig. 1 shows a typical convoy formation.

In addition, we have installed a Compact Ad-hoc Networking Radio (CANR) in each of the five robots. The CANR was developed jointly by SPAWAR Systems Center and BBN Technologies. The CANRs are able to form an ad-hoc network using a proactive link state protocol where links are monitored and transmission paths routed automatically before a link is broken due to situations resulting in non-LOS between radios. The CANR, shown in Fig. 2, is a complete, stand-alone system that interfaces with the SBC via an Ethernet port.[3]

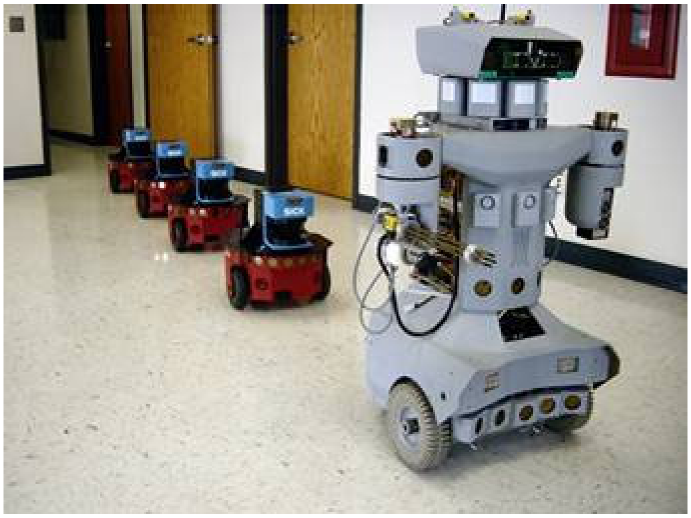

Figure 1. ROBART III is the lead robot in the convoy with four ActivMedia Pioneer 2-DX robots trailing behind. Since this photo was taken, ROBART III has been upgraded to use a SICK LMS 200 laser rangefinder, seen mounted on the Pioneers. Each robot is equipped with a CANR for wireless ad hoc network relaying.

Figure 2. A CANR is about the size of a pack of playing cards. It consists of a StrongARM single board computer (SBC), an 802.11 PCMCIA radio, and the Radio Interconnect Board (RIB). Software on the SBC performs ad hoc wireless network routing in real time. A new CANR design will be much smaller and use much less power than the current prototype.

Much of the development was accomplished using the Stage robot simulator, part of the Player / Stage open source software distribution.[4] Developing in simulation allowed us to test our algorithms and behaviors much more often than using the actual robots. Another benefit for using Player / Stage is that the algorithms and behaviors developed under simulation were directly transferable to the real robots without any modification. To make the simulation more realistic, we used the floor plan of our laboratory as the simulation environment.

4. INTELLIGENT BEHAVIORS

Imagine you are a first responder and your team must navigate inside of a very large indoor structure. To make matters worse, there is no floor plan available, and nobody around to ask directions. However, each member of the team has a radio. Good luck!

This is essentially the concept of operations for teams of mobile robots operating in unknown environments. In order for robots to even begin to compete with humans under these conditions, robots must first possess the requisite sensor suites and intelligent behaviors to successfully navigate indoors. Examples of sensors include stereo vision, laser range-finding, sonar, infrared ranging, and deduced reckoning from internal sensors (such as motor encoders and inertial measurement units). Examples of behaviors include map building, localization, obstacle avoidance, object recognition, and path planning. This section describes the intelligent behaviors that are integrated into the AMCR software.

4.1. Communication & The Publish-Subscribe Paradigm

It was once said that a computer without input and output capabilities is just a paperweight. The same is true for mobile robots in unknown environments; they are of little use without the ability to receive commands or transmit sensor information. But just as it's important for human operators to communicate with robots, it's also important for robots within a team to communicate with each other. There are many multi-robot algorithms currently being developed that rely on timely robot-to-robot communication.

