1-1-2011

An augmented reality interface for multi-robot tele-operation and control

Sam Yung-Sen Lee
Wayne State University,

AN AUGMENTED REALITY INTERFACE FOR MULTI-ROBOT TELE-OPERATION AND CONTROL

by

SAM YUNG-SEN LEE

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2011

MAJOR: ELECTRICAL ENGINEERING

Approved by:

____

Advisor Date

____

____

____

____

DEDICATION

For Mom and Dad

ACKNOWLEDGMENTS

First and foremost, I express my heartfelt thanks to my thesis advisor, Dr. Pandya for this mentoring, encouragement, patience, and wisdom during my time at Wayne State University. I also sincerely thank Dr. Ellis for his many helpful suggestions and support of my thesis work. His advice improved my thesis in many areas.

I would like to thank Dr. Hassoun for his support and his valuable feedback. I would like to thank Dr. Auner for his support and for serving on my dissertation committee.

Thank you to Nathan Lucas, who has patiently helped me on the subject testing, statistics, and edits.

I would like also to thank the entire CARES group for all the suggestions, kind help and making the robotics lab a fun place to work. Thank you so much to all of you.

Finally, to my beloved wife Wan-Chuan, thanks for providing continued support and encouragement these past years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

The field of mobile robotics has made important progress in the past decade. Advancements in mobile robotic technology have the potential for reducing human risk in hazardous environments. Robots are now found in practical applications, such as clearing roads of improvised explosive devices (IEDs), location and destruction of mines, guarding boarders and building, and space exploration. For example, in military applications, explosive ordnance disposal (EOD) is one of the most dangerous activities for a soldier and the threat from roadside bombs is increasing. Both QinetiQ's Talon [1] and iRobot's PackBot [2] robots are being used in Iraq and Afghanistan, and the bomb disposal robots are able to be telecontrolled by a human operator to search and remove explosive devices [3]. Bomb disposal robots have also found application in law enforcement [4]. Law enforcement officers use them for hazardous material handling and bomb disposal. Moreover, in search and rescue missions, a mobile robot has been used to assist humans to find the victims in collapsed buildings [5], and is used for the detection and identification of landmines [6]. In space, the twin-rovers Spirit and Opportunity have been successfully deployed to the surface of Mars and have returned extremely useful information and images and have been a boon to NASA's planetary missions.

In these challenging application domains, however, many missions are complex and cannot be performed by a single robot alone. Multi-robot systems can often deal with tasks that are difficult. For example, teams of robots can complete tasks such as multipoint surveillance, cooperative transport, and explorations in hazardous environments. Additionally, time-critical missions may require the use of multiple robots working simultaneously to efficiently accomplish the tasks. For instance, the Multi Autonomous Ground-robotic International Challenge (MAGIC 2010) [7] has successfully demonstrated the use of ground robotic teams that can execute an intelligence, surveillance and reconnaissance mission in military operations and civilian emergency situations.

The aim of the current robotics technology is to increase the level of autonomy, but until robots permit effective fully autonomy, the human operator cannot be removed from the loop. Feedback [8] from explosive ordnance disposal (EOD) robot users conveyed that cutting the operator out of the control loop could possibly limit them in their ability to effectively inspect and survey the surrounding environment. Particularly, in multi-robot scenarios, human-in-the-loop control will be required because operators must supply the changing, goals that direct multi-robot activity [9]. Results [10] confirm that having a human in the loop improves task performance, especially with larger numbers of robots. Thus, there is a need to research and develop technologies that can enable an operator to control groups of robots more effectively.

It is necessary to create user interfaces that support efficient human robot interaction. Human robot interfaces for supervision and control of multiple robots must be very different from single robot interfaces. Multi-robot control increases the difficulty of an operator's task because the operator has to shift his attention among robots. Moreover, increasing the number of robots will increase the complexity of data and information generated by robots and will increase an operators' workload significantly. The use of Virtual Reality (VR) and Augmented Reality (AR) displays to improve operator performance for ground robot tele-operation are discussed in this thesis.

A Virtual Reality (VR) interface assists in tele-operation control in several ways. A virtual environment provides an external perspective which allows the operator to see the environment and the robots. The user is able to see all of the robots in action in the interface from many angles. Without the VR interface, the user would only see the video from the on-board cameras. Particularly, a Virtual Reality interface allows the user to see the robot's location relative to the other robots.

The Augmented Reality (AR) system described in this thesis makes it possible to overlay planning, sensory data and status information provided by robots over the user's camera field of view. Hence, the goal of this work is to create new Augmented Reality interface technology that integrates with imaging, sensing and robotics systems that the operator uses, and compare it with existing technology.

1.1 Motivation and Problem Statement

One of the main new technologies developed and tested here is human multi-robot interaction based on Augmented Reality. AR allows computer generated 3D model from data of mission sensors or internal states of robots to merge with the real world in real time. The virtual objects display information that the operator of the system may not able to see directly, for example, future planned trajectories of a robot, the surrounding terrain information or sensor data. Moreover, mobile robot control requires position sensing, which can be achieved with Augmented Reality provided pose (position and orientation data) measurement capability. The hypothesis of this work is that applying advanced technology for human multi-robot tele-operation and control will enhance the performance of cooperative robots for practical applications. It will also reduce the operator's workload and promote situational awareness. In this thesis, we focus on the development, the usability testing and comparison of a relatively new technology (human multi-robot interface based on Augmented Reality) to traditional interfaces like joystick control or virtual displays. We have used human factors testing and evaluation to determine the validity and usability of the developed system to estimate if the human multi-robot interaction technology actually improves operator performance.

Most human-robot interfaces implement joysticks combined with live video as the main method of control and focus on relaying data and status messages collected from the robot. Hence, a conventional interface consists of several separated display windows to show information from the robot. For example, the iRobot® Packbot® Operator Control Unit (OCU) and the QinetiQ TALON robot controller use joysticks as input devices, and a video panel to display information in a similar manner. The display may require the operator to integrate complex information, and this may increase the operator's workload. The traditional operator control unit (OCU) or graphic user interface (GUI) fails to provide the user with a practical and efficient interface because it only allows the user to focus his attention on one robot at a time. An alternative to conventional tele-operation interfaces is a 3D virtual environment display based on a robot simulation or an amalgamation of the benefits of physical and virtual reality is Augmented Reality [11, 12]: an advanced visualization technology that allows computer generated virtual images to merge with video views of physical objects in real time. AR can be designed as a bilateral means of communicating and controlling groups of robots. On one hand, the AR interface allows the user to interact with the 3D virtual environment to manipulate the physical robots. The physical robots also provide alerts, sensor data, task timelines, and progress to goals directly on the augmented view registered to the 3D location.

1.2 Research Objective and Specific Aims

This research describes the development, implementation, and usability testing of the Augmented Reality human multi-robot system. The AR system uses a top-down view camera with a team of mobile robots to develop the system. The research and dissertation focus on two types of robot operation cases. First, there is the case of multi-robot command and control requiring high-level instructions. The second operation case provides data and status information from robots for operators in an Augmented Reality view. This research combines these two operation cases in an intuitive manner. Figure 1 represents the Augmented Reality system for remote multi-robot control used in this research.

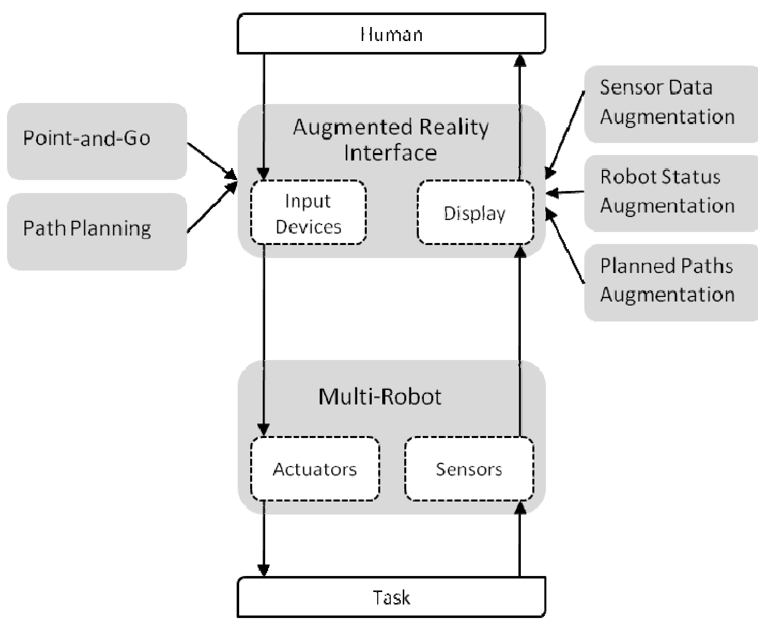


Figure 1: AR interface has as inputs high-level instructions, path planning algorithms, and sensor information, etc.

The main technical objective for this work is to develop an AR human multi-robot interface to improve the operator's performance of tasks such as contaminant localization. The underlying hypotheses are that (1) A human operator in the control loop performing cooperative algorithms will be able to effectively control multiple robots behaving semi-autonomously. (2) An Augmented Reality system will significantly improve the performance of multi-robot tele-operation and control.

The specific aims of this thesis are as follows:

1. Develop human multi-robot system with Augmented Reality.
2. Develop human multi-robot coordinated control for contaminant localization.
3. Perform a subject study to evaluate the Augmented Reality human multi-robot interface performance.

The main aim of this research is to demonstrate a novel human multi-robot interface that has the potential to improve performance and logistics between cooperative multi-agent teams for practical applications.

1.3 Organization of the Thesis

The remaining chapters of this dissertation are organized as follows. Chapter 2 provides a review of the literature that is relevant to this work. Chapter 3 presents the development and implementation of the virtual environment system to assist in tele-operation control. Chapter 4 presents the development and implementation of the AR human multi-robot system. Chapter 5 describes the evaluation of the system using a series of subject testing. Finally, Chapter 6 contains the closing remarks of the work, summarizing the main contributions of this thesis and outlining the potential areas of future work.

Chapter 2: Background and Related Work

This chapter begins by discussing related work on multi-robot control, and is followed by a review of the current state of human robot interfaces. The chapter ends by providing a summary of the lessons learned in the review of the current state of visual interfaces for robotic tele-operation. This work is separated into Virtual Reality interface and Augmented Reality interface.

2.1 Multi-robot Control

2.1.1 Multi-robot Level of Autonomy

In multi-robot scenarios, the level of automation is a key and critical issue. How many remote robots a signal human can manage is dependent on the level of autonomy (LOA) of the robots [13]. The idea of many levels of automation have been discussed in the literature [14]. Various levels of automation which specify the degree to which a task is automated are possible. Luck et al. [15] proposed four levels of automation, (1) full tele-operation, (2) guarded tele-operation, (3) autonomous obstacle avoidance, and (4) full autonomy. A recent study [16] found that increasing autonomy allows robots to have longer neglect time making it possible for a single operator to control more robots. In addition, Crandall et al. [17] presented a class of metrics to measure which autonomy levels they should employ, and how many robots should be in the team for supervisory control of multiple robots.

Thus, until the robotics technology is fully autonomous, multiple robots in a team inevitably require the operator in the control loop to process more data and issue more commands. Adams developed the Multiple Agent Supervisory Control (MASC) system [18] which included four heterogeneous mobile ground-based robots and the MASC human-robotic interface. The system permits the robots to work autonomously until the human supervisor is requested to take control or detects a problem. Adams [19] evaluated the system in which the human's perceived workload and performance to complete tasks with one, two, and four mobile robots for indoor material transportation tasks. In the results, little difference was found between the one and two-robot tasks; however the human supervisor's perceived workload significantly increased during the four-robot task, and there was a significant decrease in the number of tasks successfully completed due to their perceived workload and task completion times. He concluded that it is important to develop tools that will assist the human by guiding interactions and minimizing or optimizing the number of times the human switches between robots.

2.1.2 Multi-robot System and Behavior-based Control

In recent years, a number of researchers have worked on multi-robot systems. The following sections provide a review of recent developments in multi-robot systems and brief overview of their behavior-based control.

In the GUARDIANS (Group of Unmanned Assistant Robots Deployed in Aggregative Navigation by Scent) project; the robots autonomously navigate the site filled with black smoke that makes it very difficult for the firefighters to orientate themselves in the building. The robots serve as a guide for firefighters in finding the target location or in avoiding dangerous locations or objects. One of the aims of this project is to design a human-robot swarm interface for supporting firefighting operation.

Naghsh et al. [20] proposed human-robot interaction using tactile and visual interfaces in the GUARDIANS project. A tactile interface is attached to the fire-fighters torso, and the frequency and amplitude will be used to communicate the seriousness of hazards. A visual LED-based interface is installed within the firefighters' helmet. The visual device displays the directions from swarm robots to lead the fire-fighters to a point of interest. A new approach for robot deployment and building a map of the environment has also proposed in the project [21]. Two behavior-based formation control of the mixed human-robot team has developed in the GUARDIANS project: formation generation and formation keeping [22, 23].

The swarm-bot [24, 25] is a robotic system composed of a swarm of small robots, called s-bots, and capable of self-assembly [26-28] to adapt to its environment. The s-bot stands out among other projects because of the utilization of strong grippers to hold others to form complex structures. For example, those s-bots can self-assemble and build a structure that avoids a hole or pass a trough [29]. They also showed chains of robots can be used for forming a path between two objects [30]. The Swarmanoid project is built on the results obtained during the Swarm-bots project. The Swarmanoid project is the first to study the design and control a heterogeneous swarm robotic system. It is comprised of three types of autonomous robots: eye-bots [31], hand-bots [32], and foot-bots. The project has developed and studied numerous distributed algorithm and communication for the multi-robot system. For example, the heterogeneous recruitment system [33] allows eye-bots to search for tasks, and then recruit groups of foot-bots to perform the various tasks they have found.

The iRobot Swarm [34] is a robot swarm of over 100 units. The individual modules, SwarmBots, are five inch cubes and have a suite of sensors, communications hardware and human interface devices. McLurkin et al. [35, 36] demonstrated distributed algorithms for configuration control in the robot swarm, these algorithms includes a dispersion algorithm [37] and a distributed mapping and localization algorithm [38].

A cooperative multivehicle test-bed (COMET) [39] has been created to facilitate the development of cooperative control systems and mobile sensor networks. This platform is used to implement and validate new cooperative control techniques including formation control and goal seeking.

2.2 Human-Robot Interfaces

Most human-robot interfaces for robot control have focused on providing users data collected by the robot and giving status messages about what the robot is doing. The conventional interface consists of several separate display windows to show information from the robot [40]. The human-robot interface [41, 42] is an example of a conventional display from Idaho National Engineering and Environmental Laboratory (INEEL). It displays streaming video from the robot, information on the state of sensors, and a variety of information including pitch, roll, power, heading, and speed. The display may require the operator to integrate information, and this may increase the operator's workload. Another example of a conventional interface for multiple robots control was designed by Humphrey et al. [43]. This interface was comprised of a camera feed, halo area, status bar, radar, and the control panel. The halo area surrounding the camera feed window presents the other robots' location relative to the selected robot. The status bar indicates for each robot the time remaining until task completion and when the robot requires the operator's attention. Moreover, an example of a multiple robots interface was designed by Envarli et al. [44]. Their interface consists of a main map that conveys the location and the status of each robot in the environment, and a task management window, a group information window, and a user task window. Operators may have a high workload from needing to simultaneously integrate each status bar.

Fong et al. [45] presented a portable vehicle tele-operation interfaces using a personal digital assistant (PDA) [46, 47] with collaborative control [48, 49] for multi-robot remote driving [50, 51]. They discussed the use of collaboration, human-robot dialogue [52] and waypoint-based driving that can enable a operator to effectively control a team of robots.

A touch-based input may allow users to perform complex tasks in an intuitive manner [53]. Micire et al. [54] studied the control of a single agent with a multi-touch table. Moreover, a multi-touch (DREAM) controller [55, 56] using a multi-touch table was developed for multi-robot command and control [57, 58]. Kato et al. also proposed an intuitive interface using a multi-touch display to control multiple mobile robots simultaneously.[59].

2.3 Visual Interfaces for Robotic Tele-operation

2.3.1 Virtual Reality Interface

An alternative to a conventional interface is a 3D virtual environment display based on a robot simulation. In contrast to direct interfaces, a virtual environment provides an external perspective which allows the operator to see the environment and drive the robot from viewpoints generated by the interface. Nguyen et al. [60] describe a

"Viz" software that converts 2D stereo images in to 3D Virtual Reality (VR) based interface for space exploration. The Viz software has shown that VR interfaces can automatically generate 3D terrain models [61]. This can help the user understand and analyze the robot's surroundings and improve his situational awareness. The Rover Sequencing and Visualization Program (RSVP) for operating a rover on the Martian surface is another example of a virtual environment interface. It has been used to plan all rover traverses and produce terrain models [62] that enables quick understanding of the rover's state relative to its environment.

Mollet et al. showed a virtual reality interface, Collaborative Virtual Environments (CVE) [63], for tele-operating a multi-robot system. A group of robots in action can be seen in the interface from many angles. The collaborative system is designed for allowing a group of tele-operator to control teams of robots [64].

2.3.2 Augmented Reality Interfaces

Unlike VR, the user enters and interacts with computer-generated 3D environments, AR allows the user to interact with the virtual images using real objects [65]. Several researchers in robotics are beginning to use AR techniques in robotics because it provides a spatial dialogue for human-robot collaboration [66-68]. Previous work on using AR to enhance human-robot interface has been done. For example, Chintamani et al. [69] showed the benefit of using AR cues in remote robot arm tele-operation, and it resulted in significant improvements in robotic control performance. Giesler et al. [70] implemented an AR system that creates a topological map in an unknown environment to control a mobile robot by pointing to a location using fiducial markers attached to a wand. In the medical domain, Wang et al. [71] produced an AR facility specifically for the da Vinci surgical system to improve visualization during a robotic minimally invasive surgical procedures that will allow the surgeon to view information overlaid onto the view of the operating scene in real time.

AR has also been used to display robot input, output and state information within the real environment [72]. Collett et al. [73] developed a software tool in Player/Stage project using AR for visualization of robot data, including sonar and laser data as well as odometry history of a robot. Payton et al. [74, 75] introduced the concept of viewing the path information by using an AR technique for a robot swarm communicating information to humans. Coral et al. proposed [76] an Augmented Reality visual interface for wireless sensor networks for medical staff to monitor real time information from different kind of sensors attached to patients. Young et al. [77, 78] used AR to display bubblegrams, which are graphic balloons that appear above a robot to allow for interaction between humans and robots. Green et al. also used Augmented Reality to display the internal state of a mobile robot and its intended actions in human-robot collaboration [79-81].

A robot vision system, Virtual and Augmented Collaborative Environment (VACE) [82], allows the user to see the real environment from the robot's camera in AR view, and the user can switch anytime from the real to the virtual view [83]. Without switching between two views, Nielsen et al. [84, 85] presented an ecological interface using augmented virtuality [86]. The interface includes a 3D virtual environment, as well as a video feed from the robot's camera. The video image is displayed in the virtual environment as the information relates to the orientation of the camera on the robot.

Chapter 3: Robot Tele-operation through a Virtual Reality Interface

3.1 Introduction

This chapter describes the development of a Virtual Reality system. A high level instruction, Drag-to-Move, allows an operator to tele-operate multi-robot through the Virtual Reality interface. The 3D models of ODIS and the mast are simulated in the virtual environment.

