Evaluation of Supervisory Control Interfaces for Mobile Robot Integration with Tactical Teams*

Alexander Lalejini[1], Dexter Duckworth[2], Richard Sween[1,3], Cindy L. Bethel[1,3], Daniel Carruth[3]

Abstract— **As robotic systems become more sophisticated, they are increasingly called upon to accompany humans in high-stress environments. This research was conducted to support the integration of robotic systems into tactical teams operating in challenging and stressful environments. Robotic systems used to assist tactical teams will need to support some form of autonomy; these systems must be capable of providing operators supervisory control in cases of unpredictable real-time events. An evaluation of the relative effectiveness of three different methods of supervisory control of an autonomously operated mobile robot system was conducted: (1) hand gestures using a Microsoft Kinect, (2) an interactive Android application on a hand-held mobile device, and (3) verbal commands issued through a headset. These methods of supervisory control were compared to a teleoperated robot using a gamepad controller. The results from this pilot study determined that the touchscreen device was the easiest interface to use to override the robot's next intended movement (L^2(3,23)=11.413, *p*=.003, d=1.58) and was considered the easiest interface to use overall (L^2(3,23)=8.078, *p*=.044, d=.93). The results also indicate that the touchscreen device provided the most enjoyable, satisfying, and engaging interface of the four user interfaces evaluated.**

I. INTRODUCTION

Law enforcement Special Weapons and Tactics (SWAT) teams are required to respond to dangerous and often unpredictable environments as part of their official duties. This may involve such high-risk tasks as serving of arrest and search warrants, engaging with active shooters that may be heavily armed, subduing barricaded suspects, negotiating the release of hostages, obtaining intelligence information about criminal activity, and searching large buildings for dangerous suspects. These officers are trained to handle these high-risk situations with minimal force, injury, property damage, and/or loss of life. The use of a mobile robot in these types of incident responses adds a layer of protection between possible threats and the responding tactical team members because it can be operated to make first contact, provide critical intelligence about the environment, and become a distraction that will allow officers to more safely enter these dangerous environments. When robots have been used in these types of responses, an officer is required to be taken out of the fight to teleoperate the robot, which is a hardship particularly on small and/or part-time teams. It is critical to develop methods of supervised autonomy that will allow mobile robots to become more like members of these tactical teams and act as a force multiplier instead of having an officer dedicated solely to the operation of the robot.

The first known investigation related to the use of robots for surveillance was published by Crowley in 1987 and explored the use of mobile robots and the coordination of the actions and the perception capabilities of these robots for surveillance purposes [1]. He proposed a robotic architecture for the control of the robot and the best method of navigating these mobile robots to conduct surveillance in a building environment. This was a seminal paper in robot navigation, control, and perception in building environments.

In 1999, the Defense Advanced Research Projects Agency (DARPA) announced a Tactical Mobile Robotics Program to explore the use of mobile robots in tactical team environments [2]. DARPA at that time was very interested in this area because it was becoming evident that military conflicts would occur in populated, urban environments with buildings and similar types of terrain features. The emphasis of the program was:

"Researching and developing the capability to perform urban reconnaissance with teams of small, low-cost, semiautonomous mobile robots. Easily transportable by individuals, these robot teams will be capable of working together to perform a variety of reconnaissance functions." [2]

After this program announcement to the research community, considerable focus was devoted to this program [3]–[7]. One focus of these research efforts was on the development of small, man-portable robots with some level of autonomy for use in urban terrains; however, the development of interfaces for supervisory control was not explored. A second focus was maintaining communications with these robots in environments that were not communication-friendly because of large amounts of metal and concrete in the structures, which impeded wireless communications and robot controls during teleoperation procedures [7].

Lundberg and Christensen performed preliminary research assessing the use of man-portable robots in law enforcement applications related to tactical missions [8]. This research included a 5-month study using the Packbot Scout robot integrated with a SWAT unit in Sweden. Their data included two sets of interviews with users in high-risk environments and the incorporation and assessment of robots for use in this type of field application. In this research, the robot

[1] Computer Science and Engineering, Mississippi State University, Mississippi State, MS 39762, USA

[2] Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762, USA

[3] Center for Advanced Vehicular Systems, Mississippi State University, Mississippi State, MS 39762, USA

* This research was sponsored by the U.S. Army Research Laboratory Under Grant W911NF-13-1-0481. The views and conclusions contained in this document are those of the author's and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

was teleoperated during trainings and one field response, and there was no consideration of methods of supervised autonomy and control.