Some teams of mobile robots communicate using a wireless network because of necessity, requirements, or cost. But there are other reasons to prefer the use of RF communication. For instance, audio and visual light signals are not appropriate for stealth missions. Infrared data transmission suffers from limited range. And wired data transfer implies the use of restrictive tethers or some preexisting infrastructure. We will only focus on wireless data transfer in this paper.

As mentioned in the previous section, we use 802.11 radios that form an ad-hoc network via a proactive link state protocol, which automatically routes Internet Protocol (IP) traffic between robots and the operator using the most efficient transmission path. For efficient, real-time robot communication, an application protocol is needed. We have developed such a protocol using a new and powerful paradigm called publish-subscribe. The publish-subscribe paradigm is entirely appropriate to mobile robot communication and meets all of our needs.

In the publish-subscribe paradigm, each robot publishes data about itself as a set of variables. Typical robot variables might include pose and battery life. The act of publishing simply makes these variables available to subscribers. Other robots can then subscribe to the published data and receive updates periodically. Enhancements include the choice to receive data reliably or best effort, and also the frequency of data updates.

Data Distribution Service[5] (DDS) for Real-Time Systems is an Object Modeling Group (OMG) standard that uses publish-subscribe. A DDS implementation called NDDS[6] is commercially available from Real-Time Innovations. They describe their implementation as "real-time publish-subscribe middleware." NDDS is used in many different distributed real-time control systems, including Georgia Tech's autonomous unmanned helicopter[7]. Although there are many advantages to using NDDS, such as graphical development tools and technical support, there are also disadvantages. Current development tool licensing costs are around $50,000 initially and $8,000 annually thereafter. Finished products also require royalty fees. Another major disadvantage is that NDDS is closed-source.

The Player open source software distribution, although not designed as a general-purpose publish-subscribe solution, implements a very limited version of publish-subscribe. Limitations include lack of reliable UDP/IP support (necessary for wireless communication) and predefined variable types. The open-source nature of the software allows it to be extended to include any features desired, but the existing architecture may make Player ill-suited for general purpose communication.

Due to problems with the first two approaches, we have developed a prototype communication middleware system using open standards like Trivial File Transfer Protocol[8] (TFTP) and Extensible Markup Language[9] (XML). Our system, the *ChalkDust Shared Variable Service*, allows multiple robots to share files with each other through a single TFTP server. The human operator hosts the TFTP server on the OCU and can monitor all of the robot communication; this ensures that the human is always kept in the loop.

Most data that we transfer between robots is in an XML file format. This allows us to use standard libraries such as Expat to parse messages. When the robot starts up, it communicates with the operator control unit and downloads its current mission. After parsing the mission file, it knows what other variables to upload and download, and what to do with the data. For instance, the robot uploads its pose at a fixed, predefined, time instance. Other robots can then download this variable to determine the first robot's pose. Variable names can be extended to include the name of the robot, or they can be grouped into the same directory for a particular robot. Anything can be transferred with this system, including robot control software.

4.2. Robot Recognition

In order for the robots to position themselves appropriately, it is necessary for them to be able to detect and recognize each other. We have implemented this behavior using the SICK LMS 200 laser rangefinder and a retro-reflective barcode marker (see Fig. 3). Each robot has a marker. Other robots can then detect the barcode marker using the laser rangefinder by analyzing a range scan and looking for a pattern of highly reflective segments. The binary number encoded by the strips on the barcode is a unique identifier and can be used to detect a specific robot. In addition, robots can also detect the range, bearing, and orientation of other robots using these markers. These capabilities are necessary for both the convoying and convoy localization behaviors.

The algorithm to implement robot recognition is included in the Player / Stage open source software distribution. The Player driver is named *laserbarcode*. A visualization tool included with the distribution allows the user to visually identify barcode markers from the laser range data.