3.2 Tele-operation Test Bed Development

3.2.1 Team of robots

A team of heterogeneous robots with different dynamics or capabilities could perform a wide variety of tasks. In this study, we work with heterogeneous ground robots as follows. One of the robotic platforms was the Omni-Directional Inspection System (ODIS) robot that shown in Figure 2. ODIS is an omindirectional platform capable of translating in any direction and rotating simultaneously. The basic ODIS platform carried a video camera with tilt actuation, and was originally designed for underbody inspection. ODIS's omnidirectional drive is implemented by a three-wheel drive system, in which all wheels are capable of independent pivot and rotation. ODIS weighs approximately forty pounds and is about four inches tall and 22"x22". The operator control unit (OCU) consists of a joystick that is able to issue command to ODIS to translate and/or rotate at some speed and direction, and a monitor to display the video from the camera. ODIS has low ground clearance, and was designed for relatively smooth, flat and level surfaces.

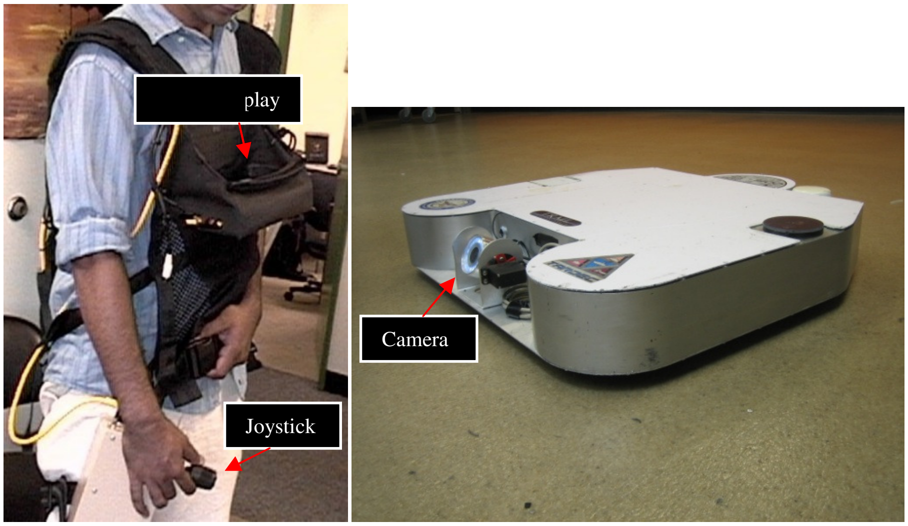


Figure 2: Omni-Directional Inspection System (ODIS) robot is an omindirectional platform capable of translating in any direction and rotating simultaneously. The left side shows the ODIS OCU includes a joystick and a monitor to display the video feed from the on board camera. The right side shows the basic ODIS platform carried a video camera with tilt actuation, and was originally designed for underbody inspection.

Another one of the platforms was the SRV-1 robot that shown in Figure 3. The SRV-1 robot supports wireless network access so that the robots can be control wirelessly. A video camera is on the SRV-1 robot and the live video from any SRV-1 robot can be viewed to support reconnaissance missions. Both ODIS and SRV-1 robot are small enough to operate in a laboratory environment.

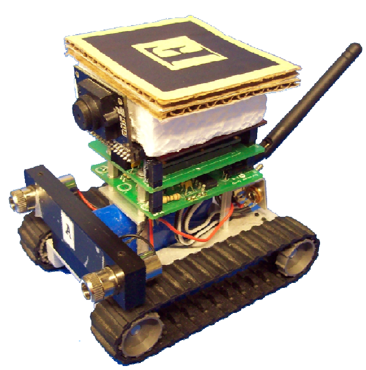


Figure 3: The SRV-1 robot supports wireless network access so that the robots can be control wirelessly. A video camera is on the SRV-1 robot.

3.3 Virtual Reality Interface for Robot Command and Control

3.3.1 Drag-to-Move

An operator interface was developed to control ground robots by using high-level commands in a virtual environment. An operator can select a virtual robot, and then drag it to the target location. The corresponding real robot will be moved to the target position in the real environment similar to the position in the virtual environment. In the virtual interface, once the position of the virtual robot is changed, the system calculates the relative orientation between the new position and the previous position. The real robot first rotates in the direction of the goal position, and then the real robot translates to the new position. At the new position, the real robot rotates until its orientation approximates the virtual robot's orientation. Several sources of error could cause the accuracy of position between real robots and their corresponding virtual robots, for instance, tracking the position of the real robot, wheel slippage, robot power, and error in modeling the actual robot, etc.

Figure 4 and Figure 5 illustrate a real robot controlled by an operator using a Drag-to-Move instruction over the virtual reality interface. This method provides capabilities that reduce the number of required commands to control multi-robots.

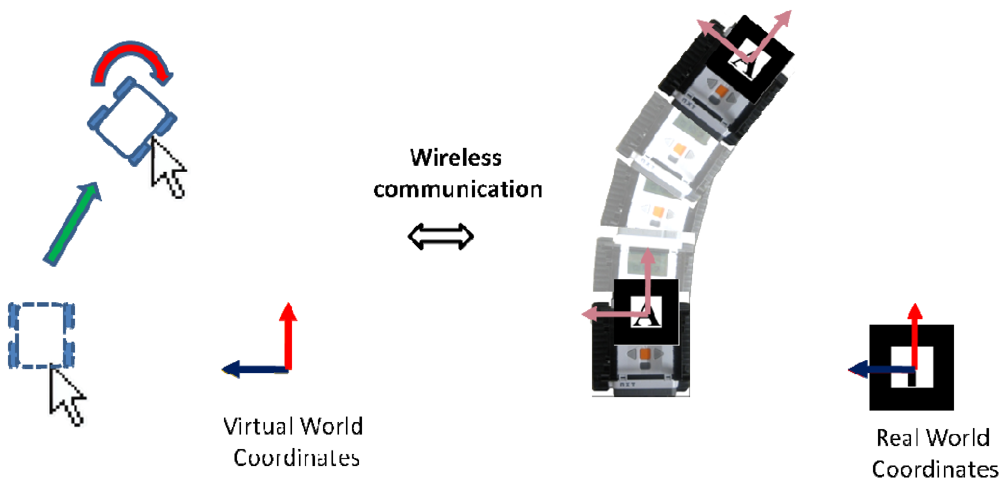


Figure 4: The diagram shows a Drag-to-Move method which allows an operator to control a real robot from a virtual robot over the VR interface.

The Augmented Reality system described here is used in the test bed to track the position of real robots using ARToolKit library [87], and also to register the real robots to the virtual robots. In the real environment, a fiducial marker was fixed on the floor of the test bed as real world coordinates. The real world coordinates was registered to the world coordinates in the virtual environment. The coordinate system allows linking every real robot to its virtual counterpart.

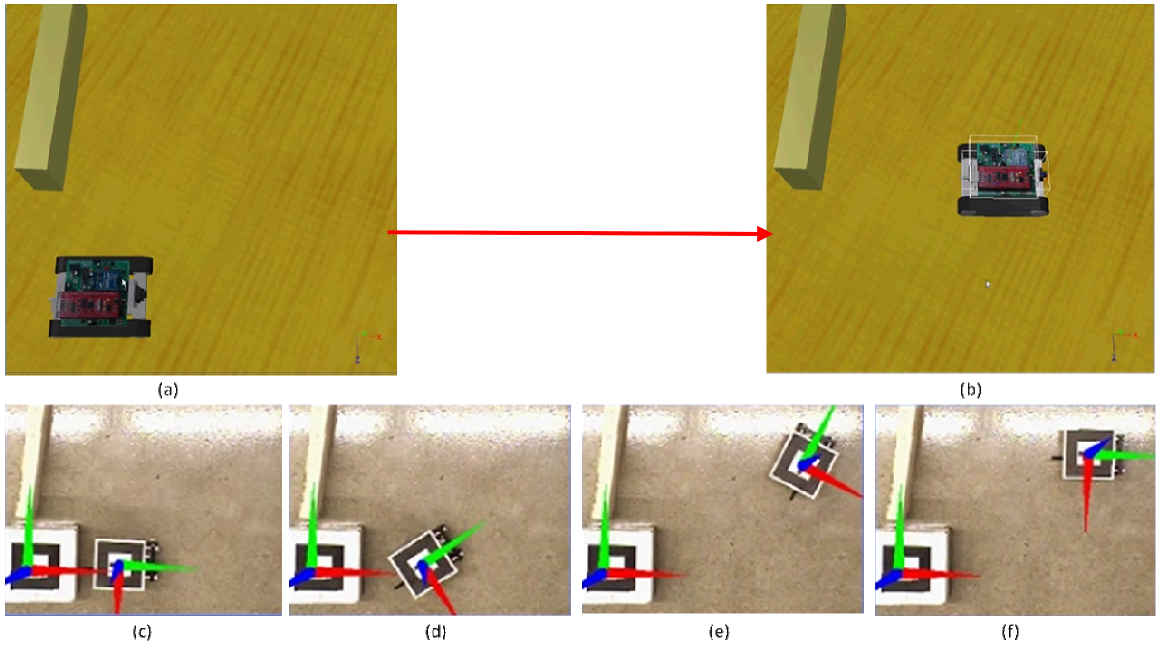


Figure 5: In a VR interface, the user interacts with the real robots through the computer generated overlays in a 3D environment. In (a) the user views the initial position. In (b) the user moves the robot to the goal position and orientation. In (c) the robot receives the command and in (d) orients to the goal position and (e) moves to that position. Finally, in (f) the robot reorients to the final goal orientation.

3.3.2 Guarded Tele-operation

Guarded tele-operation can be used to prevent operators from inadvertently driving mobile robots into walls and other objects. A simulation of ODIS robot within Webots™ is created to assist in guarded tele-operation (see Figure 6). Each robot created in Webots™ simulation was associated with motors and sensors to emulate the real robot. Software was written such that the system could send the same motor control commands to both the real robot and the virtual robot. In this system, the virtual robot was sent the commands to move first. If the virtual robot (via the physics simulation of Webots™ environment) was able to move to the goal location, the commands would be sent to the actual robot. If the robot was unable to move due to some physical obstacle or a virtual sensor reading that indicated an obstacle, the motion command would not be sent to the real robot. In this way, virtual tele-operation was achieved and the actual robot would stay within the boundaries setup by the virtual environment. In the virtual environment, the wireless communications were expanded to include TCP/IP sockets. This allowed the system software to tap into the ODIS communication stream and to execute the same commands in the simulation that the real ODIS is executing. This feature allows commands to be sent to both the actual and virtual ODIS. A tele-operator can control the physical and virtual robots simultaneously. As illustrated in Figure 7, a joystick is used to control both robots. However, the real robots use the virtual sensors on the virtual robot to avoid collisions by removing the portion of any movement that would cause a collision. In this way, changeable virtual walls and objects along with virtual sensors can be used to control and constrain the behavior of the actual robots for simulation and training.

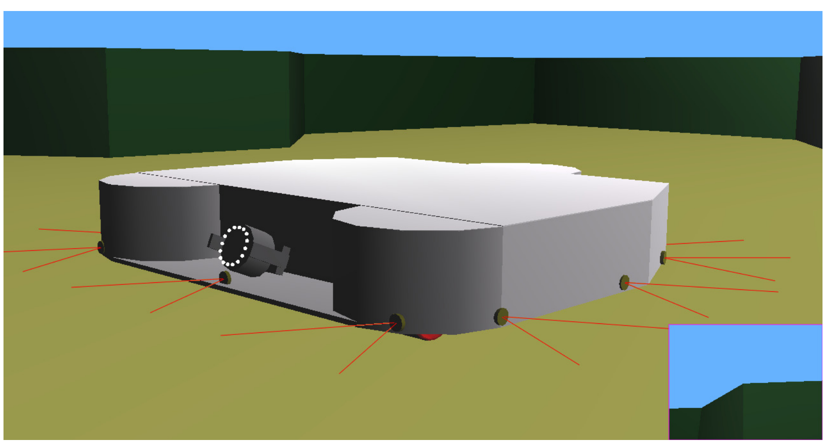


Figure 6: A virtual 3D model of ODIS robot and virtual sensors within Webots™ is created.

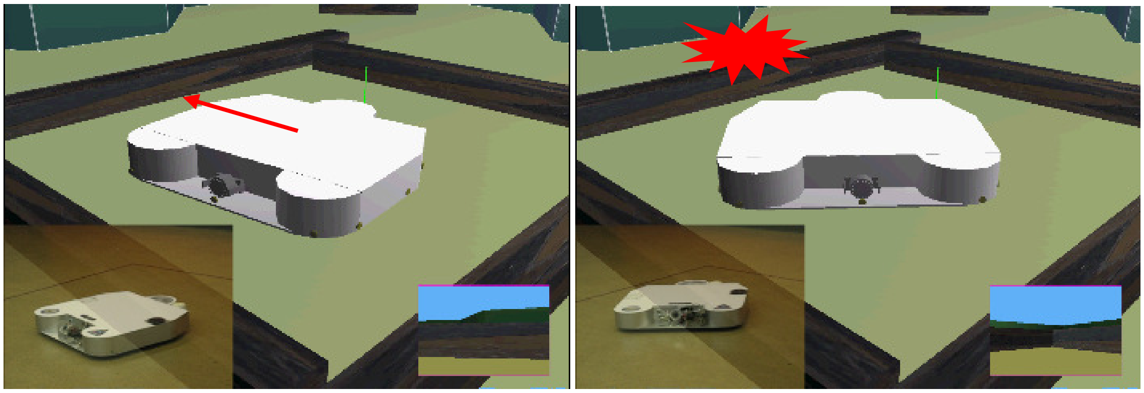


Figure 7: This illustration shows robotic guarded tele-operation over a virtual environment. The left side shows the robot was moved toward to the virtual boundaries. The right side shows the virtual robot detected the virtual boundaries and the actual robot would stay within the boundaries setup by the virtual environment.

3.4 Camera Control and Tracking

One of the issues with remote operations is that the remote vehicles as seen from a camera view need to be in constant view in the field of view of the camera. The operator must not only navigate the ground robots, but, also orient the surveillance or over watch camera. The main goal of this area was to be able to automatically track ground robots to reduce operator workload. The implementation of the ground robot tracking system was based on tracking augmented reality markers [87] using the SONY Pan/ Tilt/ Zoom (PTZ) camera to determine the robot's position and orientation with respect to the camera. The camera has 18x optical zoom and is capable of panning and tilting. It outputs image data using NTSC. It also has excellent low-light sensitivity. The Sony camera runs Sony's proprietary VISCA protocol, which is a packet-based protocol for handling internal camera control and pan/tilt functions. The pan-tilt camera is attached to a tripod (see Figure 8) or the ODIS extendable mast for providing a top-down view for multi-robot cooperative control.

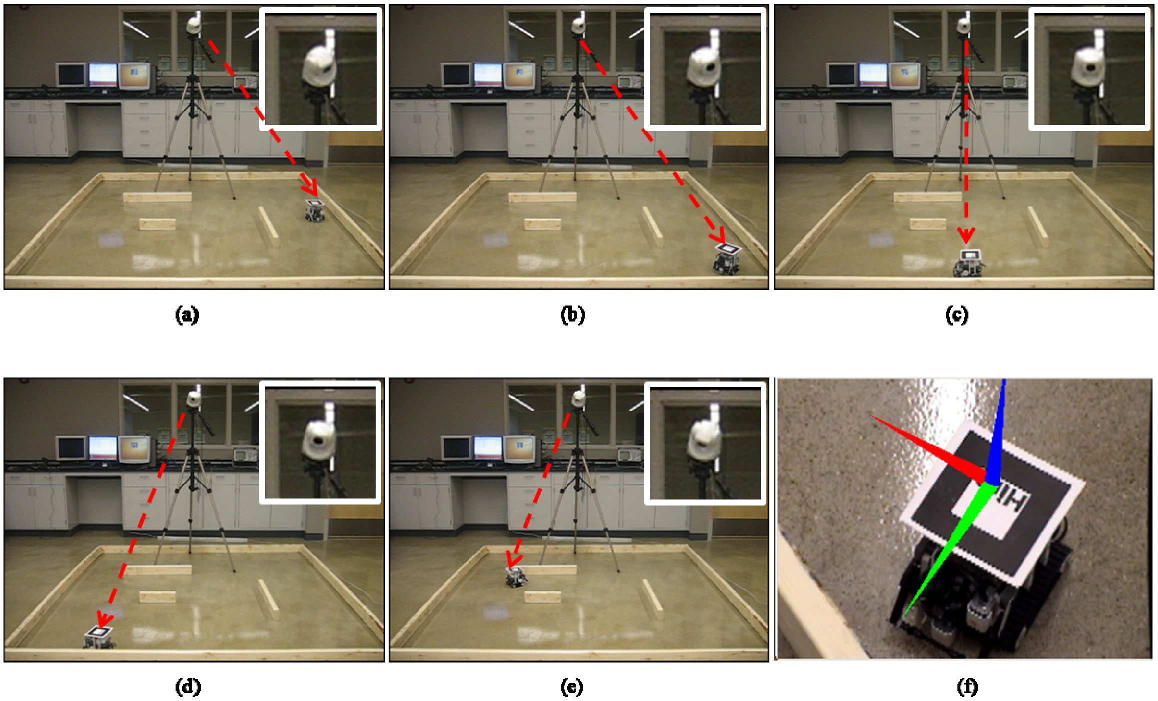


Figure 8: The camera automatically follows the ground robot, (a)-(e) show how the camera uses pan-tilt commands automatically to keep the ground robot in the center of its field of view. (f): the video feed from the pan-tilt camera shows the ground robot in the center of the camera field of view.

The marker is attached on the top of the ground robot. The marker tracking is implemented with Augmented Reality system. An Application Program Interface (API) was created for controlling the serial interface with the PTZ camera. The API is able to control the pan angle form +170º to -170º, and the tilt angle from -30º to +90º.

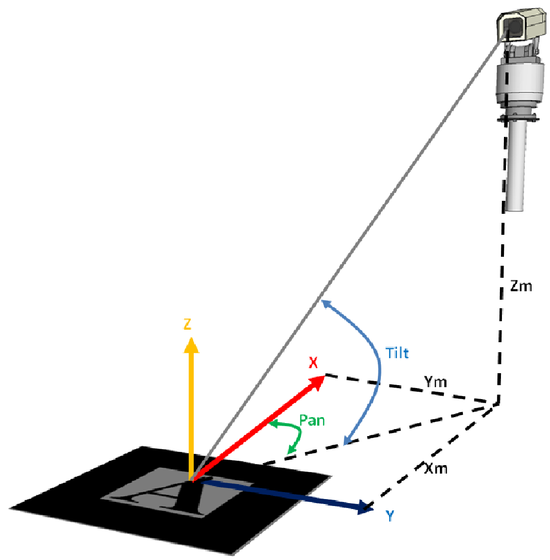


Figure 9: The illustration of the PTZ camera used in the image guidance and tracking. The corresponding angles of pan and tilt for the camera can then be calculated in real time.

The transformation between the marker and the camera can be determined by the Augmented Reality system. In Figure 9, the corresponding angles of pan and tilt for the camera can then be calculated in real time from the transformation by the equations below.

$$Pan = \tan^{-1}\left(\frac{X_m}{Y_m}\right) \qquad \text{(Equation 1)}$$

$$Tilt = \tan^{-1}\left(\frac{Z_m}{\sqrt{{X_m}^2+{Y_m}^2}}\right) \qquad \text{(Equation 2)}$$

3.5 Discussion

A Virtual Reality interface is a modality in which the operator is able to easily point to and move a simulation vehicle to achieve the desired position. The virtual environment can show what the system is doing from an arbitrary viewpoint or even from multiple viewpoints. The user is able to see all of the robots in action in the interface from many angles. Another potential use of the simulation is as a virtual world in which to plan and test maneuvers prior to executing them on the robot. Guarded tele-operation using the virtual environment can be used to prevent operators from inadvertently driving mobile robots into walls and other objects.

However, dynamic situations not modeled in Virtual Reality (VR) could pose significant problems such as collisions with dynamic objects such as other robots moving the scene. Also, due to errors in modeling the real world accurately in the virtual world, there exists difference between a robot's actual position, direction in real world and its desired position and direction in virtual world.