For the past two years, members of the Social, Therapeutic, and Robotic Systems (STaRS) research lab at Mississippi State University have collaborated with members of the Starkville Police Department's SWAT team to integrate an unmanned ground vehicle as part of their training events associated with methodical searches of large buildings. The focus has been on the development of an autonomous mobile robot capable of executing reconnaissance, surveillance, and distraction tasks, which provided an additional layer of safety and protection for the team members [8].

The primary goal, is for an autonomous robot to operate alongside the team and it must also be able to serve as a member of the team. In order to coordinate in dangerous and covert situations, a team member will often use hand signals to communicate to the rest of the team his or her next intended action, as well as to give commands to the other team members. Therefore, if a robot is to be successfully integrated with the team, it must be able to communicate its intended actions and operate independently through some method of supervisory control.

A small scale preliminary evaluation was performed to determine the effectiveness of three interfaces to convey a robot's next intended movements: (1) an LED array, (2) an Android smartphone, and (3) a Bluetooth headset. The robot used two types of messages to indicate intent: pending and active. Pending messages were sent before the robot started its intended movement to alert users who were in close proximity to the robot and provide an opportunity for them to respond and move if necessary. Active messages were sent when the robot began movement in the intended direction. The results indicated that participants preferred the LED array, followed by the Android interface, and the Bluetooth headset. Because the LED display could not be used as a method of supervisory control, the Android interface for touchscreen control, the Bluetooth headset for voice commands, and a Microsoft Kinect for hand gestures for control were investigated.

The rest of the paper presents the details of a recent study conducted for the evaluation of methods of supervisory control. Section II discusses background and related work in the literature. Section III discusses the experiment setup and methods used in this study. Sections IV and V provide an analysis and discussion of the results of the study. Finally, conclusions and future work are presented in Section VI.

II. RELATED WORK

As artificial intelligence has become more advanced, robots have become more able to operate with limited human input. As such, supervised autonomy has become a popular field of study, as it allows an artificial agent and a human agent to share control and places less cognitive load on the human operator. Several groups such as [9], [10] have designed supervisory control interfaces; however, these devices are primarily designed for use in controlled industrial or home environments and are likely unsuited for use in hostile or unpredictable environments. Although the SHERPA project [11] does involve an implementation of a supervisory control system in a search and rescue setting, it is an autonomous control node and does not provide any method for human control over the robots.

The project documented in this paper is concerned with comparing several control interfaces to determine which interface provides the most intuitive user experience and usability features. Janssen and Papanikolopoulos [12], Wang and Liu [13], and Wolf *et al.* [14], have shown that a high-level interface often proves to be a superior form of control for directing the navigation of unmanned vehicles. Therefore, it is advantageous to determine the strengths and weaknesses of various control devices when used as supervised autonomy interfaces.

In addition, limited information is available concerning the definition of a well-designed supervisory control interface. Although teleoperation interface design has become a well-researched area, such as [15] and [16], it appears that supervisory control interface design has not been an area of focus. The two areas are similar in concept, but the requirements of each are different, so it is important to determine what factors contribute to an effective supervisory control interface.

III. METHODS

The methods involved in this research study included the development and integration of the hardware interfaces with a modular architecture to coordinate software controls using the Robot Operating System (ROS). After the implementation of the hardware interfaces and software control systems, a pilot study was designed and conducted to evaluate three different methods of supervisory control and those were compared to the use of a gamepad for manual control. An experiment site was designed and constructed for the purposes of this study.

A. System and Controls

1) Hardware: Four devices (shown in Fig. 1) were used in this research to control the robot: (1) a Microsoft Kinect to send commands using gesture recognition, (2) an Android smartphone that could send commands using its touchscreen, (3) an audio headset with attached microphone that was

Fig. 1: a) Microsoft Kinect, b) Android Smartphone, c) Headphones with microphone, d) Logitech gamepad

connected to the Android smartphone and could be used to send verbal commands, and (4) a wireless gamepad for manual teleoperation. The robot used for the study was a TurtleBot 2 (shown in Fig. 2) equipped with an onboard netbook and wireless router. The smartphone and Kinect were connected to the robot via Wi-Fi and used socket connections to send and receive data. The gamepad was connected to the laptop via wireless USB.