4.3. SLAM

Simultaneous Localization and Mapping (SLAM) is major breakthrough in modern robotics. In order to localize, the robot needs a map of the surrounding environment. But in order to build a map, the robot needs to know precisely where it is! SLAM does both at the same time.

We are using the SLAM algorithm developed by SRI[10] on our iRobot ATRV laboratory robot. We are in the process of porting the software over to ROBART III, and eventually the Pioneers. For now, we are using a pre-built SLAM map for testing in simulation. The map acts as both the environment for the Stage simulation and as the basis for Monte-Carlo Localization (see Section 4.5). Fig. 4 shows a map of our test facility (Battery Woodward) generated using the SLAM algorithm.

Figure 3. A Pioneer robot is shown with a retro-reflective barcode marker. The marker encodes the binary number 10101, which is the number 21 in decimal. In addition to reading this number, another robot can also determine the range, bearing, and orientation of this robot using this marker.

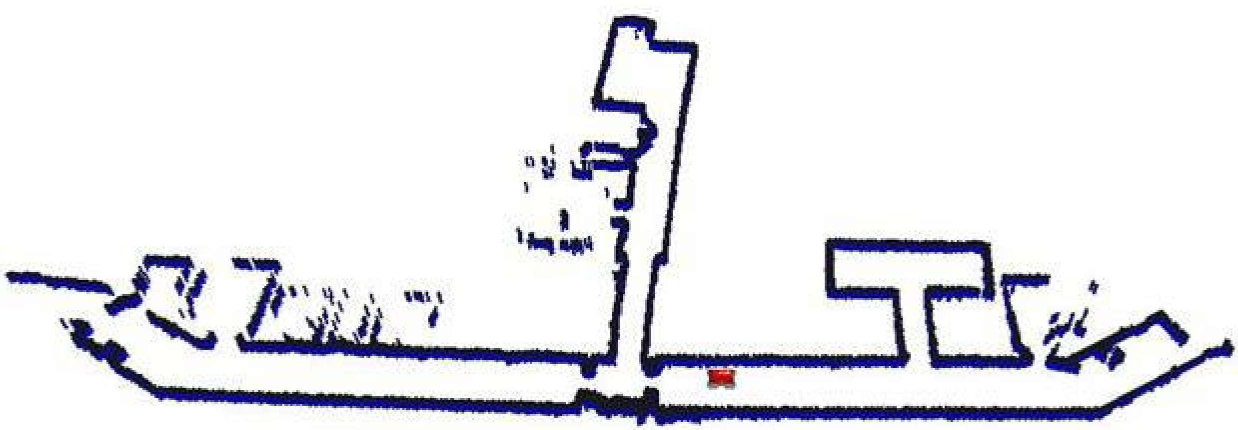

Figure 4. A map created by SRI's SLAM algorithm. This 2D map is being represented in 3D, with the robot overlaid on top.

4.4. Convoy Localization

In an unknown environment with no a priori map, how can multiple robots localize themselves to each other? One way is for each robot to build a map independently, and later synchronize the maps.[11] This technology requires that each robot be capable of mapping the environment on its own. However, we have chosen a simpler approach.

The key idea is that we use the robots themselves as landmarks. The lead robot builds the map using the SLAM algorithm while the convoying robots localize themselves to the leader. We call this method *convoy localization*. Once the map is constructed, the convoying robots can use this map for localization.

Convoy localization requires three capabilities, 1) a pose estimate from the leader using dead reckoning or SLAM, 2) a distance, bearing, and orientation measurement of the robot in front of the current robot, and 3) a way to communicate this information to the other robots in the convoy. The leader starts off with a pose estimate, along with a degree of uncertainty. The robots then convoy inside of the building by detecting the robot in front and following the same path. Fig. 5 illustrates how the robots in the convoy communicate. See Ref. 1 for a detailed description of the convoy localization algorithm.