In a purely virtual environment, the operator's attention is drawn away from the physical environment which reduces situational awareness. An alternative to the virtual reality interfaces is an amalgamation of the benefits of physical and virtual reality. This is the topic of Augmented Reality: an advanced visualization technology that allows computer generated virtual images to merge with video views of physical objects in real time.

Chapter 4: Augmented Reality Human Multi-robot Interface

4.1 Introduction

The technical aim of the thesis is to create a test bed Augmented Reality system to demonstrate a novel multi-robot interface that has the potential to improve performance and logistics for practical applications. This chapter presents a way to design an AR human multi-robot system, including multi-robot coordinates of the system. It also provides a detailed description of the control algorithms that can be used to operate multi-robot effectively. An extended mast was developed for the Omni-Directional Inspection System (ODIS) robot which can provide an aerial view to control team of robots.

4.2 Development of Human Multi-robot Interface

The test bed developed in this work is an AR interface for human to ground robot coordination. The system combines an Augmented Reality and certain robot control algorithms to create an interface that allows human supervisor to control multiple robots. The role of this human multi-robot interface is to allow an operator to control groups of heterogeneous robots in real time in a collaborative manner. The human multi-robot interface is an AR-enhanced top-down view from a stationary camera. We assumed that the top-down view can be taken by any number of methods: manned robots, unmanned aerial robots, satellites, fixed cameras, etc.

4.2.1 Hardware

A primary goal of this research is to allow an operator to control a team of robots for contaminant localization tasks. To enable the demonstration of this capability, four

Mindstorms® NXT robots (see Figure 10) were used as the remote robots. The NXT robot built here including two NXT motors with encoders used for differential drive and a passive caster wheel in the rear to provide stability. A marker on top of a NXT robot (see Figure 10) is used by the camera subsystem to capture the position and orientation of each of the robots. This position information is also used for viewing robot status and sensor data directly on the video view. The NXT robots are controlled through a Bluetooth connection. An infrared sensor with a 240 degree view is attached on the NXT robot (see Figure 11) to search and detect an infrared beacon, which is to simulate a contaminant source. A HiTechnic infrared electronic ball was used as the infrared source. The infrared ball was hidden by one of the decoys.

Figure 10: A marker on top of a NXT robot and an infrared sensor is attached in front of the robot.

The video scene becomes the medium through which the operator directs robots which then communicate back to the user important information. The test bed was equipped with a Logitech Webcam Pro 9000 with autofocus to obtain video frames at a resolution of 1280x1024 pixels and at a refresh rate of 10 frames/ sec. The video was displayed on a 17" liquid crystal display (LCD) computer monitor. As the robots move, their position and orientation is known relative to the camera view and this allows the user to potentially control the robots and for the robots to paint information back to the user both directly on the video canvas.

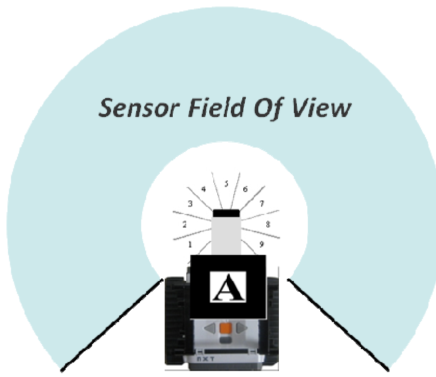


Figure 11: An infrared sensor with a 240 degree view is attached on the NXT robot to search and detect infrared beacons.

4.2.2 Software

The client-server system for data communication, illustrated in Figure 12, has been developed in the test bed. The internet protocol suite, the Transmission Control Protocol and the Internet Protocol (TCP/IP), are all used for data communication in the system. It allows computers to distribute data over a network to and from each other and the robots in the field. In addition, this architecture allows the system to distribute computing loads and data across a number of clients. For example, it allows multiple robot clients to connect to the server from a heterogeneous group of robots and communicate and display information on the same video scene.

In Figure 12, an AR server is connected to an input device (joystick), the video camera, and the display. Each robot client connects to the AR server and can receive movement commands and send sensor or state information to the server. The AR server is also connected to a pose estimation system which computes the best path for a particular robot given obstacles in its path. In the following sections, each of software subsystems will be discussed.

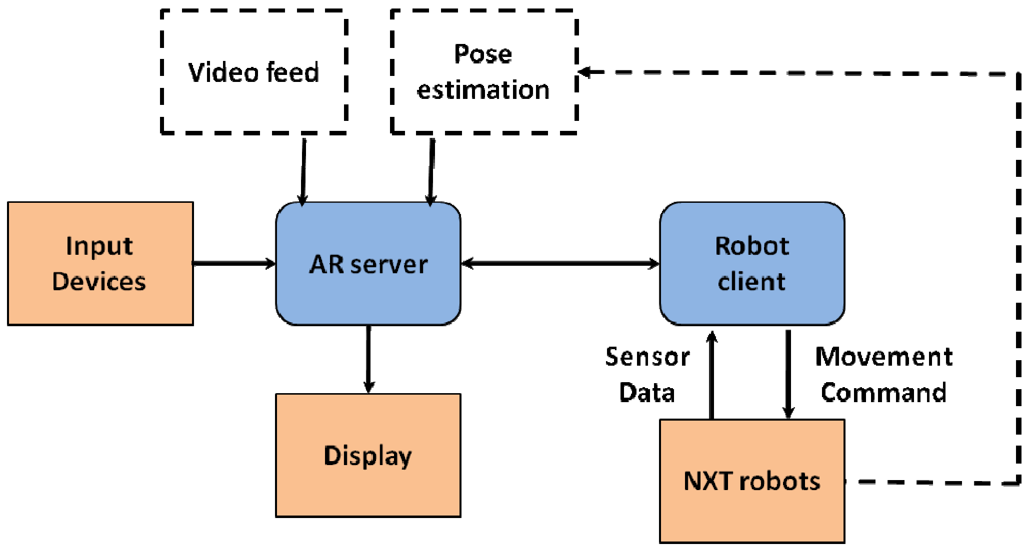


Figure 12: AR server programs share robot pose information and display the synthesized view. Robot client programs read robot sensor data and send movement commands.

AR server

The AR server is at the heart of the system. It communicates with the pose estimation system, gets input from the user and the robots and displays information back to the user via the video display. C++ is the programming language used for the AR server software development. The main functions of the AR server programs are to compute the transformations required to estimate the robots' pose in the camera and render graphics using these transforms. A software library for building a marker-based Augmented Reality applications used in the AR server is an open source ARToolKit library [87]. It can determine the position and orientation of a pre-defined set of markers. This library will essentially read a video feed, look for a particular pattern that it is pretrained to recognize and using the intrinsic parameters of the camera system, compute the pose (both position and orientation) of the marker.

Robot client

The robot client communicates the desired motion for each of the robots and sends the commands. It is programmed to communication with NXT robots using Bluetooth. C++ is the programming language used for the robot communication API software development. The basic control loop for the robot client is shown in Figure 13.

Not eXactly C (NXC) is the programming language used for NXT robot to configure the infrared sensor and robot communication. In order to command a robot to a particular location, the system must be able to know where each robot and the potential obstacles are in the world coordinate system. The next section describes the transformations needed to perform this task.

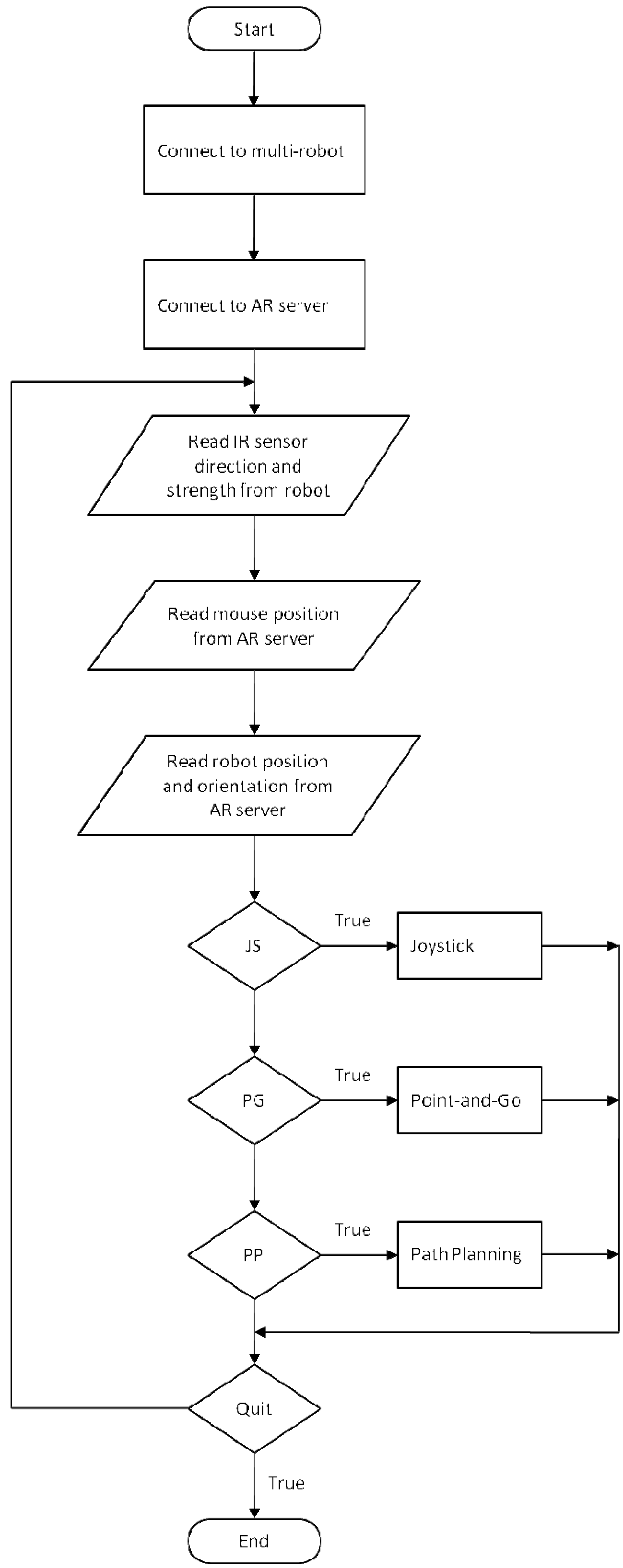


Figure 13: The basic control loop for robot client.

4.2.3 Position and Orientation of the Robot Platform

To obtain the robots' position and orientation related to the image window, the solution can be computed from the three coordinate frames, shown in Figure 14. These are the marker or world coordinates (X_R, Y_R, Z_R), the camera coordinates (X_C, Y_C, Z_C), and the image pixel coordinates (X_p, Y_p). For example, the transformation matrix from the robot A's marker coordinates to the camera coordinates represented in Equation 3.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & R_{02} & T_1 \\ R_{11} & R_{12} & R_{13} & T_2 \\ R_{21} & R_{22} & R_{23} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{R\_a} \\ Y_{R\_a} \\ Z_{R\_a} \\ 1 \end{bmatrix} = T_{C-a} \begin{bmatrix} X_{R\_a} \\ Y_{R\_a} \\ Z_{R\_a} \\ 1 \end{bmatrix} \qquad \text{(Equation 3)}$$

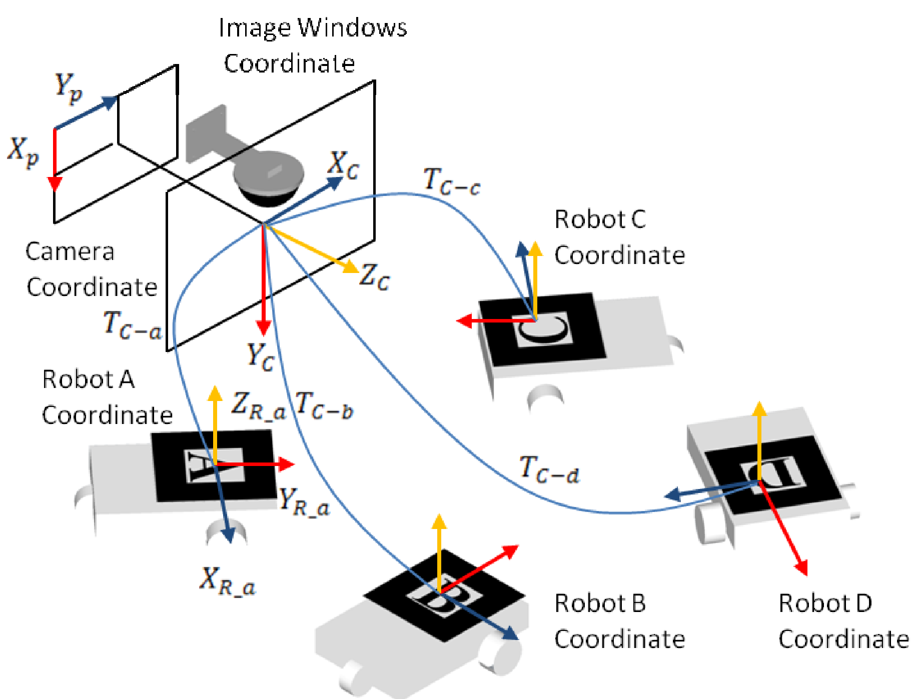


Figure 14: Geometric transformations between different coordinate systems were used to estimate the pose of each mobile robot in the camera.

Referring to camera calibration model [88], a projective mapping from camera coordinates to the image pixel coordinates is denoted using the camera matrix in Equation 4. These parameters in the matrix encompass the focal length in terms of pixels (w_x, w_y), the skew parameter (s) which is the angle between the x and y pixel axes, and the principal point (u_0, v_0). Each robot's X and Y position are able to be converted to the image pixel coordinates represented in Equation 5.

$$\begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix} = \begin{bmatrix} w_x & s & u_0 & 0 \\ 0 & w_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} \qquad \text{(Equation 4)}$$

$$\begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix} = \begin{bmatrix} w_x & \Upsilon & u_0 & 0 \\ 0 & w_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{00} & R_{01} & R_{02} & T_1 \\ R_{11} & R_{12} & R_{13} & T_2 \\ R_{21} & R_{22} & R_{23} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{R\_a} \\ Y_{R\_a} \\ Z_{R\_a} \\ 1 \end{bmatrix} \qquad \text{(Equation 5)}$$

The T_{C-a}, T_{C-b}, T_{C-c} and T_{C-d} are the transformations of the robot A, B, C, D in the camera coordinates respectively. In Equation 6, the inverse of T_{C-a} multiplied by T_{C-b} gives the transformation of robot B in robot A coordinates. This is how a robot is related to the other robot by describing the rotations and the translations needed to transform one robot coordinate to another.

$$T_{a-b} = \mathrm{T}_{\mathrm{C-a}}^{-1} \times T_{C-b} \qquad \text{(Equation 6)}$$

In Equation 3, R is the rotation matrix (3 × 3) for Euler angles and T is the vector (3 × 1) representing the translation matrix as shown in Figure 15. T_1, T_2, and T_3 represent the corresponding translation in X, Y, and Z directions. The rotation parameters R_{00} and R_{01} in the rotation matrix are used to compute the orientation of each ground robot, since the robots only rotate in the Z direction. The observation model and process used for computing the robot's orientation are illustrated in Figure 16 and Equation 7. α is the robot's oreitation measured from Y direction.

$$\text{Rotation} \longrightarrow \begin{bmatrix} R_{00} & R_{01} & R_{02} & T_1 \\ R_{11} & R_{12} & R_{13} & T_2 \\ R_{21} & R_{22} & R_{23} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \longleftarrow \text{Translation}$$

Figure 15: ARToolKit give us the transformation data in Rows

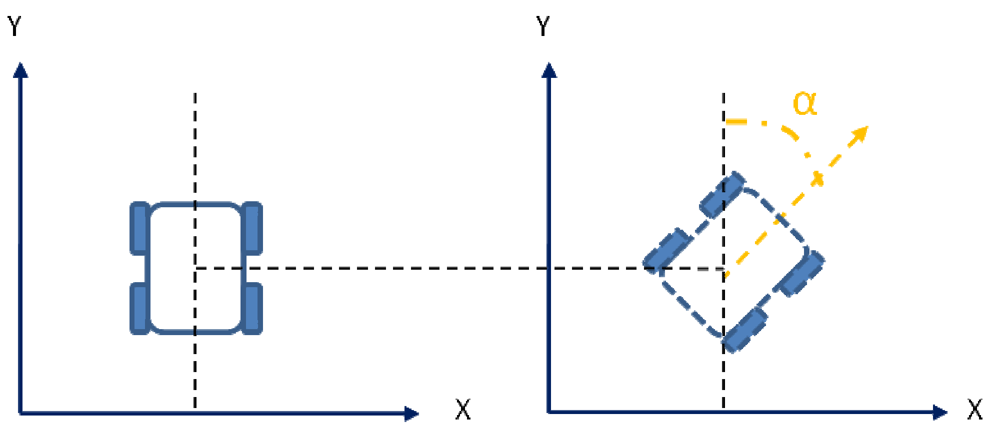


Figure 16: Illustration of the orientation of a ground robot

$$\begin{cases} \alpha = cos^{-1}(R_{00}) \times 180 \div \pi, & if\ R_{01} > 0 \\ \alpha = 360 - (cos^{-1}(R_{00}) \times 180 \div \pi), & if\ R_{01} \leq 0 \end{cases} \quad \text{(Equation 7)}$$

4.2.4 Visualization of Sensor Data

The AR interface displays sensor information from each robot in real-time and drop color-code arrows on the robot's path to generate a sensor data map. It allows the operator to use fused sensor information to augment decision making in order to direct multiple ground robots towards a source. The capabilities allow the robots to localize multiple sources simultaneously.

The infrared sensor used in the NXT robot is able to detect infrared light sources and determine their direction and approximate strength. The sensor has five infrared sensor elements arrayed at 60 degree intervals. A total of 240 degree view is configured to 9 directions of infrared signal by programming in the NXT robot using NXC. The NXC code allows the NXT robot to send the direction and distance data to the robot client in real time, and then the robot client sends the sensor data to the AR server to display this information on the video view of the interface.

The sensor data is programmed to display in a virtual image of an arrow on top of the robot using the robot's position and orientation information described in the previous section. The direction of the arrow points to the approximate source location, and the color of arrow represent distance value based on a color scale as shown in Figure 17. A set of the arrows are plotted along on the robot's path when the robot traverses toward a source target.

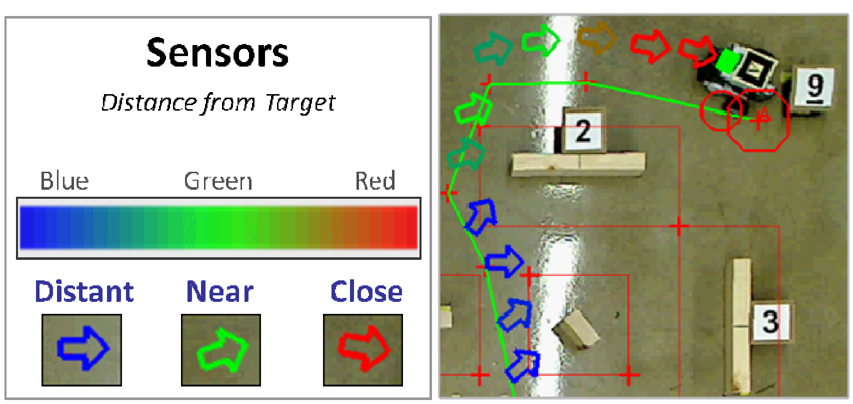


Figure 17: The robots detect infrared signals and drops color-code arrows when in motion. Arrows indicate the direction and distance from a sensor to a source.