2) Software: The software architecture used in this experiment was designed as a distributed control system that supported a high degree of modularity that allowed for hardware components to be easily added or removed as necessary. This architecture was implemented using ROS; details of the developed architecture can be seen in Fig. 4. A publicly available code repository has been set up for sharing of the software developments from this research effort, located at https://github.com/stars-lab/Robot-Intent-and-Control-Project.

ROS allows for any number of independently running software programs to send messages back and forth between the robot and the control devices. The programs achieve this by publishing data across a ROS topic. Topics are named pathways in which data can stream unidirectionally. Modules can receive data from other modules by subscribing to topics. This design keeps each module independent from the others and drastically simplifies the overall software architecture.

The two largest nodes in the architecture are the Command Queue and the TurtleBot Mover. The Command Queue node is the central module responsible for managing movement requests and sending movement commands to the TurtleBot Mover. It has the ability to prioritize messages based on their source or message type (for example, an "Emergency Stop" message would be prioritized over a regular movement message). The Command Queue is also responsible for sending pending and active movement messages to the different intent and control interfaces via the "command_out" topic. The TurtleBot Mover node is responsible for converting movement commands from the Command Queue into actual motor movements and relaying the state of the robot's movement back to the Command Queue via the "mover_status" topic.

The implemented software architecture allows for a high degree of portability to other robotic platforms. To port the system architecture used in this study to other robots, only two modules would need to be modified or exchanged: the odometry server and the TurtleBot movement module.

3) Control Methods: Each control device used a different method for sending an override command to the robot. The Android device received pending and active messages and displayed them to the user via messages on the screen. During a pending message, the user could tap the screen to bring up the override interface. This interface consisted of a left and right arrow. The user tapped the arrow button that corresponded to the direction they wanted the robot to rotate. When the participant used the audio headset connected to the Android device, the pending messages were played through the headset. If the user wanted to override the command, they would tap the override button on the Android device and then speak the command they wanted the robot to execute. These commands were not explicitly defined; the user could say any command as long as it contained a direction, so "Go left", "Turn left", and simply "Left" all produced the same result.

Participants using the Microsoft Kinect viewed a video feed from a camera on the robot that was displayed on a wall in the experiment site. The Microsoft Kinect device was mounted on the wall below the video feed, along with an Android tablet that displayed the pending and active messages to the user. Similar to the Android condition, when the tablet device displayed the pending message, the user could perform a specific gesture to override the intended action of the robot. The gestures for this study simply involved the user sweeping his or her arm across the body. The arm used determined which direction the robot turned (e.g., sweeping the left arm across the body would tell the robot to turn left).

A wireless gamepad device was used as a control condition. Participants used the gamepad to manually control the robot. There was no autonomy to override; participants simply drove the robot through a defined maze. This simulates how tactical teams who work with robots currently control the robot.

4) Experiment Site: This study was carried out in the Human Performance Laboratory at the Center for Advanced Vehicular Systems (CAVS) at Mississippi State University. The maze used in the study was constructed as a modular

Fig. 2: TurtleBot 2 Robot

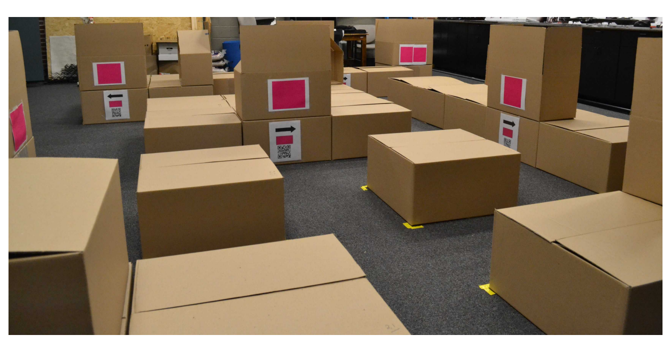

Fig. 3: The maze used for the study.