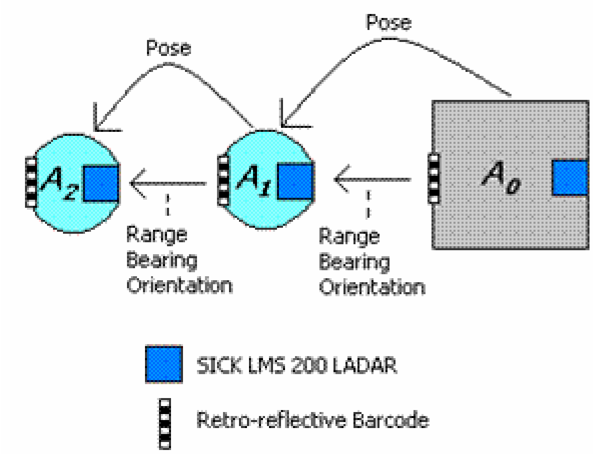

Figure 5. Robot A_0 is the lead robot and creates a map using the SLAM algorithm. Robots A_1 and A_2 follow along in the convoy by detecting the range, bearing, and orientation of the robot just ahead, and then moving to maintain an appropriate distance. At the same time, each robot transmits its pose to the previous robot just behind. Using these two pieces of information, a robot can calculate its current pose with respect to the lead robot. This allows the robots to eventually use the map that the lead robot generates, because they will all have the same frame of reference.

4.5. Adaptive Monte-Carlo Localization

Map-based localization works by figuring out where the robot would need to be on the map in order for its laser rangefinder scans to make sense. There are a number of different techniques to accomplish this. One way is the Monte-Carlo Localization (MCL) technique, which uses a Bayesian inference mechanism. Adaptive Monte-Carlo Localization[12] (AMCL) is an extension to regular MCL where it adaptively changes the size of the particle filter used in making the localization estimate. We are using the AMCL implementation that is part of the Player Robot Server open source software distribution.

Since AMCL works best when it has a reasonably good initial guess of the correct pose, we use the output of the convoy localization algorithm as the initial estimate for the AMCL algorithm. This ensures that the AMCL algorithm will quickly converge to the correct pose. Without this initial estimate, the algorithm would require too much time and memory to converge correctly. Fig. 6 shows the graphical output of the AMCL algorithm.

4.6. Breadcrumb Navigation

A relay node can use a map generated by the SLAM algorithm running on the lead robot to rejoin the convoy. This can be accomplished using path planning technique. However, we chose to use a much simpler method: breadcrumb navigation. The lead robot keeps a record of its pose over time (virtual breadcrumbs) as it traverses and maps the unknown environment. When a relay robot wants to seek out the lead robot, it downloads the map and virtual breadcrumb list from the lead robot, and then navigates from breadcrumb to breadcrumb until it catches up to the lead robot. Although this behavior doesn't allow a robot to move just anywhere, it is simple and fast. It also has the added security of making sure robots only move to areas that are known to be safe. We have used this method in another application that demonstrated a logistics transportation capability through leader following.[13] (In this outdoor application, GPS data replaces the need for a map and AMCL.)