4.3 Control Method and Behaviors

4.3.1 Point-and-Go Interface Design

A point-and-go algorithm was developed for a single human operator controlling multiple robots. The operator is able to choose any ground robot using a mouse left click, and then designates a goal location on the interface (see Figure 18 left). A navigation algorithm is developed that allows the robot to turn toward the desired goal location, drive straight toward the goal, and then stops at the target. If a robot is stuck at an obstacle, the user is able to reverse the robot using mouse right click (see Figure 18 right). Because the interface has tracked markers and an global marker, the system knows the locations of the robots and also the location or the user selected points in the video scene all in the same reference frame. This allows for the computation of simple behaviors like the point and go behavior.

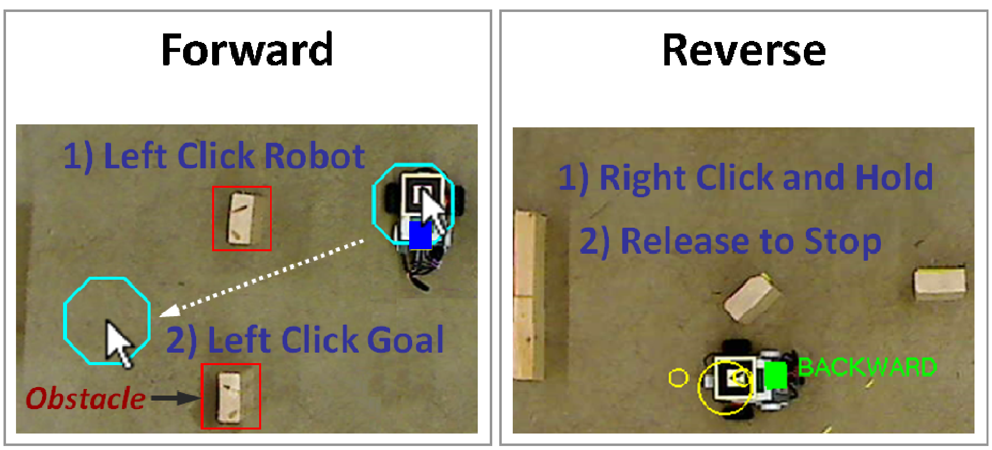


Figure 18: Point-and-Go is a high level instruction that allows an operator to control multiple semi-autonomous robots simultaneously

Figure 19 illustrated how a robot rotates toward the desired goal location. (R_x, R_y) is the robot's current position, and (M_x, M_y) is the goal position. The orientation of the robot is able to be computed by the pseudo-code in the Figure 20.

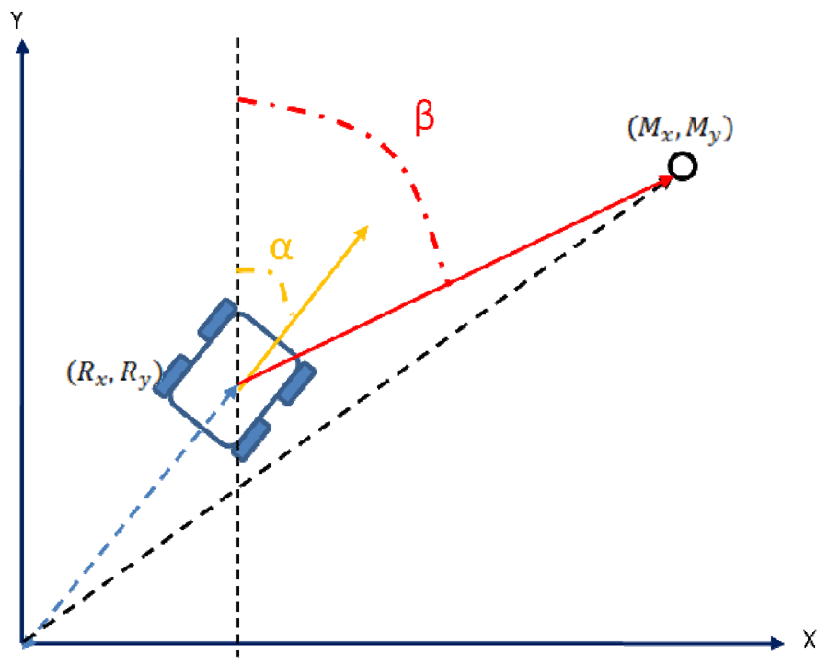


Figure 19: Illustration of the robot as it turns toward the desired goal location

Algorithm: The orientation of desired goal location

$V_x \leftarrow M_x - R_x$
$V_y \leftarrow M_y - R_y$
if $V_x > 0$ **then**
$\beta = \tan^{-1}\left(\frac{V_y}{V_x}\right) \times \frac{180}{\pi} + 90$
else
$\beta = \tan^{-1}\left(\frac{V_y}{V_x}\right) \times \frac{180}{\pi} + 270$
endif

Figure 20: Pseudo-code for the orientation of desired goal location

```
Algorithm: Point-and-Go
 if target point is clicked & robot is ready to move forward to the target position then
    if robot is at the target position then
       robot stop
    else
       if ( α < β ) then
          if (β - α) > 180 then
             robot rotate left
          else
             robot rotate right
          endif
       else
          if ( α – β ) > 180 then
             robot rotate right
          else
             robot rotate left
          endif
       endif
       if α = β then
          robot  move straight forward
       endif
    endif
endif
```

Figure 21: Pseudo-code for Point-and-Go algorithm

4.3.2 Path Planning Interface Design

To increase the level of autonomy for obstacle avoidance, the robot must be able to find a trajectory to another position in the environment. In this work, a path planning system was built for multi-robot based on the development of a modified form of the Probabilistic RoadMap planner (PRM) [89].

The PRM is a motion planning algorithm, which is able to determine a path between a starting position of the robot and a goal position while avoiding obstacles. The basic PRM begins by taking random sample nodes from the configuration space of the robot. The colliding nodes which are within the obstacles are rejected and the remaining nodes which are in the free space are used. The starting and goal nodes are then added in.

Next, a local planner attempts to connect these nodes to each other. After this process is completed, the roadmap is created. A graph search algorithm can then be applied to the roadmap to determine a path between the starting and goal nodes. A variety of search algorithms are available to search roadmap to obtain the shortest path including Dijkstra's algorithm [90] and the A* search algorithm [91, 92]. The A* search algorithm is implemented in the system. An example of the path planning is shown in Figure 22. As illustrated, the environment is populated with random points that are obstacle free. Next a network is generated which connects each node to every other node in the scene. Then the A* algorithm is used to compute the shortest path. If the obstacles do not change, it is not necessary to redistribute and connect the node. If on the other hand, a path cannot be generated, more random points must be added to the scene and the process duplicated.

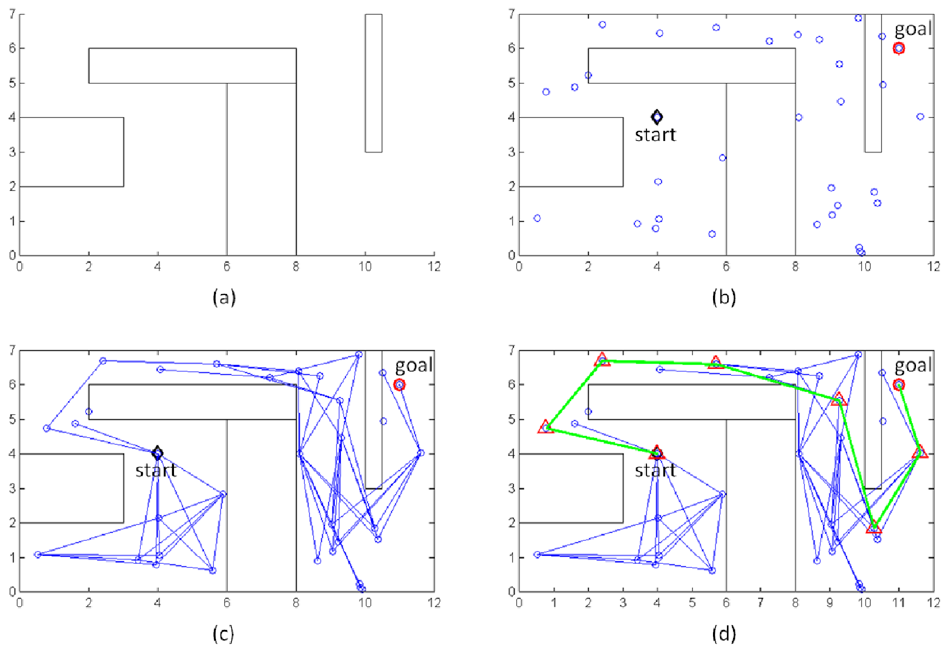


Figure 22: The figures describe the path planning used in this research.

4.3.3 Joystick Interface Design

The interface use a flight-style joystick is shown in Figure 23 (ExtremeTM 3D Pro; Logitech, California). The robots are controlled by pressing the buttons located on the top of the joystick and manipulating the directional gimbal on the joystick in the direction of the desired motion. To move a robot, the user must press the corresponding button, and then push/ pull the joystick axe to control the forward and back movements for translation of the robot, and twist left/ twist right the joystick rotation axe to control the turn left and turn right movement for rotation of the robot.

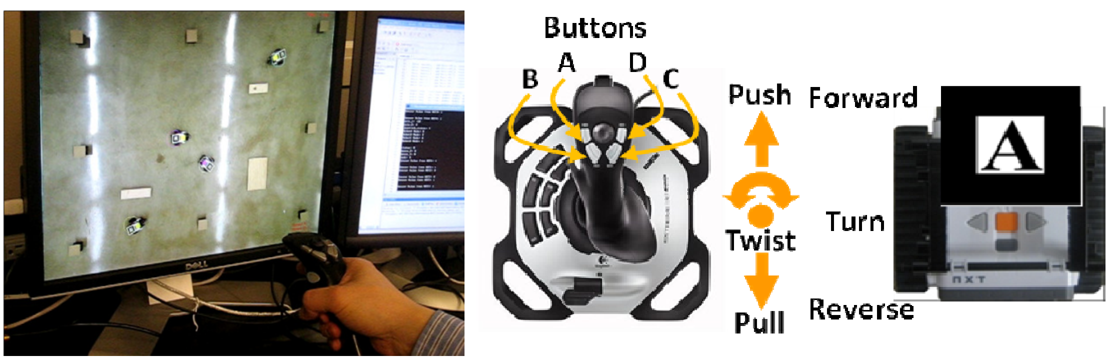


Figure 23: Interface that uses a joystick for input with Augmented Reality display (Left). Illustration shows the joystick functionality for robot control.

The joystick interface only allows the user to control one robot at a time. If the button corresponding with the robot is pressed, the robot is toggled in joystick mode, and only this robot is allowed to be moved by the user using the joystick. The user can choose between the four robots to control by pressing the corresponding button at anytime.

4.4 Extendable Arm of Ground Vehicle

An extended mast was developed for the Omni-Directional Inspection System (ODIS) robot which can provide an aerial view to control team of robots.

4.4.1 Mast Design and Operation

The prototype mast uses a telescoping pulley system on top of an unmanned ground vehicle, ODIS, and is controlled wirelessly. The mast and payload reaches up to eight feet from the platform with a gripper that can pick up objects. The platform has an operator using a remote-control device to move the arm and the robot. It is equipped with a pulley system that can also be used to extend a camera for providing an aerial view.

The design of the mast is shown in Figure 24. A window-lift system powers the movement of the pulley system. This gives the mast the capability of holding a payload steady at any height from about 1- 8 feet. This mast is capable of folding down; it is operated by a 12V DC motor with a worm gear.

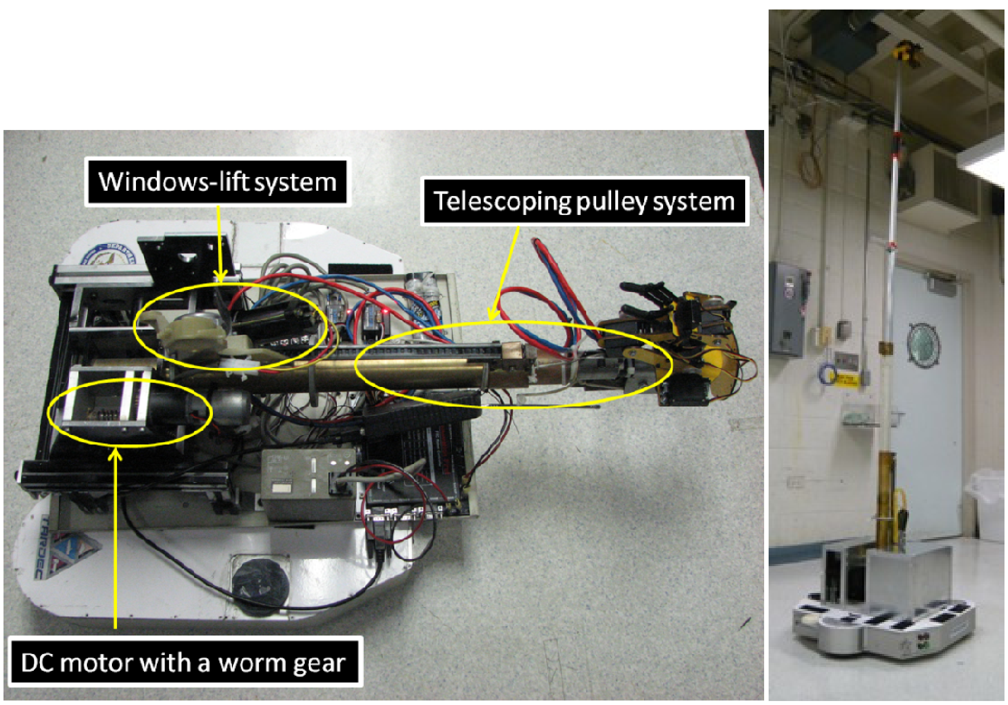


Figure 24: The mast is composed of a window-lift system to power a telescoping pulley system, and a worm gear. This system illustrates another way of getting aerial views to control multiple robots. A robot with a telescoping camera setup could be used to control the movements of other robots.

4.4.2 Simulation and VR Interface for Robot Control

In order to control multiple robots and have the flexibility for multiple sensor inputs, custom and software was developed. For instance, a simulation of ODIS within Webots™ has been created. The model is a physically-based representation of ODIS with mass inertia and friction inputs. Having a virtual environment for development and testing is advantageous for simulation of various inspection tasks. The virtual reality interface also assists in tele-operation control in several ways. The virtual environment can show what the system is doing from an arbitrary viewpoint or even from multiple viewpoints. Without the virtual environment, the user would only see the video from the cameras onboard ODIS. Unless the cameras had a portion of the robot in view, the user would have no direct feedback on the position and orientation of the robot.

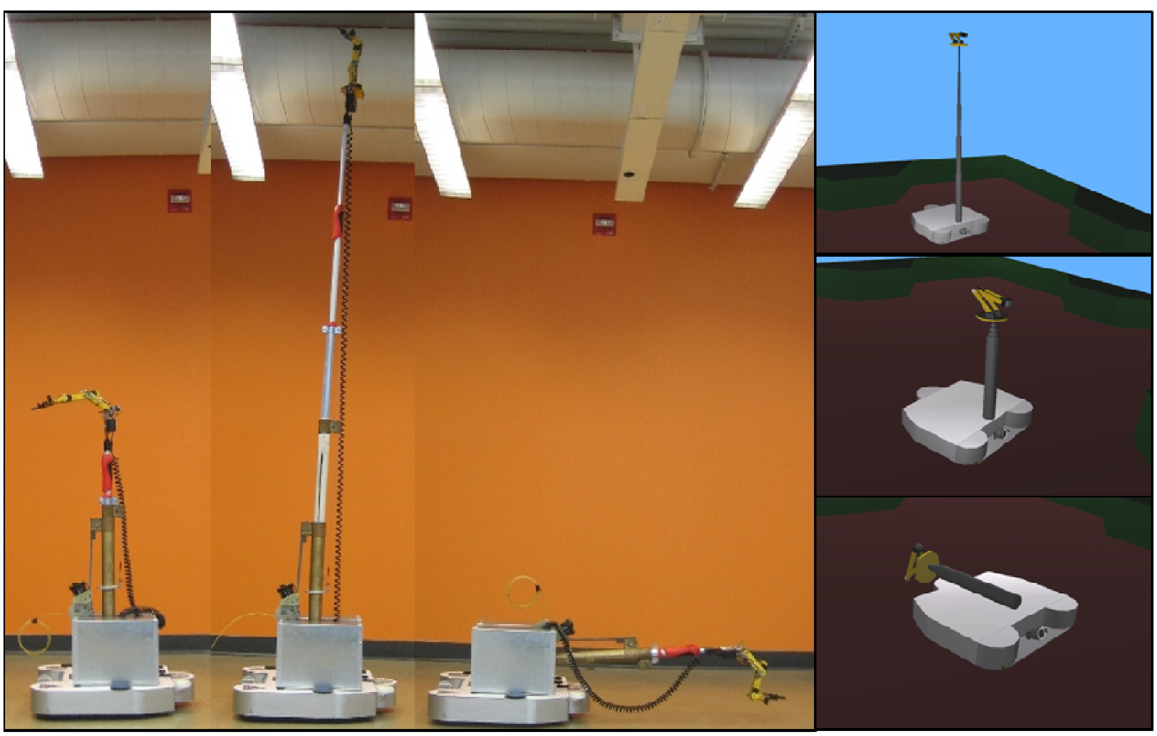


Figure 25: The extendable mast with a gripper is developed to integrate with a tele-operated ground robot. The virtual model is created in a virtual environment.

4.4.3 Design of Robot Arm Control System

A manipulator used on explosive ordnance disposal (EOD) robotic platforms includes an arm/mast and a gripper that is used for executing various tasks including securing and transporting dangerous materials. The kind of EOD manipulator or the extended mast for ODIS consists of multiple joints with a variety of motors. This work developed a flexible and cost-effective board to control a variety motors for an EOD manipulator type robot.

The robot arm control design includes hardware and software development. The hardware design created an electronic board that is targeted to control DC motors, servo motors, and stepper motors. The software designed and developed included both the firmware and communication protocol for the system. Figure 26 shows the robot arm control board block diagram.

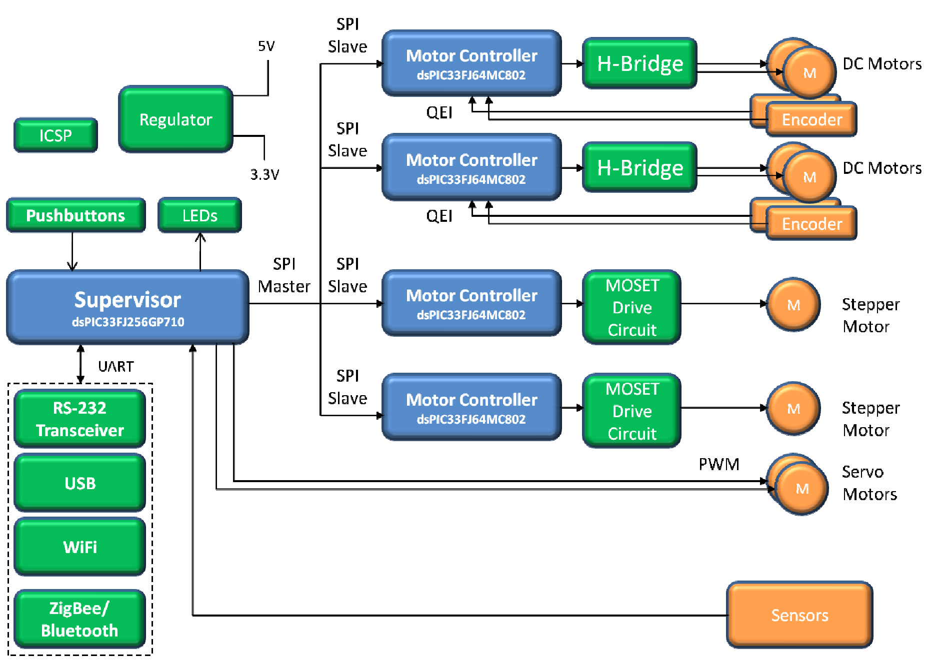


Figure 26: Illustration of the robot arm control board block diagram. The motion control board is designed to control different types of motors for robotic arms.