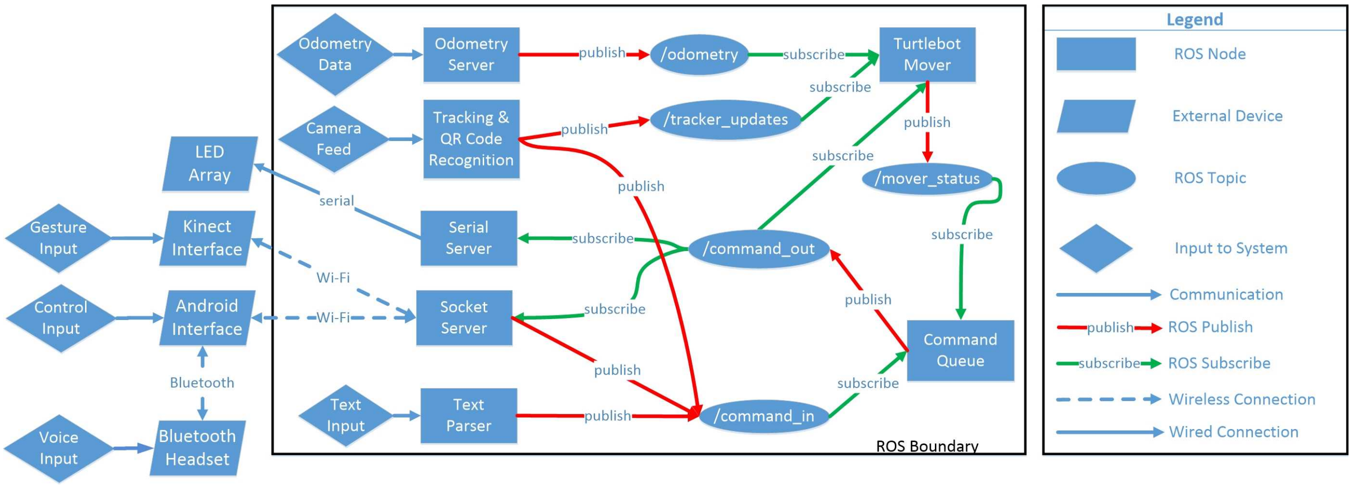

Fig. 4: The ROS Intent and Control Software Architecture

system with multiple pathways of navigation, as shown in Fig. 3. The robot was able to execute a particular path through the maze using instructions embedded in Quick Response Codes (QR codes). These QR codes were mounted at decision points throughout the maze and could be swapped to generate different autonomous navigational behaviors for the robot; this modularity allowed the study to easily implement different maze patterns each time a participant navigated the maze.

Because of an accumulated odometry error, the TurtleBot's odometry data was not sufficient for the robot to accurately navigate the maze according to the instructions embedded in the QR codes. To compensate for this, large pink squares were placed at the end of long corridors in the maze, and a module was implemented that allowed the robot to correct its heading using the pink squares. As the robot moved through a corridor, it used a forward-facing camera to track clusters of pink present in the camera feed.

B. Method

This study was a pilot study designed to evaluate a proof of concept for three different methods of supervisory control. The study was a within-subjects design that evaluated four conditions for the control of a robot: (1) hand gestures using a Microsoft Kinect, (2) an interactive Android touchscreen application on a hand-held mobile device, (3) verbal commands issued through a headset, and (4) the control condition of using a gamepad controller to manually teleoperate the robot.

The study involved a participant following a robot as it autonomously navigated a prefabricated maze described in the Experiment Site section. At each intersection of the maze, a visible arrow was displayed to indicate the correct direction for the robot to navigate. The robot conveyed its next intended direction to move to the participant through one of the control interface methods. At different intersections throughout the maze, the robot indicated to the participant that it planned to execute an incorrect turn. It was the participant's responsibility to use the specified control interface for that condition to override and correct any erroneous decisions made by the robot. In the control condition, the participant used a gamepad to manually navigate the robot through the maze following the direction indicators at the intersection points.

Each participant followed the robot through the maze a total of four times, navigating a different maze pattern each time. The course tested one method of control for each pathway traversed; the four methods were randomized and counterbalanced. After each time through the maze, participants were asked to evaluate the method of control used to assist the robot's navigation. At the end of the study, the participant completed a questionnaire that requested demographic information and answers to general questions about his/her interactions with the robot.