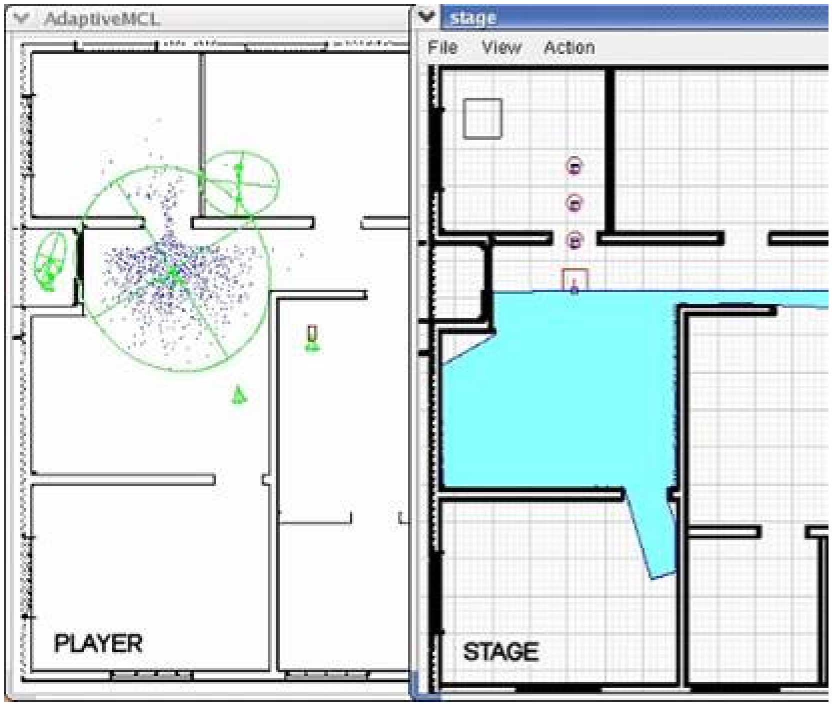

Figure 6. The output of the AMCL algorithm is shown graphically on the left side of the figure. The green circles indicate the uncertainty of the robot's pose. The blue dots indicate the points in the particle filter. The right side of the figure shows the simulation environment used to test the AMCL algorithm. The robot running the algorithm is represented by the rectangle at the front of the convoy. The blue region indicates the free space as seen by the simulated laser rangefinder. There are three smaller relay robots following the lead robot. The black rectangle in the upper-left corner represents the base station.

5. MIXED REALITY INTERFACE

Mixed reality is essentially taking real-world information such as video or maps and combining it with mission specific information such as locations of friendly and enemy forces, way points, areas of contamination, moving objects, and any other information that can be gleaned from an intelligent robot in the area of interest. For instance, it is not enough for a robot to just be able to build a map and localize to that map. A robot should also be able to identify rooms inside of a building and be able to navigate from room to room, without relying on absolute XY coordinates. This added capability will allow an operator to retain situation awareness of the entire mission.

For this purpose, we are developing an intuitive robot interface that uses the concepts of mixed reality. Our current interface, Joint Multi-robot Operator Control Unit (JMOCU), is an important component in SPAWAR's groundbreaking Composable FORCEnet architecture. JMOCU is entirely web-based, and allows an operator to give high-level robot commands from anywhere in the world. The intelligence onboard the robot ensures that these commands can be carried out safely and efficiently. JMOCU already has support for commanding multiple platforms, but only one at a time. Fig. 7 shows a screenshot of the JMOCU software.

Our next version will focus on a point-and-click interface. Point-and-click navigation is a powerful way for a single operator to command a team of mobile robots with very little effort. But it requires that the robots have very sophisticated SLAM, path planning, and obstacle avoidance technologies. The interface is exactly like modern real-time strategy (RTS) video games. The operator selects a single robot or a group of robots using a mouse or some other pointing device. Then, the operator commands the robot or group of robots to perform some action. Typical actions might include, 1) move to this location, 2) patrol this route, 3) convoy behind this leader, 4) search for intruders, 5) stand guard, 6) refuel / recharge, and 7) return to base. The possible actions will be application specific and will also depend on the physical construction and the availability of certain behaviors on the robotic platforms. We plan to integrate all of the intelligent behaviors outlined in this paper into a team of robots that can be commanded through a point-and-click interface.

Figure 7. JMOCU is our web-based robot command interface and is a component in SPAWAR's Composable FORCEnet architecture. Multiple robots of many different types can be commanded to perform various actions. Shown here, an URBOT (Urban Robot) is using GPS to navigate along a selected path on the map. On the left is live video from the URBOT. On the right is an aerial image with information about the robot overlaid on top.