The board described in Figure 26 was developed such that it can be configured in different ways for use with digital signal controllers from Microchip Technology Inc. Figure 26 shows a simplified block diagram of the motor control board. Microchip's specialized motor control digital signal controller dsPIC33 device is used with the gate driver to drive the motors. Another dsPIC33 microcontroller is used to supervise the control of all motors, communicate with external hosts, and take all of the sensor readings. The board includes various circuitries to perform the following functions:

- Drive a DC motor, stepper motor, and servo motor
- Measure the feedback signals (e.g., Quadrature Encoder)
- Communicate with a host computer or an external device via USB or RS-232 interface (can be expanded to WiFi, Zigbee, and Bluetooth wireless communication)
- LED indicator for power outputs and motor status
- In-Circuit Serial Programming™ (ICSP™) connector for programming a dsPIC DSC device

The schematic and PCB layout were developed using Altium Designer. Figure 28 is the PCB showing the device layout.

The firmware is programmed in C and includes the following features:

- Pulse-width modulation (PWM) signal for motor speed control
- Quadrature Encoder Interface (QEI) for motor encoder reading
- Serial Peripheral Interface (SPI) for master dsPIC and slave dsPIC communication

- Universal Asynchronous Receiver/Transmitter (UART) for RS-232 communication

The communication protocol was also created for position and speed control.

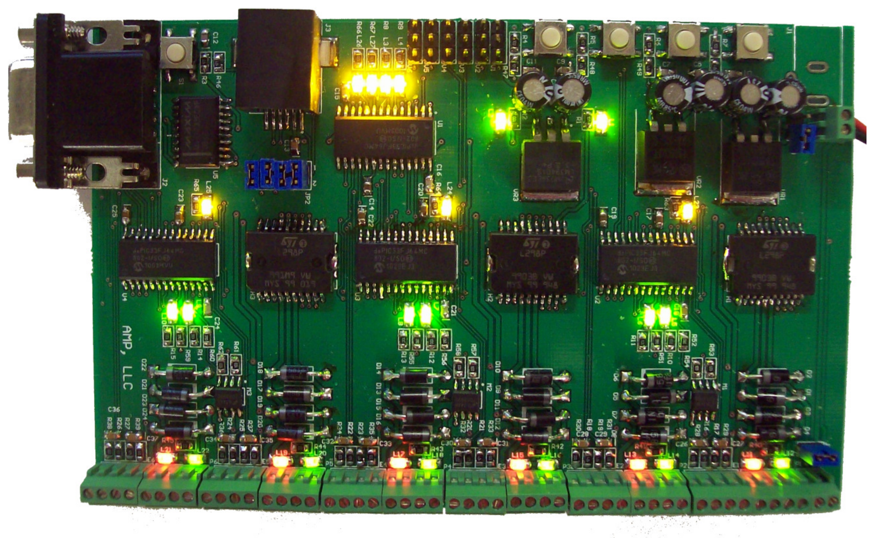


Figure 27: Screenshot of the motion controller board created for use with various robotic systems.

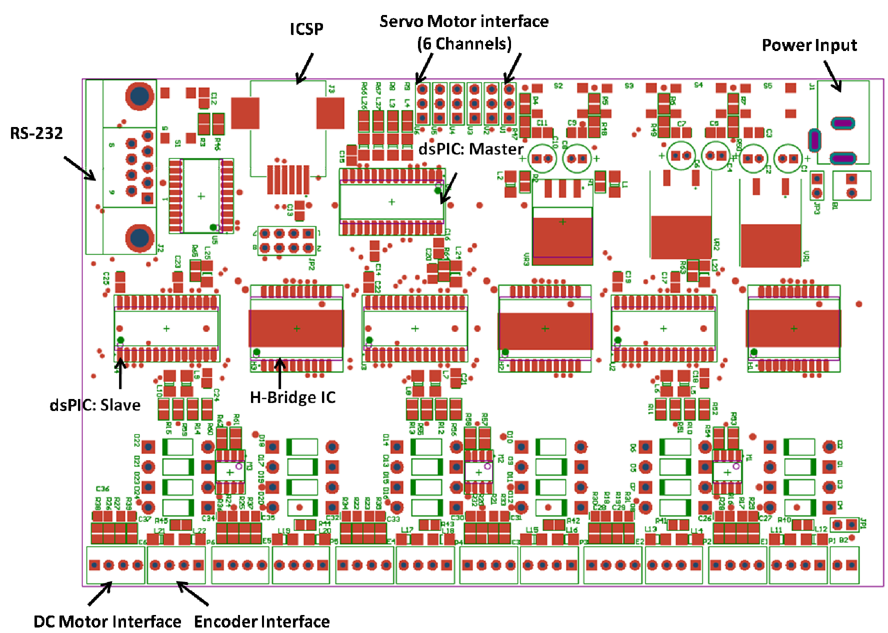


Figure 28: Screenshot of the Printed Circuit Board component layout.

This chapter described the various functionality of the AR testbed. This includes a point-and-go algorithm, a joystick interface and a path planning system which automatically computes the trajectory of the robot. In the next chapter, details of how a subject test was performed to compare these various features are described.

Chapter 5: Human Multi-Robot Interface Testing

5.1 Introduction

In any human robot technology that will be eventually used by the users, it is not enough to develop the technology. It must be validated and improved with user testing. Hence, well-developed and focused subject testing is used to quantify any improvements. This chapter describes the experiments that were performed to quantify the performance of a user using various techniques of control. Here the main three control methodologies are compared with subject testing. The methods included (1) point-and-go: where the user simply points to a robot and a goal location and the system moves the robot to this point without any regard to obstacles (2) path planning: where the user doesn't have to worry about the obstacles, the robots automatically maneuver around them. There is however some error in this computation due to the inaccuracies of computing the exact position and orientation and (3) joystick: where the user is allowed to control one robot at a time using a joystick.

5.2 Experimental Design

5.2.1 Apparatus

All trial runs were conducted on a rectangular arena which was eight feet wide and ten feet long. Eight identical numbered boxes were placed at fixed positions, two per side of the arena, equal distance from the center of the course (See Figure 29). A total of sixteen wood blocks, representing obstacles, were placed between the boxes and the center of the arena. Eight equal size barriers served as fixed points against which the boxes were positioned, impeding both the physical path and line-of-sight to each box.

For each trial, one numbered box was randomly assigned as the target and a concealed omnidirectional infrared source, representing hazards (e.g., explosive), was placed inside. The remaining seven boxes served as decoys during the trial. Sensors integrated on the robots detected the signal strength and direction to the signal origin with respect to the robot frame of reference.

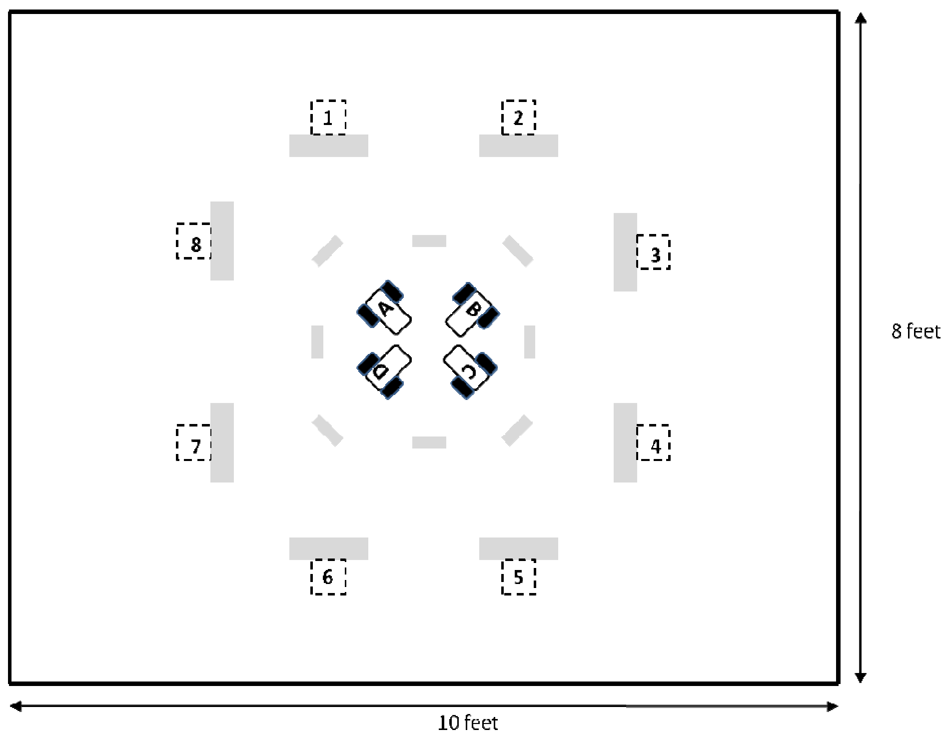


Figure 29: Illustration shows the layout of the obstacles and the decoys, as well as the initial position of the four robots used in the work.

5.2.2 Procedure

The experiment consisted of three phases: pre-experimental, condition cycle, and post-experimental. Each participant was first introduced to the test bed and briefed on the experiment. The participants have been shown what the robots looked like and the location of the overhead camera. Participants read a Research Information sheet explaining the general scope of the experiment and the voluntary nature of his or her participation. Participants filled out a pre-experiment questionnaire requesting demographic data and reporting their relevant experience with automobile driving, video game play, remote control devices, and mobile robot operation. The complete set of the pre-test questions given to each subject during the test is provided in the appendix B.

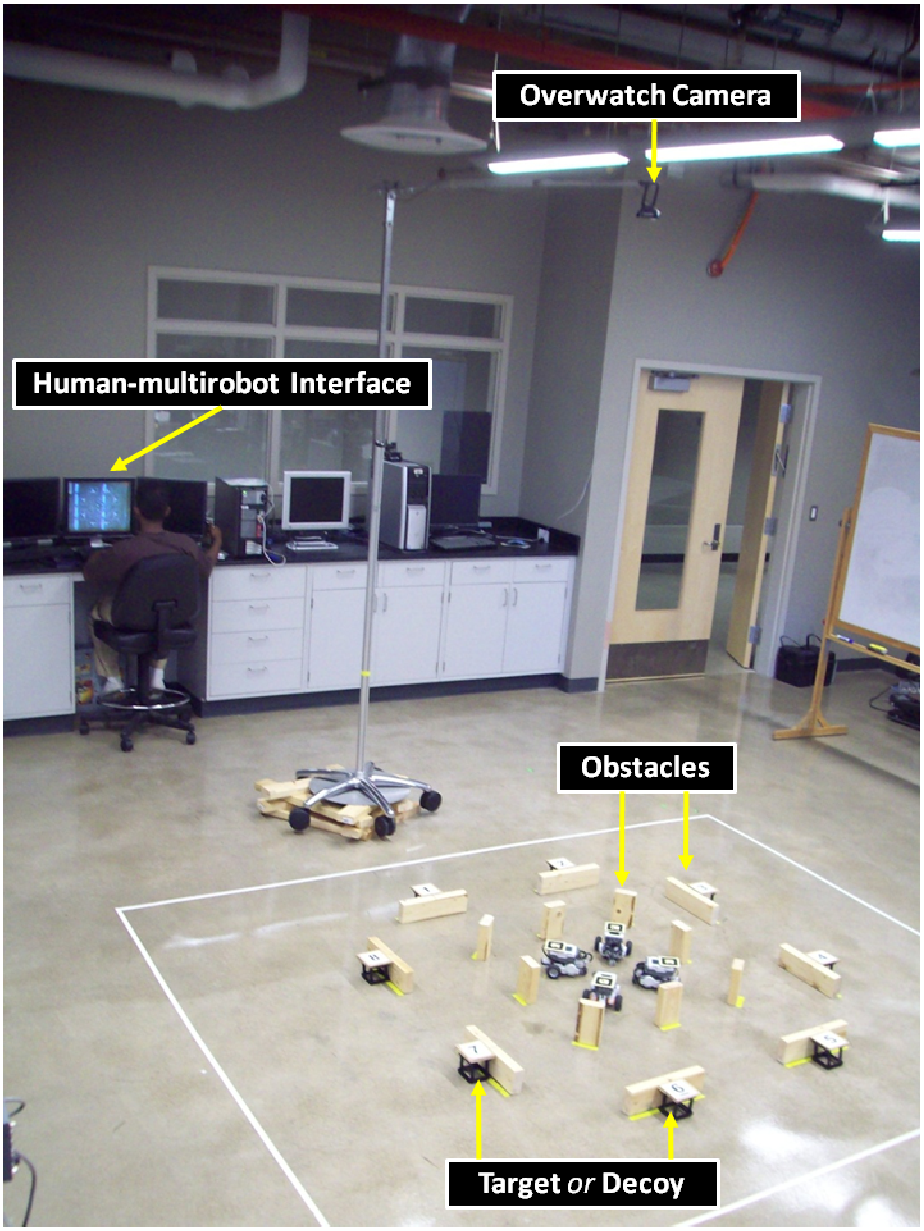


Figure 30: A group of semi-autonomous robots is controlled using the human multi-robot interface by a participant

Participants next viewed a self-paced presentation introducing the format of the experiment and summary of tasks to be performed. A self-paced presentation provided specific instruction on how to control the robots using the interface in different conditions under evaluation. The material also covered the display and interpretation of sensor and status indicator graphics. The self-paced presentation is provided in the appendix B.

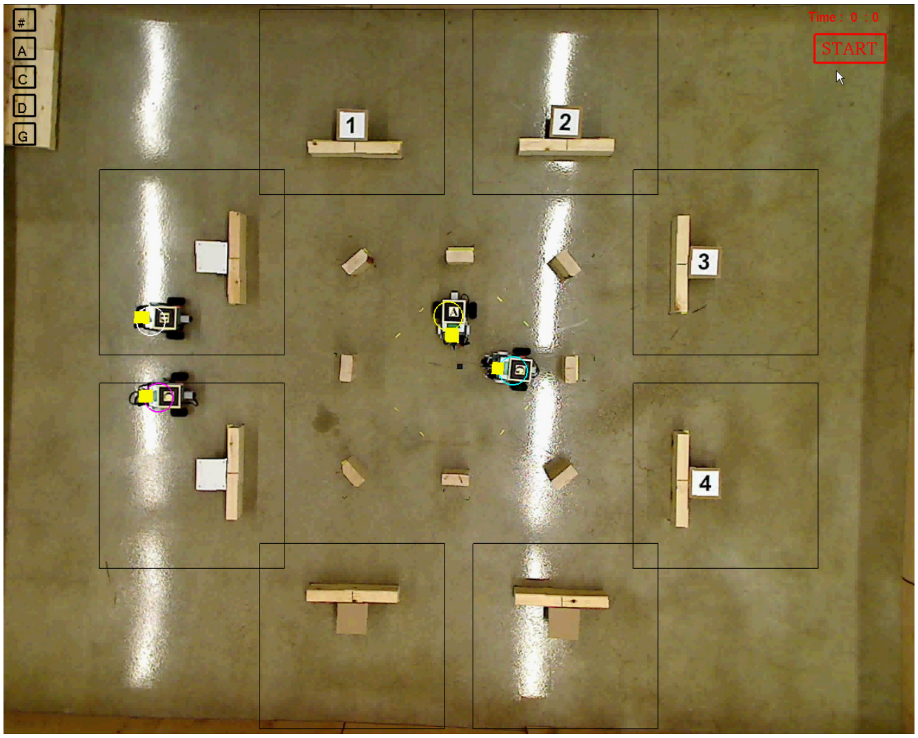


Figure 31: Two robots were placed at the center of the arena as a starting position for a practice trial.

A practice trial was conducted with a limited search task to familiarize the subject with the interface and task performance. Two robots were placed at the center of the arena as a starting position (Figure 31) and one target was randomly selected among four potential targets (decoy number one to number four). Participants were asked to maneuver the area to find the target during a timed proficiency period. Participants were required to meet the timed proficiency standard established by pilot testing. The practice scenario was repeated until proficiency was demonstrated.

After participants become comfortable with using the interface to remote control the robots and proficient with the interface under test, participants performed three evaluation trials. The four robots were moved to the starting location at the center of the arena (Figure 32) and put one target randomly selected among all of the eight potential targets. The target location number was randomly generated from Microsoft Excel so that knowledge of the target location gained from previous trial would not transfer to the current trial. At the end of each trial, the participant assessed their perceived mental workload while performing the evaluation tasks, utilizing a software implementation of NASA-TLX [93].

After completing all trials, operators answered several post-run questions related to their experience with the interface on a seven point scale to assess five usability factors: learnability, efficiency, memorability, errors, and satisfaction. A comments field at the end of each usability assessment provided an opportunity for participants to offer feedback. Lastly, participants were asked to select an interface condition preference and provide rationale for their selection. The complete set of the post-test questions given to each subject during the test is provided in the appendix B. After a break, these steps using the other interface condition were repeated.

5.2.3 Tasks

Participants were asked to complete two tasks: for all trials:

1. Locate and report position of the target (IR source)
2. Move all robots within a specified target range (Range indicated by a solid rectangular perimeter line around the identified target)

First, the participants had to use the mouse to click the AR "START" icon on the upper-right corner of the interface to begin a trial whenever they are ready to run a test, and then after three seconds countdown, the system would allows the user to command the robots. Figure 32 shows the user interface. There were three different control mechanisms that were used (as decribed) point-and-go, path planning and joystick.

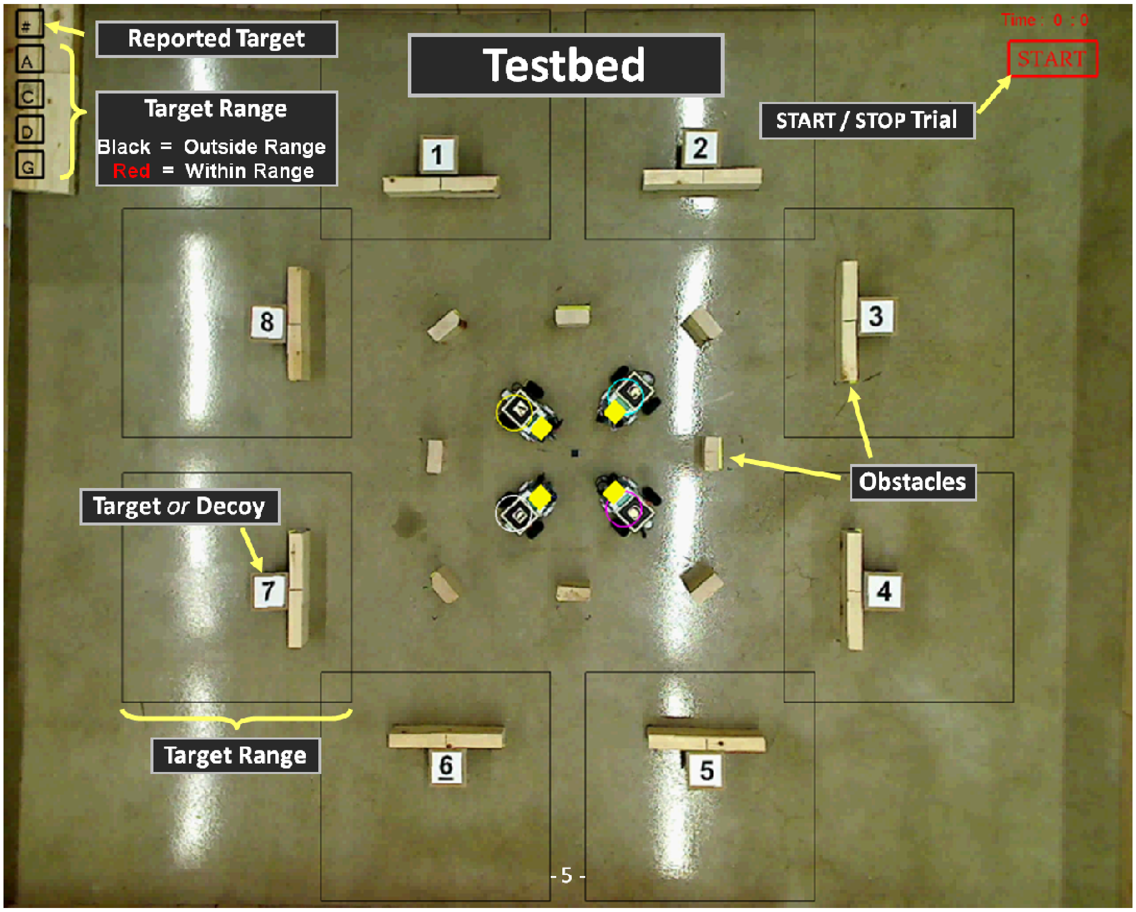


Figure 32: The human multi-robot interface is an aerial view from the stationary camera.