C. Participants

The study population consisted of 23 participants. Of those participants 70% were male and 30% female (16 and 7, respectively). Approximately half of the participants were in the 18-25 age range (48%, 11 participants) with the remaining between the ages of 26 and 65. A majority of the participants had no previous robotics experience (74%, 17 participants), and no participants indicated that they had strong prior robot experience (>4 on a 7 point Likert scale). Thirteen of the 23 participants had prior military or law enforcement experience (57%).

IV. RESULTS

Each participant was asked to answer 19 questions for each of the three methods of supervisory control for this project. These questions included both usability (e.g., *How easy/difficult was it to use this interface to override the robot's intent?* or *How easy/difficult was the interface to use?*) and user experience questions (e.g., *How frustrating was the interface to use?* or *How fun was the interface to use?*). The responses for each survey question were weighted heavily toward the most positive response of the Likert or Semantic Differential Scales used for the survey questions.

It was decided that the categorical variables used for each question would be recoded into dichotomous variables with 1 indicating the most positive response and 2 indicative of all other responses between 2 and 7, with 7 representing the most negative response for each question. Because the data collected was either dichotomous or categorical, it was decided that the appropriate data analysis method would be a Chi-Square test. Additionally, because the sample size for this pilot study was small, it was determined that in order to increase statistical power, the appropriate evaluation would be a Likelihood Ratio Chi-Square (L^2) test.

The results for two usability questions indicated statistically significant results. The first question that was statistically significant was *How easy/difficult was it to use this interface to override the robot's intent?* (see Fig. 5). The results of the Likelihood Ratio Chi-Square were $L^2(2,23)=11.413$, $p=.003$, d=1.58. According to interpretations for Cohen's d for effect size, this was considered a strong effect ($> .8$). The results indicated that the interactive Android touchscreen application on a handheld mobile device was the easiest to use to override the robot's next intended movement (19 participants of 23 rated it with a score of 1 indicating *easy*). There was not a significant difference in the ease of overriding the robot's intent for the voice command using a headset (13 participants rated it with a 1 for easy and 9 rated it with some other rating between 2 and 7). There was not a statistically significant result for hand gestures using the Microsoft Kinect (9 participants rated is with a 1 for easy, and 14 rated it with some other rating between 2 and 7). For a limited number of participants, data was missing for some of the devices.

Another question had a statistically significant result, which involved the survey question *How easy/difficult was the interface to use?* (see Fig. 6). The results of the Likelihood Ratio Chi-Square test were $L^2(3,23)=8.078$, $p=.044$, d=.93. Based on the interpretation of Cohen's d for effect size, this was a strong effect ($> .8$). The result from this question indicated that 19 of 23 participants found the interactive Android touchscreen application on a handheld device was the easiest interface to use. Only three participants rated

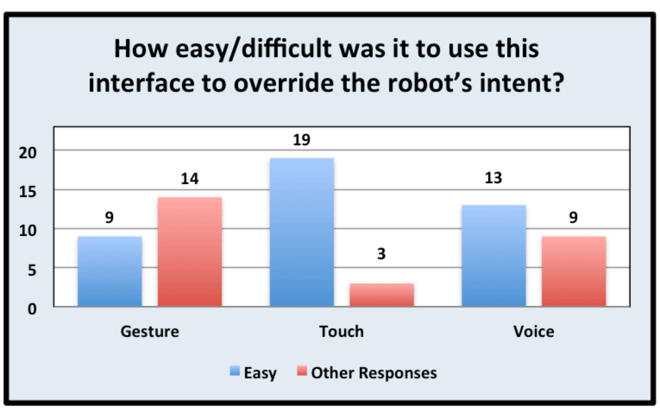

Fig. 5: Statistically significant results for the question *How easy/difficult was it to use this interface to override the robot's intent?* with recoded variables.