6. CONCLUSION

We have described our current work on several intelligent behaviors and enabling technologies that will allow a team of convoying robots to deploy units at various locations to provide communication relaying for a lead robot, then requesting a map from the lead robot and using it to catch up to the lead robot at a later time. These behaviors include: 1) retro-reflective barcode beacon-based recognition, which allows each robot to recognize other robots in the convoy, 2) convoy localization, which allows each robot to obtain a rough estimate of its location with respect to other robots in the convoy, 3) Adaptive Monte Carlo Localization (AMCL), which allows each robot to use the rough position estimate of the previous step as a seed to obtain its precise location on the map, 4) SLAM, which the lead robot uses to generate the map, and 5) breadcrumb navigation, which each relay robot uses in conjunction with AMCL and the map generated in the last step to seek out the lead robot. Overarching these behaviors is a publish-subscribe protocol that allows the robots to communicate with each other, and a mixed reality interface that allows the user to maintain total situation awareness over the robots, as well as controlling each individual robot or group of robots as the need arises.

ACKNOWLEDGMENTS

This project has been funded by the Defense Advanced Research Projects Agency / Information Processing Technology Office (DARPA/IPTO) under the Mobile Autonomous Robot Software (MARS) program.

We would like to acknowledge Chinh Nguyen for his incredible work on JMOCU and Composable FORCEnet. We would also like to thank Dionysus M. Blazakis at the University of Maryland for his assistance in implementing a prototype version of the *ChalkDust Shared Variable Service*. His work was sponsored by the Naval Research Enterprise Internship Program through the Office of Naval Research.

REFERENCES

1. H.G. Nguyen, N. Pezeshkian, A. Gupta, and N. Farrington, "Maintaining communication link for a robot operating in a hazardous environment," in *Proc. ANS 10th Intl. Conf. on Robotics and Remote Systems for Hazardous Environments*, Gainesville, Florida, March 28-31, 2004.
2. H.R. Everett, E.B. Pacis, G. Kogut, N. Farrington, and S. Khurana, "Towards a warfighter's associate: eliminating the operator control unit," in *Proc. SPIE Mobile Robots XVII*, October 27-28, 2004.
3. H.G. Nguyen, N.M. Farrington, and N. Pezeshkian, "Maintaining communication link for tactical ground robots," AUVSI Unmanned Systems North America 2004, Anaheim, CA, August 3-5, 2004.
4. B.P. Gerkey, R.T. Vaughan, and A. Howard, "The Player/Stage project: tools for multi-robot and distributed sensor systems," in *Proc. IEEE 11th Intl. Conf. Advanced Robotics*, pp 317-323, Coimbra, Portugal, 2003.
5. Object Management Group, *Data Distribution Service for Real-Time Systems Specification*, Final Adopted Specification, May, 2003.
6. G. Pardo-Castellote, S. Schneider, and M. Hamilton, "NDDS: the real-time publish-subscribe middleware," *http://www.rti.com/scripts/literature.php?file=RTPSNETWORK&product=ndds*, 1999.
7. Real-Time Innovations, "Unmanned Georgia Tech helicopter flies with NDDS," *http://controls.ae.gatech.edu/gtar/2000review/rtindds.pdf*, 2001.
8. K. Sollins, "The TFTP Protocol (Revision 2)," RFC 1350, Massachusetts Institute of Technology, 1992.
9. T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)," World Wide Web Consortium, February 04, 2004.
10. J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Intl. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.
11. K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, "Map merging for distributed robot navigation," in *Proc. IEEE Intl. Conf. on Intelligent Robots and Systems*, Las Vegas, 2003.
12. D. Fox, "KLD-Sampling: adaptive particle filters," in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2001.
13. H.G. Nguyen, G. Kogut, N.M. Farrington, et al., "A Segway RMP-based robotic transport system," in *Proc. SPIE Mobile Robots XVII*, October 27-28, 2004.