The "locate and report position of the target" task required the robots be commanded to search for the target. As the robots were navigated by the operator through the test bed environment, sensor information on the interface provided indications of where a randomly assigned target was positioned. Potential targets not assigned served as decoys. Participants were instructed to report the suspected target by pressing the number key on a computer keyboard corresponding to the box number of the suspected target.

The reported number of the target displayed in the Augmented Reality view on the upper-left corner of the interface, and the target range defined by a rectangular perimeter around the target will display in red color.

Upon reporting the target, the “move all robots within a specified target range” task required the user to move all robots to within the target range. Color changing display icons, one for each robot, indicated when the range task was completed. Once the correct target had been identified and all robots successfully navigated into the range, the operator have to report completion of all tasks by click the AR “STOP” icon on the upper-right corner of the interface. Figure 34 shows a task for a trial completed by a participant in Path Planning condition.

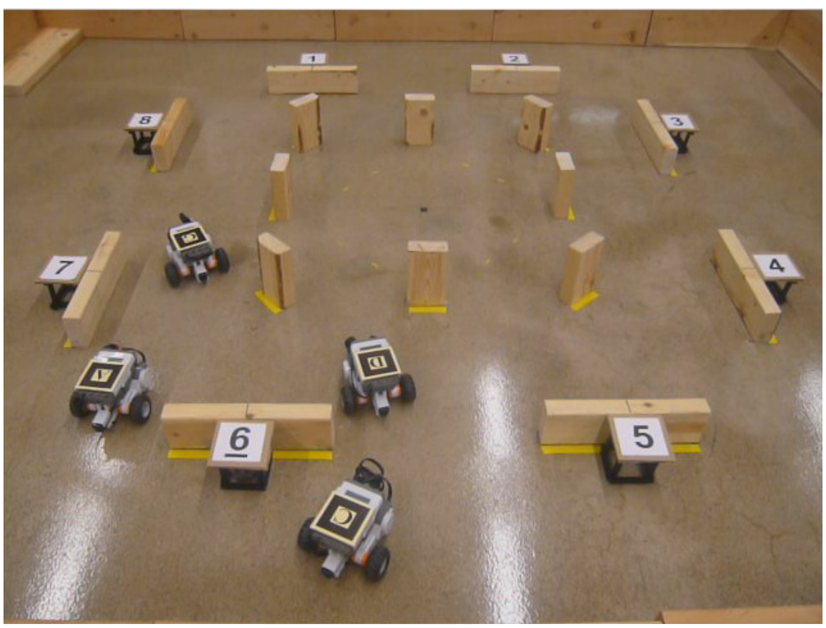


Figure 33: Four robots are moving toward the target during subject testing.

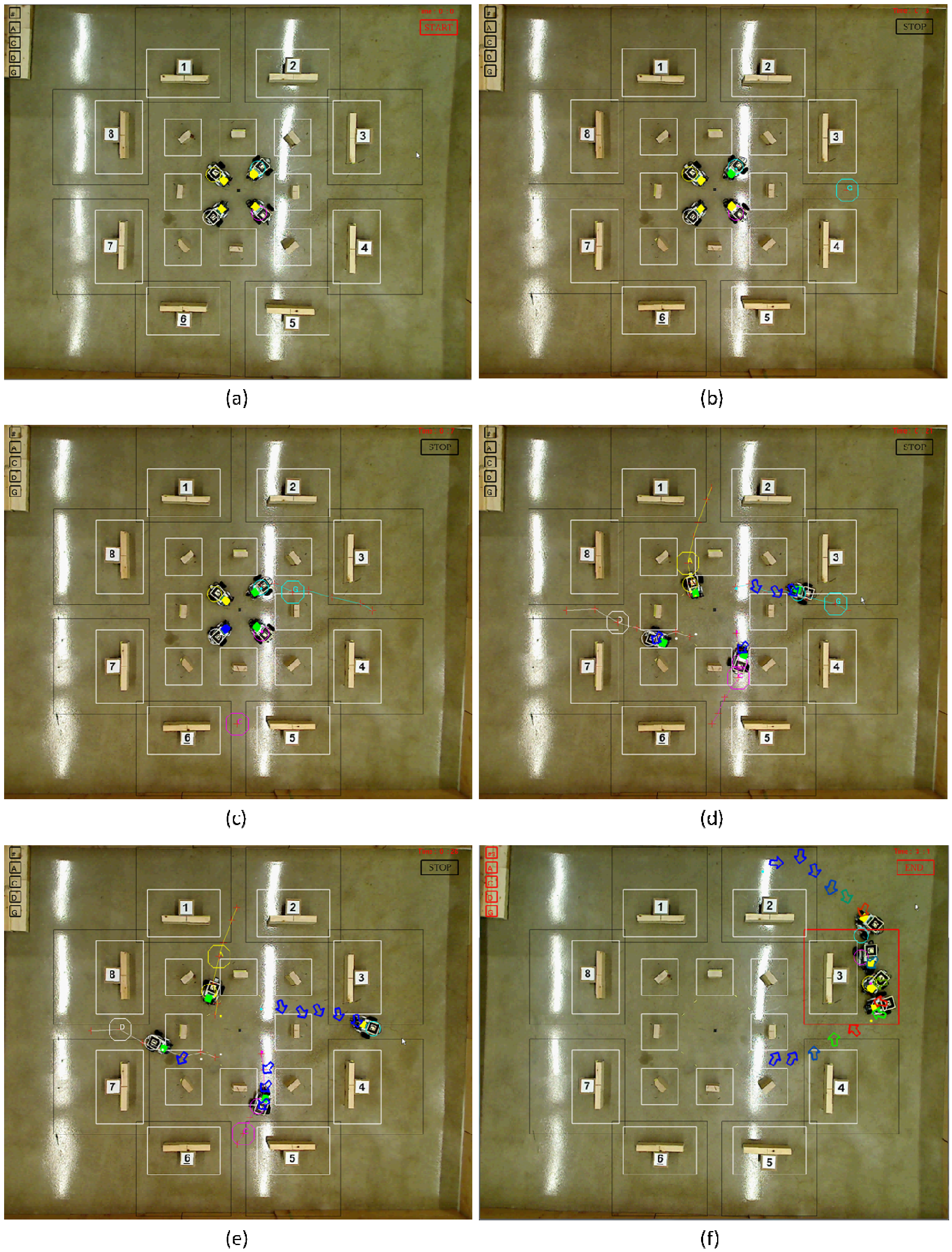


Figure 34: The no-fly/ no-go zones are showed in white lines. User report target, and then move all four robots within the target range defined by a rectangular perimeter around the target.

5.2.4 Data Collection

Three types of data were collected: logs, video, and observer notes. Custom logging software program captured each time the participants changed or activated controls. Every participant provided input to the system via the mouse or the joystick as well as the corresponding outputs of the robots consisting of robot position and orientation, robot status and time stamps were written to the logging text file during each trial. The logging text files also automatically recorded the start and end time of each trial, the task completion time, and target number reported by the participants. The video from the human multi-robot interface screen was captured during a trial.

5.2.5 Participants

Eighteen individuals that included fifteen males and three females with an average age of 23 years were selected from the student and faculty bodies of Wayne State University participated in this study. All participants were treated ethically, took part in the study voluntarily, and were assured that results would be kept anonymous and confidential.

Table 1, Table 2, and Table 3 shows the Pre-experiment questionnaire results. Five of the participants reported that they drive an automobile up to fourteen hours or more a week on average and four participants don't drive an automobile. Eleven participants played video games frequently or almost daily.

Table 1: Pre-experiment questionnaire results

Question: On average, how often do you play video games?				
Answers	Almost Never	Occasionally	Frequently	Almost Daily
Participants	3	4	9	2

Table 2: Pre-experiment questionnaire results

Question: On average, up to how many hours a week do you drive an automobile?						
Answers (Hours)	0	4	7	11	14	More
Participants	4	4	4	1	1	4

Table 3: Pre-experiment questionnaires results

Questions	Average Answer	Answers
Rate your level of experience with remote control devices (e.g. RC cars)	4.2	None 1 2 3 4 5 6 7 Extensive
Rate your level of experience controlling mobile ground robots	2.2	None 1 2 3 4 5 6 7 Extensive
Rate your level of experience controlling mobile ground robots	1.5	None 1 2 3 4 5 6 7 Extensive

5.2.6 Measurements and Data Analysis

The following dependent variables were analyzed:

Robot Switch Count: If the user selected a different robot, it was counted as one robot switch. The robot switches were summed over the entire trial.

Navigation Error Count: If the user commanded a robot to move in reverse, is was counted as one error. The errors were summed over the entire trial.

Target Identification Time: The identify source time was computed as the time that the last indication by the user of which box contains the infrared source. The identify source time was measured in seconds from the beginning of the trial.

Mission Completion Time: The complete task time was computed as the time measured in seconds from the beginning of the trial until the user clicked the stop icon to indicate that all tasks are complete.

Wait Time: The time a robot waited to be serviced after it reached its goal. The robot is idle and waiting for the operator's next command. The robot is not selected for user control, and not within the target proximity of the infrared source.

Subjective Operator Workload: The subjective perceived workload experienced and reported by the user. The perceived workload was assessed with the NASA-Task Load Index (TLX).

The NASA-TLX [94] is a self-reported questionnaire of perceived demands in six-dimensional rating method to assess subjective mental workload: mental, physical, temporal, effort (mental and physical), frustration, and performance. The NASA TLX procedure consists of two parts: ratings and weights.

Analysis of variance (ANOVA) test and t-test were used to analyze all dependent variables described above.

5.3 Results and Discussion

5.3.1 Task Performance

Robot Switch Count

The analysis of the robot switches revealed that there was a main effect of three conditions. $F(2,153) = 97.171$, $p < 0.001$, $F(crit) = 3.055$ (Figure 35). Participants changed robot selection more using Joystick than using Point-and-Go. Table 4 shows the paired t-test results for robot switch count. Participants using Joystick had less switch count compared to Path Planning. No significant differences were noticed in switch count between Point-and-Go and Path Planning.

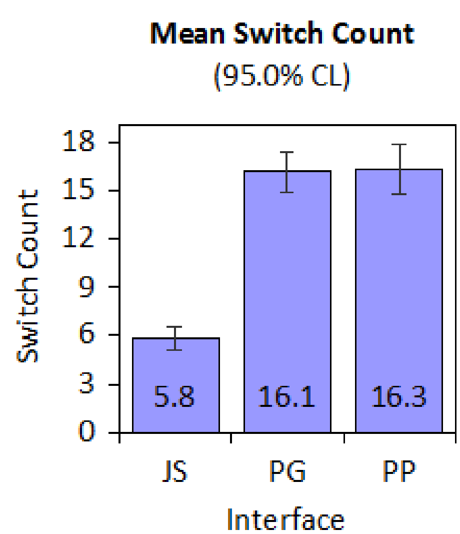


Figure 35: Robot switches are compared among participants who completed the task using Joystick (JS), Point-and-Go (PG), and Path Planning (PP). Participants using JS finished with significantly less number of robot switches.

Table 4: Participants changed robot selection more using Joystick than using Point-and-Go. Participants using Joystick had less switch count compared to Path Planning. No significant differences were noticed in switch count between Point-and-Go and Path Planning.

Average Switch Count		t(53)	two-tailed	
Joystick	Point & Go	-14.06	$p < 0.001$	✓
Joystick	Path Planning	-12.31	$p < 0.001$	✓
Point & Go	Path Planning	-0.21	$p = 0.84$	

Navigation Error Count

There was a significant difference for navigation error count between Point-and-Go (PG) and Path Planning (PP). $F(1,102) = 5.663$, $p = 0.019$, $F(crit) = 3.934$ (Figure 36). Table 5 shows the Paired t-tests were conducted on navigation error count. Navigation errors were fewer in Point-and-Go than in Path Planning.

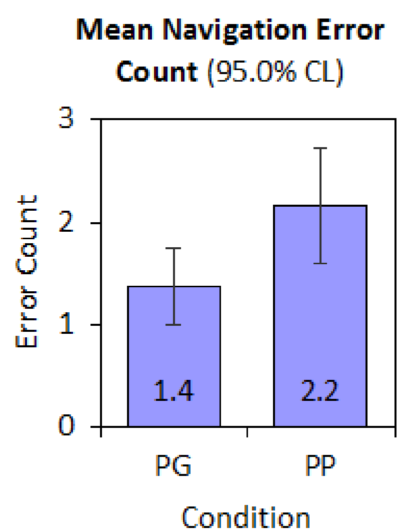


Figure 36: ANOVA was used to test differences between means for significance for navigation errors, the participants performed more reverse maneuver in Path Planning than in Point-and-Go.

Table 5: Paired t-tests were also conducted on navigation error count. Navigation errors were fewer in Point-and-Go than in Path Planning.

Average Navigation Error Count		t(53)	two-tailed	
Point & Go	Path Planning	-2.09	p = 0.042	✓

Mission Completion Time

The analysis showed that Joystick (JS), Point-and-Go (PG), and Path Planning (PP) significantly affected the mission completion time. F(2,153) = 60.272, $p < 0.001$, F(crit) = 3.055 (Figure 37). Table 6 shows the paired t-tests results for mission completion time. Participants spent more time to complete mission in Joystick than in Point-and-Go.

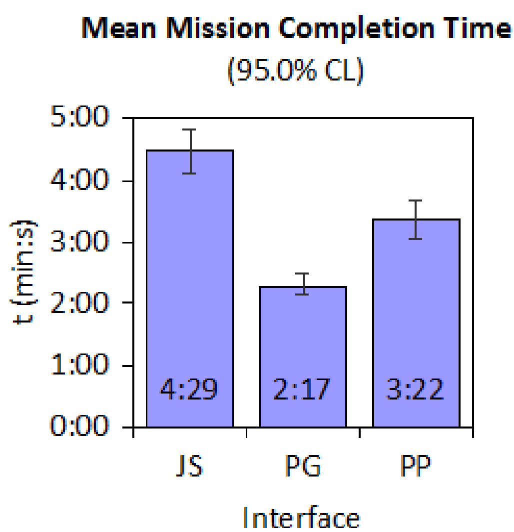


Figure 37: The analysis showed that Joystick (JS), Point-and-Go (PG), and Path Planning (PP) significantly affected the mission completion time.

Table 6: Participants spent more time to complete mission in the Joystick condition than in the Point-and-Go condition.

Average Mission Completion Time		t(53)	two-tailed	
Joystick	Point & Go	11.59	p < 0.001	✓
Joystick	Path Planning	5.26	p < 0.001	✓
Point & Go	Path Planning	-6.29	p < 0.001	✓

Target Identification Time

The ANOVA analysis showed that Joystick (JS), Point-and-Go (PG), and Path Planning (PP) significantly affected the target identification time. Statistically significant mean differences in distance were observed between the three groups. $F(2,153) = 11.165$, $p < 0.001$, $F(crit) = 3.055$ (Figure 38). Table 7 shows the paired t-tests results for the target Identification Time. Joystick and Path Planning from target identification time did not show any significant differences.

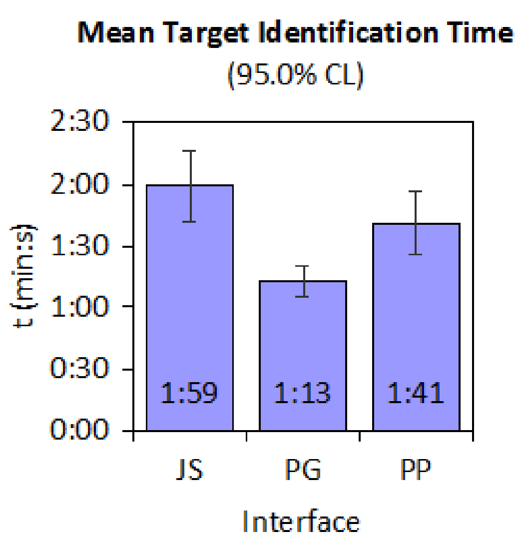


Figure 38: The ANOVA analysis showed that Joystick (JS), Point-and-Go (PG), and Path Planning (PP) significantly affected the target identification time.

Table 7: Joystick and Path Planning from target identification time did not show any significant differences.

Average Target Identification Time		t(53)	two-tailed	
Joystick	Point & Go	4.90	p < 0.001	✓
Joystick	Path Planning	1.84	p = 0.071	
Point & Go	Path Planning	-3.48	p = 0.001	✓

Wait Time

The robots' wait times are presented in Figure 39. The ANOVA analysis showed that significant differences in mean across the three groups, $F(2,153) = 139.58$, $p < 0.001$, $F(crit) = 3.06$. Table 8 shows the paired t-test results for wait time. The robots' wait time was significantly more in joystick condition than in Point-and-GO condition as well as more than in Path Planning condition.

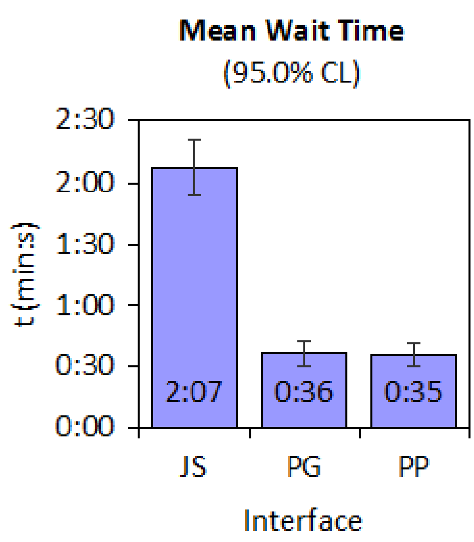


Figure 39: JS control indicated longer wait times significantly.

Table 8: No significant differences were noticed in wait time between Point-and-Go and Path Planning

Average Wait Time		t(53)	two-tailed	
Joystick	Point & Go	12.19	p < 0.001	✓
Joystick	Path Planning	13.23	p < 0.001	✓
Point & Go	Path Planning	0.10	p = 0.92	

5.3.2 Perceived Workload

Participants' NASA-TLX scores are presented in Figure 40. ANOVA results for operator workload deviation and did not show any differences in mean across the three groups. F(2,153) = 2.969, p = 0.054, F(crit) = 3.055. Table 9 shows the paired t-test results for each of the data sets. No significant differences were noticed in workload between Point-and-Go and Path Planning. Participants experienced lower workload in the Joystick condition.