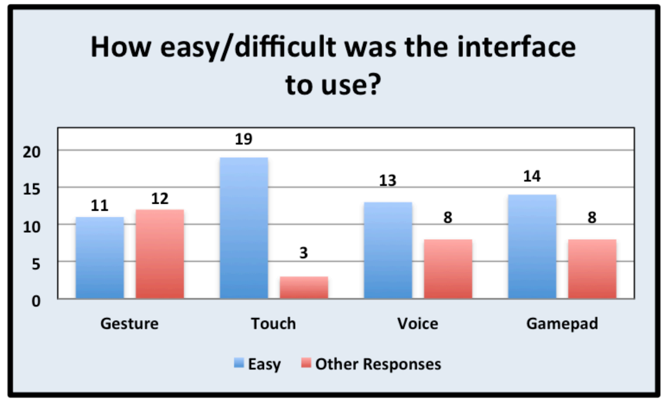

Fig. 6: Statistically significant results for *How easy was the interface to use?* with recoded variables.

it with some other rating. The use of voice commands using a headset had no statistically significant difference, with 13 participants rating it a 1 for easy and 8 with some other response. The results were also not statistically significant for hand gestures using a Microsoft Kinect (11 rated it as easy, and 12 some other response) or for the manually operated gamepad (14 rated it as easy, and 8 rated it as some other response). For a limited number of participants, responses were not recorded for some interfaces and the data points were missing.

For the most part, the participants reported their interactions with all four methods of control as fun, enjoyable, exciting, and satisfying from a user experience perspective. In the case of the interactive Android touchscreen application on a handheld device, all but one participant rated the experience as positive, checking user experience responses for fun, satisfying, exciting, engaging, and/or enjoyable. The one participant rated his experience with this interface as stressful and frustrating.

V. DISCUSSION

The results demonstrate that the Android interface was both the easiest to use in general and specifically for overriding the robot's next intended movements. Several factors could be responsible for this result. The most obvious reason may be that people are more comfortable using a device that is familiar to them; both the form factor and visual style of the Android interface would be instantly familiar to smartphone users. Another factor could be that the Android interface was the most straightforward to use - participants only needed to tap the screen, then tap the arrow that corresponded to the direction they wanted the robot to turn. For the voice recognition, participants had to tap a button on the screen, then recall a valid voice command. The Kinect gesture recognition condition had its own issues as described in the following paragraph. However, one key point raised by a participant was the fact that using the Android interface required them to "split attention between the interface, the robot, and the environment". Future improvements to the Android interface may include video feeds from the robot to improve awareness of the environment, in addition to the integration of a system for displaying the robot's state on the interface to reduce the demand for directly monitoring the robot.

Based on anecdotal comments, participants found the gestures used with the Microsoft Kinect "counter-intuitive" and "weird." Participants also noted that there were issues in the accuracy of the gesture recognition, with one participant saying, "I scratched my face, and the system sent a stop message." This would likely explain why the Kinect interface received lower responses in the survey results.

Comments on the gamepad controller were generally positive, noting that several participants liked the complete control of the robot's movements this interface provided to the user. Participants also noted how "precise" the control was and how fluid the robot's movements appeared. However, as mentioned earlier in this paper and echoed by one participant, using this interface prevented the user from doing anything else with their hands, diminishing their effectiveness as a teammate.

VI. CONCLUSIONS AND FUTURE WORK

The development of a user interface for the supervisory control of a robot is an area that has not received significant attention from the research community, especially as it relates to the integration of a robot as a member of a tactical team. The development and evaluation of an effective method of supervisory control of a robot is essential for SWAT teams. This is very important for smaller municipalities, that may not have the resources to staff and train a full-time tactical team. These teams cannot afford to take an officer "out of the fight" to manually operate a robot in unpredictable and often hostile environments. This study proposed and evaluated three supervisory control methods for the operation of a robot: gesture recognition using a Microsoft Kinect, an interactive Android touchscreen interface, and verbal commands through a headset.

The results presented in this study indicate that, out of the three interfaces tested, the interactive Android touchscreen application was both the easiest interface to use in general and to override the robot's next intended movement. Based on participant feedback, hand gestures interpreted by a Microsoft Kinect resulted in the least desirable user experience. The participant feedback on the voice command interface was neutral for most measures. Overall the study received positive feedback from most of the participants.