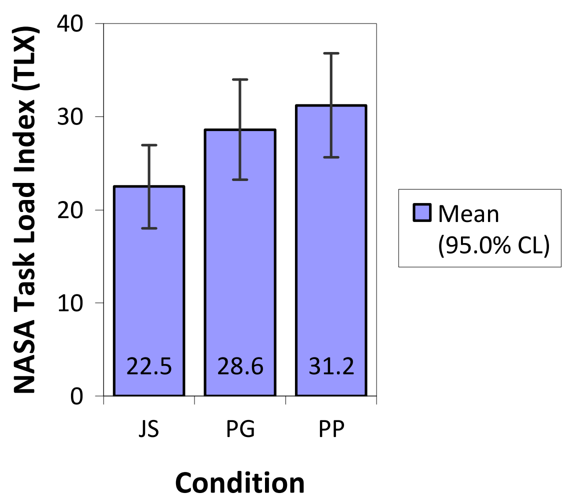


Figure 40: No significant differences in overall workload were observed across the three groups.

Table 9: No significant differences were noticed in workload between Point-and-Go and Path Planning.

Average Weighted Subjective Workload		t(53)	two-tailed	
Joystick	Point & Go	-3.10	p = 0.003	✓
Joystick	Path Planning	-3.26	p = 0.002	✓
Point & Go	Path Planning	-0.83	p = 0.41	

5.3.3 Usability Assessment

A usability questionnaire captured participant preferences for the JS, PG, and PP conditions. The results, which are the average response, are given in Table 10 and Table 11. In addition, there was a subjective question for which answers are provided in Table 12 and Table 13.

Table 10: Questionnaire Analysis

Questions	Average Answer	Answers	ANOVA P Value	Significant α = 0.05?
1. How easy was the joystick to learn and use?	5.4	1 - Extremely difficult. 2 - Reasonably difficult. 3 - Somewhat difficult. 4 - So-so. 5 - Somewhat easy. 6 - Reasonably easy. 7 - Extremely easy.	< 0.05	Yes, the Answers are different.
1. How easy was the Point and Go to learn and use?	6.4			
1. How easy was the joystick to learn and use?	6.7			
2. How effective was the Joystick?	4.1	1 – Extremely ineffective. 2 - Reasonably ineffective. 3 - Somewhat ineffective. 4 - So-so. 5 - Somewhat effective. 6 - Reasonably effective. 7 - Extremely effective.	< 0.05	Yes, the Answers are different.
2. How effective was Point and Go?	5.8			
2. How effective was Path Planning?	5.6			
3. How easy was it to remember Joystick commands?	6.3	1 - Extremely difficult. 2 - Reasonably difficult. 3 - Somewhat difficult. 4 - So-so. 5 - Somewhat easy. 6 - Reasonably easy. 7 - Extremely easy.	= 0.278	No, the Answers are not different.
3. How easy was it to remember Point and Go commands?	6.6			
3. How easy was it to remember Path Planning commands?	6.7			
4. How easy was it to prevent or correct mistakes with the Joystick?	4.3	1 - Extremely difficult. 2 - Reasonably difficult. 3 - Somewhat difficult. 4 - So-so. 5 - Somewhat easy. 6 - Reasonably easy. 7 - Extremely easy.	= 0.288	No, the Answers are not different.
4. How easy was it to prevent or correct mistakes with Point and Go?	5.1			
4. How easy was it to prevent or correct mistakes with Path Planning?	5			

Table 11: Questionnaire Analysis (Continue)

Questions	Average Answer	Answers	ANOVA P Value	Significant α = 0.05?
5. Overall, how satisfied were you with the Joystick controls?	3.8	1 - Extremely unsatisfied. 2 - Reasonably unsatisfied. 3 - Somewhat unsatisfied. 4 - So-so. 5 - Somewhat satisfied. 6 - Reasonably satisfied. 7 - Extremely satisfied.	< 0.05	Yes, the Answers are different.
5. Overall, how satisfied were you with the Point and Go controls?	5.8			
5. Overall, how satisfied were you with the Path Planning controls?	5.4			

Table 12: Table of Responses for Subjective Question

Question: Joystick, Point and Go or Path Planning Which do you prefer and Why?
1. Path planning due to the ease of use
2. I would prefer point and go as the system sits right now but if path planning had adjustable waypoints and a deselect option it could become more efficient.
3. Point and Go because it was easy to use and it went the fastest. There weren't long waiting times, and if there were I could move another robot in the meantime.
4. Point and go, ease of control, and the ability to control multiple robots movements instead of waiting for one to stop before moving another. Joystick was a close second.
5. I prefer Path Planning over everything because of its simplicity and the ability to control multiple robots. It is very effective as the operator can focus most of his concentration on finding the target rather than determining the best path for the robot.
6. Path Planning, because searching was easy and didn't have to worry about the obstacles. Easy to learn and easy to correct.
7. Point and Go. Because it is much easy and intuitive. Joystick does not have multiple robot control. Path Planning is not effective in this small test bed.
8. Path planning felt a little slower and got caught in the white areas every now and then, but I prefer it because you have a blend of being able to control all the robots at once while not having to check and correct every one of them every two seconds. You could watch them all and correct the few errors more easily because they didn't all need correcting at once.
9. Path planning because the projected path helped in correcting mistakes early on. Also, the physical load was much less.
10. The path planning because the robot did avoid the boards and when it failed I wasn't as frustrated. I could work easier with the 4 robots, probably way more than 4 if I had to. Easy and fun.
11. Path planning has my highest preference because it was the easiest to manipulate and still had a good path of pebbles to follow in order to locate the target. It combined the best factors of the joystick (good pebble path to follow) and the point and go (ease of use) modes.
12. Path planning because it required the least amount of work. It allowed me to send a robot and not worry about it while I focused on the other robots.

Table 13: Table of Responses for Subjective Question (Continue)

Question: Joystick, Point and Go or Path Planning Which do you prefer and Why?
13. Point and Go. Easiest to use and very easy to control multiple robots. Very responsive and could put a robot in box very fast and find target fast.
14. I prefer the Path Planning method overall. It is easy and quick to use and I had the ability to change the path before the robot hit an obstacle. It seems to work better then the Point and Go because I don't have to worry about the robot hitting an obstacle while traveling its to its destination.
15. The path planning was easiest because the obstacles were avoided by the computer with little error that needed to be fixed by the user.
16. Joystick for more control over a single robot. And point and go for being able to control multiple robots at the same time. Path planning takes too long so I didn’t like it.
17. I would choose point and go, because it allows me to set the path and also has a much better effectiveness (in terms of time). I would like some of the objectives from the path planning incorporated in the point and go. Especially, obstacle avoidance.
18. While the joystick was the most "fun" to use, path planning was the fastest and most effective.

5.4 Discussion

We analyzed the Human Factors evaluation of this system in which three interface conditions are tested for source detection tasks. Significant reductions in wait time were observed with Point-and-Go and Path Planning. Results show that the novel Augmented Reality multi-robot control (Point-and-Go and Path Planning) reduced mission completion times compared to the traditional joystick control for target detection missions.

There was a correlation between the switch count and the wait time. The switch count may affect the wait time. Participants switched more among the robots in Point-and-Go and Path Planning condition more than in Joystick condition, the mean wait time of the robots is reduced in Point-and-Go and Path Planning condition.

There was a correlation between the navigation error and the target and mission times in Point-and-Go condition and Path Planning condition. Participants had more navigation errors in the Path Planning condition than in the Point-and-Go condition. This may have caused participants to increase their target and mission completion times.

Participants experienced higher workload in Path Planning condition than in Point-and-Go condition. The navigation error may influence the subjective workload. Due to the difference of the robot power level, the navigation error had occurred in Path Planning. To optimize the Path Planning will reduce the navigation error.

Chapter 6: Summary, Contributions and Future Work

This thesis work provides a new solution for the human multi-robot control problem which is faced by all multi-robot tele-operation researchers. This chapter summarizes the demonstrated results and contributions of this work, followed by the future work of the thesis.

6.1 Summary

This dissertation has provided several key components of tele-operation and control for multi-robot. It has provided a human multi-robot interface for high level coordination for a team of robots. The Augmented Reality interface displays the visualization of sensor data, search path, and the control status of robots. The simultaneous use of these components can improve the performance of the user over the human multi-robot interface.

The research explains preliminary Human Factors evaluation of this system in which several interface conditions are tested for source detection tasks. Results show that the novel Augmented Reality multi-robot control (Point-and-Go and Path Planning) reduced mission completion times compared to the traditional joystick control for target detection missions.

The developed system is based on advanced Augmented Reality technologies also has broader impact and application. It will provide an easily translatable AR interface for aerial to ground robotics coordination applications in many different domains including space exploration, border security, homeland security, military robotics, and search and rescue events in hazardous condition. Other applications include sea applications where robots are used for search or cleanup of vast areas. In addition, direct (AR) linkage of medical robotic systems to patient data is of critical importance for successful operations. There is a significant opportunity for commercialization of this technology for multiple useful applications.

6.2 Contributions

The contributions of this thesis are as follows.

1. *Human multi-robot interface designed for high level command and control of teams of heterogeneous robots and individual control of single robo*t: The novel interface allows the user to interact with the ground robots by pointing and clicking on them from an over watch video view. An operator uses high-level commands to manipulate multiple robots along with advanced path planning algorithms for obstacles avoidance. It also allows the user to modify the user-selected goal position at any time to change the traverse path.
2. *Visualization of sensor and path information:* The AR interface displays virtual sensor information from each robot in real-time and drops arrows on the robot's path to generate a sensor data map. It allows the operator to use fused sensor information to augment decision making in order to direct multiple ground robots towards a source. The capabilities allow the robots to localize multiple sources simultaneously. In addition, the interface displays predictive paths for robot navigation.
3. *Validation of high-level commands of AR human multi-robot interface compared to a traditional joystick-based of AR human robot interface*: A

multi-robot interface was designed for multi-robot control that demonstrated better performance to traditional joystick-based robot control. We experimentally showed that significant reductions in wait time were observed with Point-and-Go and Path Planning. Results also show that the novel multi-robot control reduced mission completion times compared to the traditional joystick control for target detection missions.

6.3 Future Work

The research in this thesis points toward several lines of future work.

1. *Multi-human multi-robot tele-operation*: The system designed in this work is for a single tele-operator. Multi-human multi-robot tele-operation systems could support multiple human operators with the ability to jointly perform complex tasks, and share control in a remote environment while simultaneously receiving sensor feedback from multi-robots. The development of the multi-human multi-robots interfaces could be expanded base on the AR server and robot client architecture developed in this research. The interfaces could be designed to support all levels of human operation (direct manual control, tele-operation, shared control, and supervisory control), while also supporting multiple robot operators in multi-agent team configurations.
2. *Augmented Reality terrain data for ground robots obstacle avoidance*: The interface has only been designed and tested for the two dimensional case. Currently, the AR interface displays a two dimensional virtual lines representing the robots' path. The current path planning algorithm could

be extended to include terrain-based path planning and dynamic extrapolation of other robotic movements in its planning. It could give customized information about the distance to other structures, information about what that structure is connected to and a 3D-rendered augmentation of obstructed structures or hidden structures due to poor visibility or lighting.

3. *Advanced predictive display*: A physically-based Augmented Reality allows actual robots to interact with virtual objects or virtual robots to interact with real environment. Ground and aerial robots could be modeled on earth or for instance, lunar gravity situations. This would allow users to test maneuvers on virtual robots to see the effect before they attempt the task on the actual hardware. This predictive display would allow the users to not only test the task timeline, but, also mitigate the problems associated with time delay.

APPENDIX A: HIC APPROVAL

WAYNE STATE UNIVERSITY

HUMAN INVESTIGATION COMMITTEE
101 East Alexandrine Building
Detroit, Michigan 48201
Phone: (313) 577-1628
FAX: (313) 993-7122
http://hic.wayne.edu

NOTICE OF EXPEDITED CONTINUATION APPROVAL

To: Abhilash Pandya
Electrical & Computer Engineering
5050 Anthony Wayne D, Rm 3129

From: Ellen Barton, Ph.D. ____________
Chairperson, Behavioral Institutional Review Board (B3)

Date: April 14, 2010

RE: HIC #: 082105B3E(R)

Protocol Title: Human Factors Analysis for Robotic Teleoperation

Sponsor:
- U.S. ARMY TANK-AUTOMOTIVE & ARMAMENTS COMMAND (TACOM)
- NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
- CHILDREN'S HOSPITAL OF MICHIGAN

Protocol #: 0705004861

Expiration Date: April 13, 2011

Risk Level / Category: Research not involving greater than minimal risk

---

Continuation for the above-referenced protocol and items listed below (if applicable) were APPROVED following Expedited Review by the Chairperson/designee of the Wayne State University Institutional Review Board (B3) for the period of **04/14/2010 through 04/13/2011**. This approval does not replace any departmental or other approvals that may be required.

- Flyer
- Information Sheet (dated 5/22/08)
- Consent Form (dated 6/4/07)

- Federal regulations require that all research be reviewed at least annually. You *may* receive a *"Continuation Renewal Reminder"* approximately two months prior to the expiration date; however, it is the Principal Investigator's responsibility to obtain review and continued approval ***before*** the expiration date. Data collected during a period of lapsed approval is unapproved research and can ***never*** be reported or published as research data.
- All changes or amendments to the above-referenced protocol require review and approval by the HIC **BEFORE** implementation.
- Adverse Reactions/Unexpected Events (AR/UE) must be submitted on the appropriate form within the timeframe specified in the HIC Policy (http://www.hic.wayne.edu/hicpol.html).

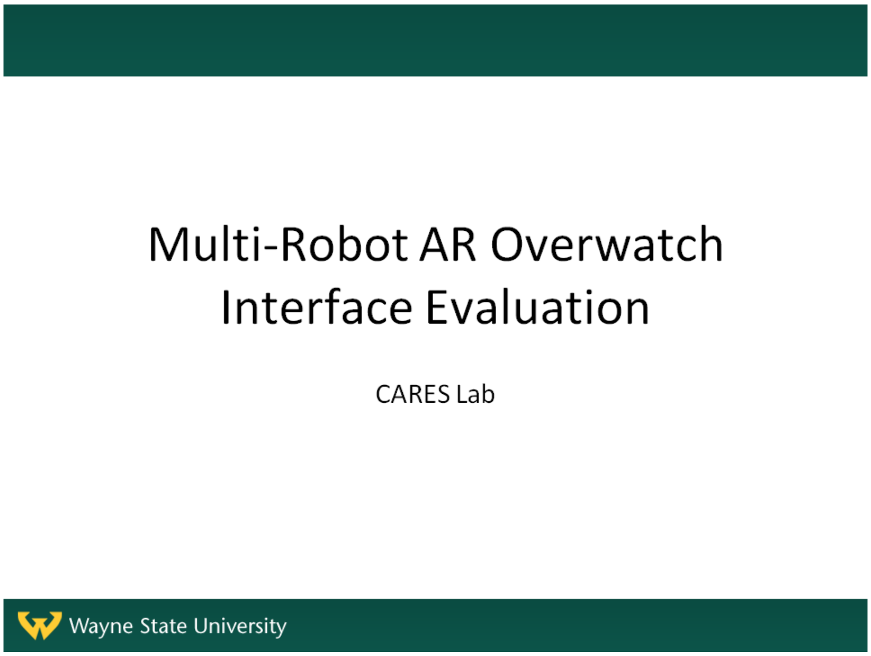
NOTE:

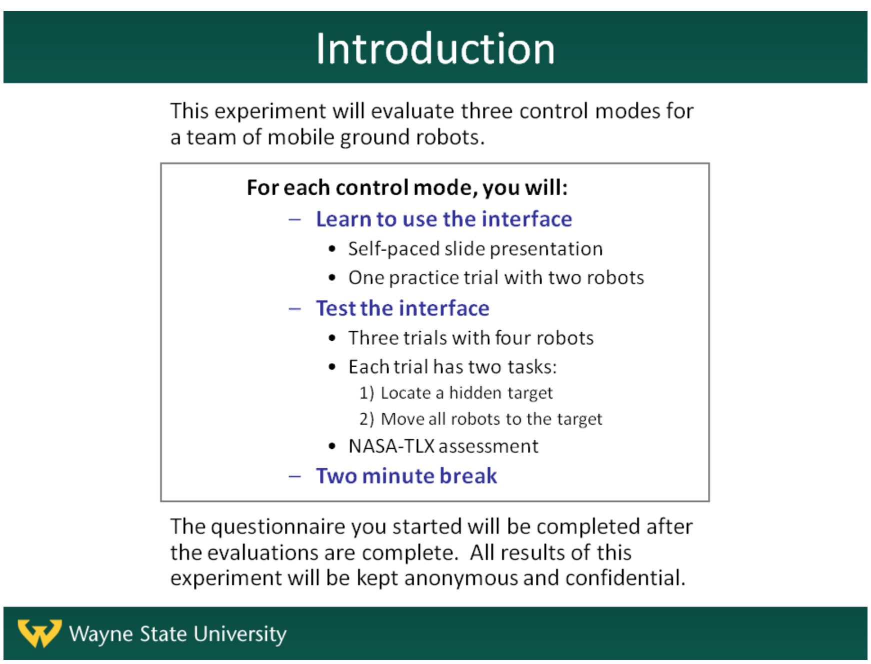
1. Upon notification of an impending regulatory site visit, hold notification, and/or external audit the HIC office must be contacted immediately.
2. Forms should be downloaded from the HIC website at ***each*** use.