The results from this pilot study will be used to adjust future participant surveys to better differentiate the strengths and weaknesses of each interface, and to determine why specific interfaces received certain ratings. In future studies, gesture interfaces such as a more sensitive wearable system (e.g., JPL's BioSleeve [14]) could provide an overall better user experience. Future work includes the integration and evaluation of an interactive Android touchscreen application for supervisory control ported to the robotic system currently used in training exercises with the Starkville Police Department SWAT team.

ACKNOWLEDGMENTS

The authors wish to thank Paul Barrett, John Kelly, Jacob Mason, Malcolm McCullum, and Nathan Smith for their assistance on the development team for this project, as well as Brendan Cogley, Kayla Huddleston, Daniel Waddell, and Jesse Williams for assisting with data collection. We would also like to thank the Starkville Police Department and the Mississippi State University G.V. "Sonny" Montgomery Center for America's Veterans for their support and assistance.

REFERENCES

[1] J. L. Crowley, "Coordination of action and perception in a surveillance robot," *IEEE Expert*, vol. 2, no. 4, pp. 32–43, 1987.

[2] E. Krotkov and J. Blitch, "The defense advanced research projects agency (darpa) tactical mobile robotics program," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 769–776, 1999.

[3] R. C. Arkin, "Real-time cooperative behavior for tactical mobile robot teams - skills impact study for tactical mobile operational units," Georgia Institute of Technology-DARPA/ATO, Tech. Rep., November 2000.

[4] R. C. Arkin, T. R. Collins, and Y. Endo, "Tactical mobile robot mission specification and execution," in *SPIE 3838, Mobile Robots XIV*, vol. 3838, no. 150. SPIE–The International Society of Optical Engineering, November 15 1999.

[5] M. Barnes, H. R. Everett, and P. Rudakevych, "Throwbot: Design considerations for a man-portable throwable robot," vol. 5804, no. 511. SPIE - The International Society for Optical Engineering, June 02 2005.

[6] T. R. Collins, R. C. Arkin, M. J. Cramer, and Y. Endo, "Field results for tactical mobile robot missions," Georgia Institute of Technology, Tech. Rep., 2006.

[7] H. G. Nguyen, N. Farrington, and N. Pezeshkian, "Maintaining communication link for tactical ground robots," Space and Naval Warfare Systems Center, Tech. Rep., August 2004.

[8] C. L. Bethel, D. Carruth, and T. Garrison, "Discoveries from integrating robots into swat team training exercises," in *10th IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, November 5-8 2012, pp. 1–8.

[9] C. Fischer, M. Buss, and G. K. Schmidt, "Hierarchical supervisory control of service robot using human-robot-interface," in *1996 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96)*, vol. 3, November 4-8 1996, pp. 1408–1416.

[10] I. M. Begg, J. Gnocato, and W. E. Moore, "A prototype intelligent user interface for real-time supervisory control systems," in *1st International Conference on Intelligent User Interfaces (IUI '93)*. New York, NY, USA: ACM, 1993, pp. 211–214.

[11] M. Furci, A. Paoli, and R. Naldi, "A supervisory control strategy for robot-assisted search and rescue in hostile environments," in *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*. IEEE, September 2013, pp. 1–4.

[12] M. Janssen and N. Papanikolopoulos, "Utilizing queued actions to increase interaction efficiency in robot control interfaces," in *2013 21st Mediteranean Conference on Control and Automation (MED)*. IEEE, June 25-28 2013, pp. 34–39.

[13] M. Wang and J. N. K. Liu, "A novel teleoperation paradigm for human-robot interaction," in *2004 IEEE Conference on Robotics, Automation, and Mechatronics*, IEEE, Ed., vol. 1, December 1-3 2004, pp. 13–18.

[14] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the jpl biosleeve," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 6-10 2013, pp. 1160–1165.

[15] W.-D. Lai, "Zigbee remote control interface for a robot dog design for older users," in *5th International Conference on Ubiquitous Information Management and Communication (ICUIMC '11)*. ACM, 2011, pp. 84:1–84:5.

[16] E. Solovey, K. Jackson, and M. Cummings, "Collision avoidance interface for safe piloting of unmanned vehicles using a mobile device," in *25th Annual ACM Symposium on User Interface Software and Technology (UIST)*. New York, NY, USA: ACM, October 7-10 2012, pp. 77–78.