*Based on the Expedited Review List, revised November 1998

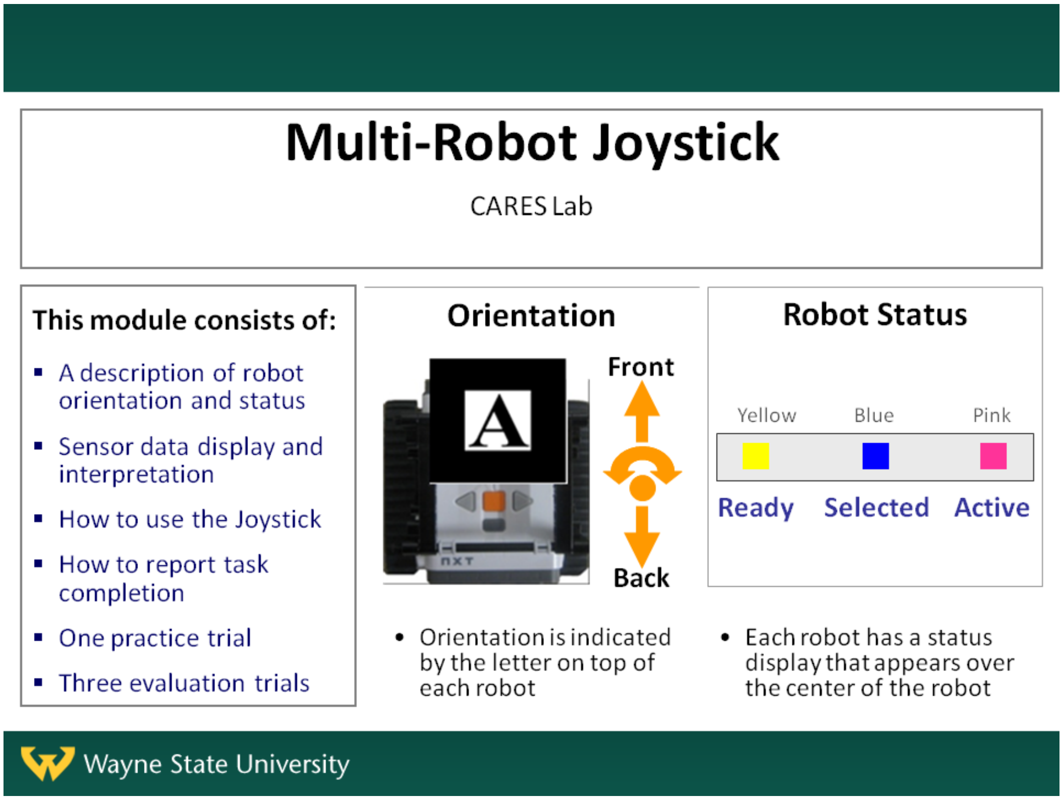
APPENDIX B: HUMAN FACTORS STUDY SUBJECT TESTING MATERIAL

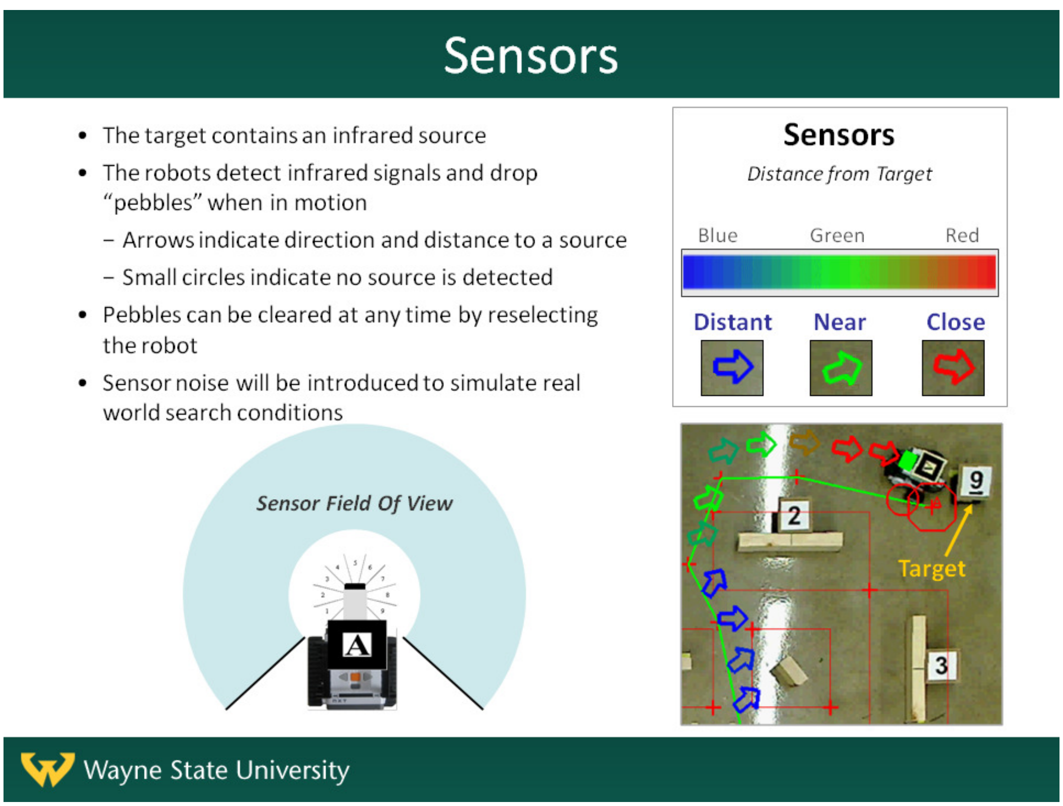
Self-Paced Instructional Presentation Slides- Introduction



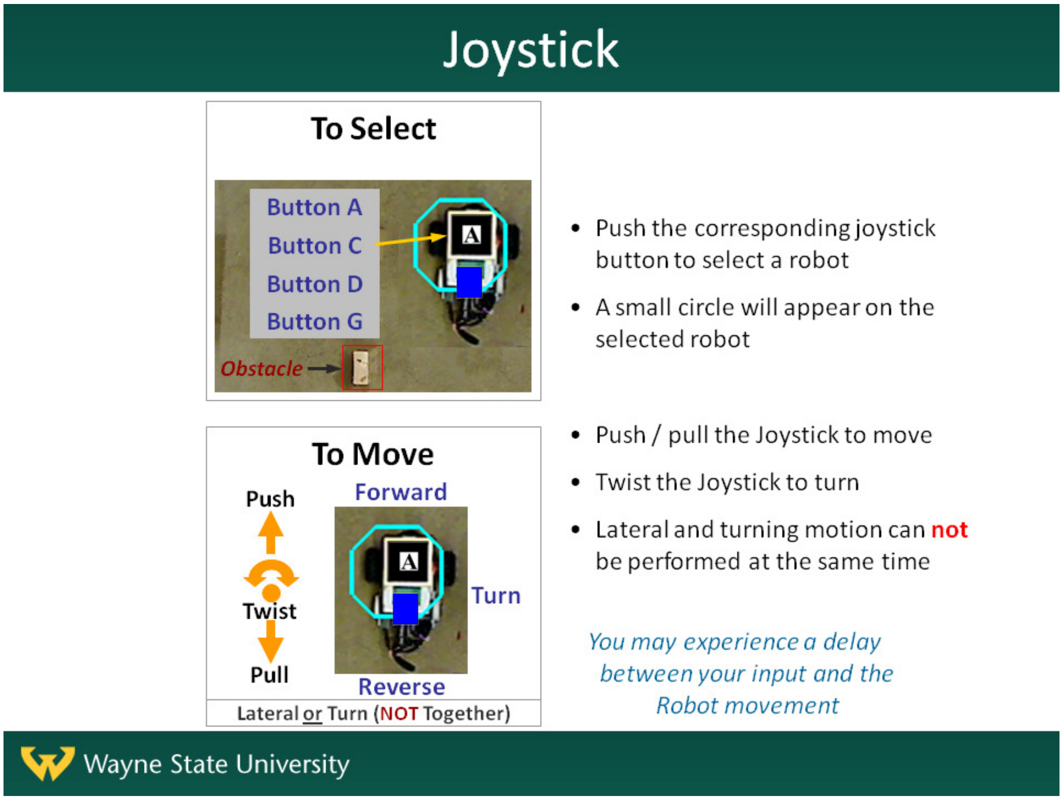


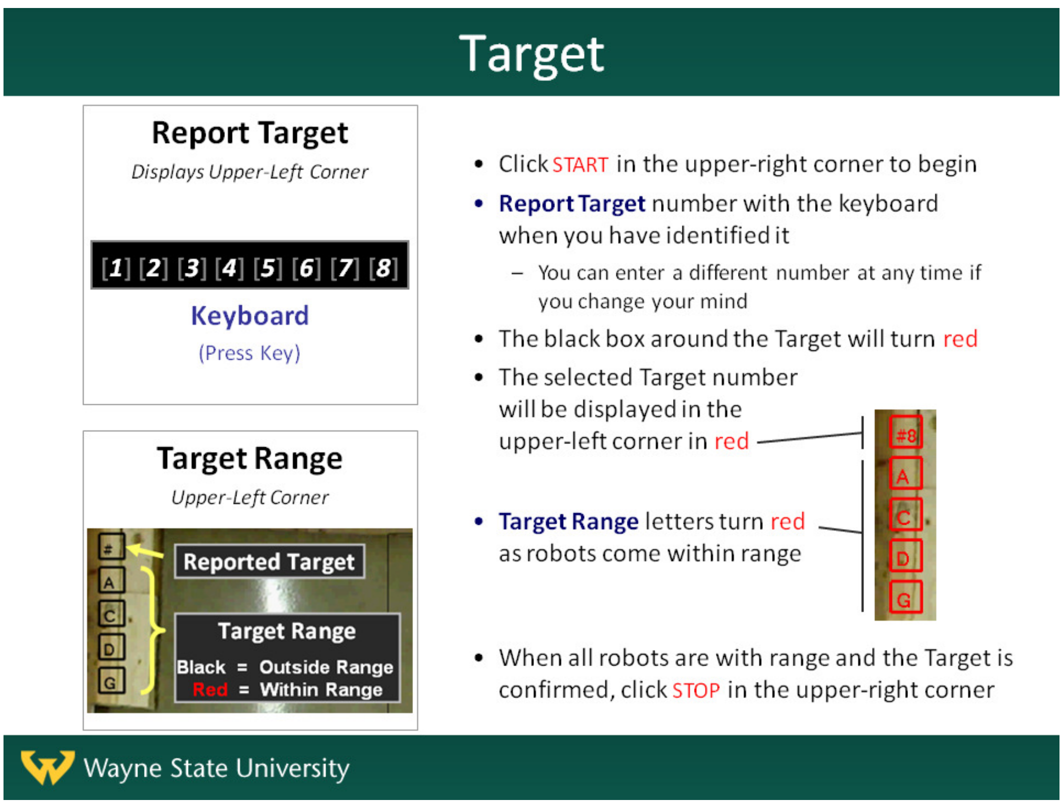
Self-Paced Instructional Presentation Slides- Joystick



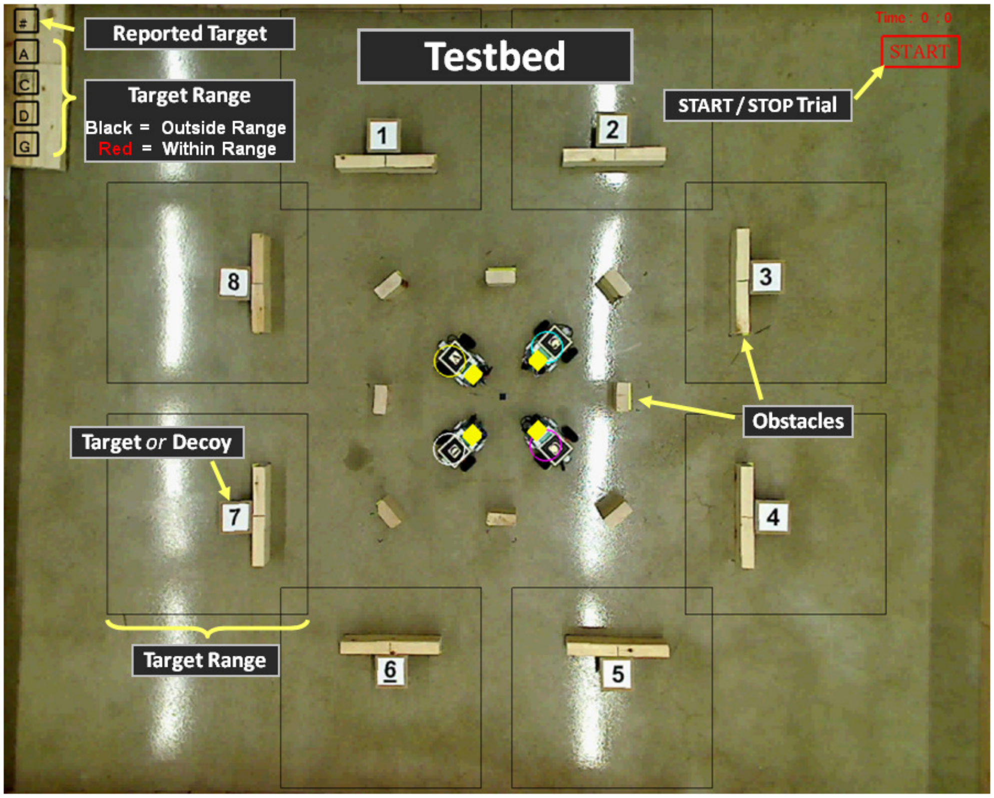


Self-Paced Instructional Presentation Slides- Joystick (Cont.)





Self-Paced Instructional Presentation Slides- Joystick (Cont.)



Joystick Practice

Practice Tasks

Click START to begin

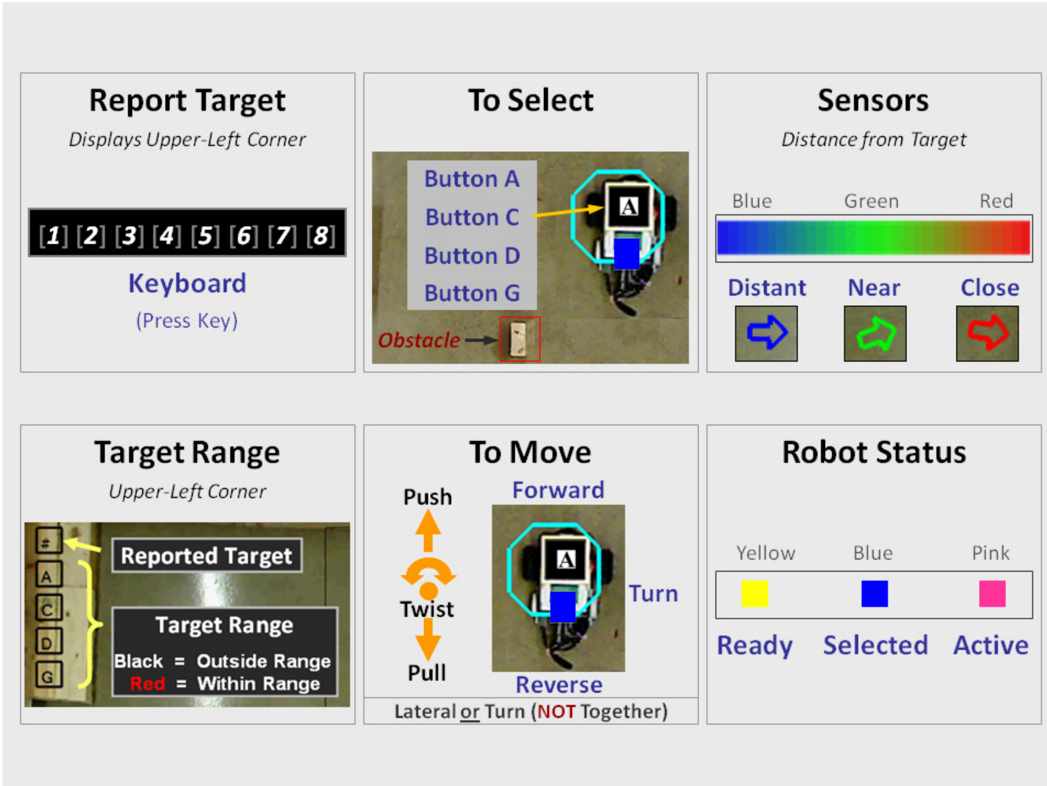
1) **Locate and Report the Target**
2) **Move Robots within Target Range**

Click STOP when done

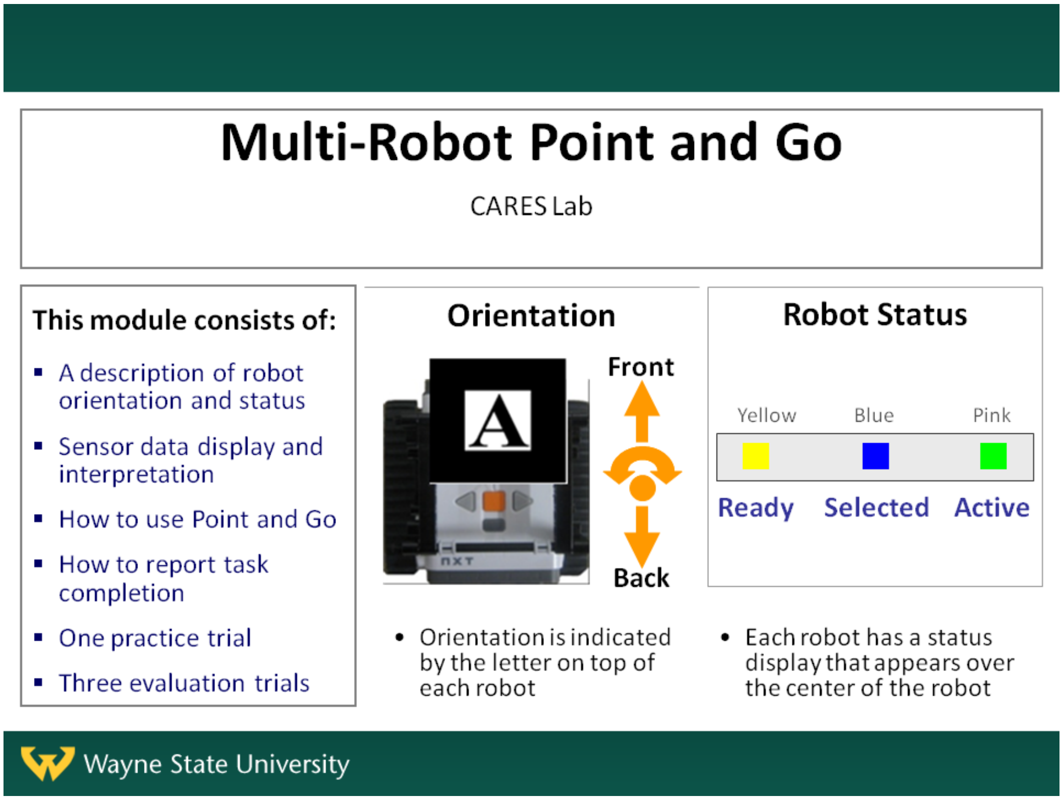
You must complete both tasks within the specified time
You have an unlimited number of attempts

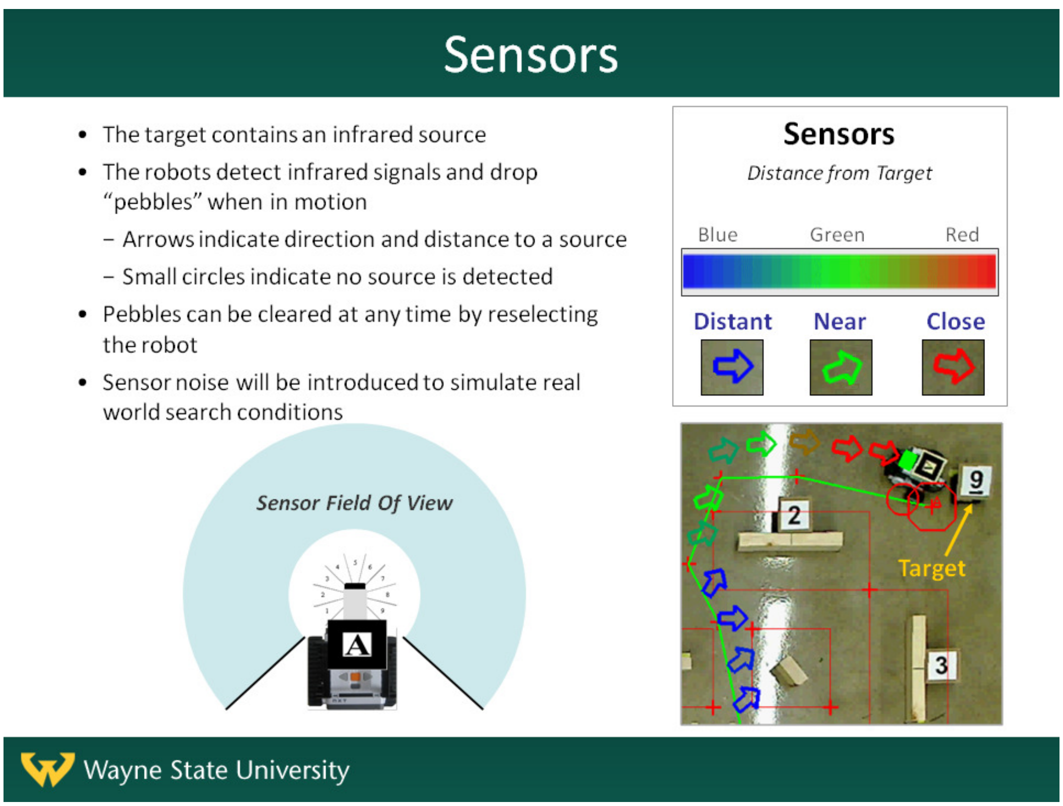
Wayne State University

Self-Paced Instructional Presentation Slides- Joystick (Cont.)

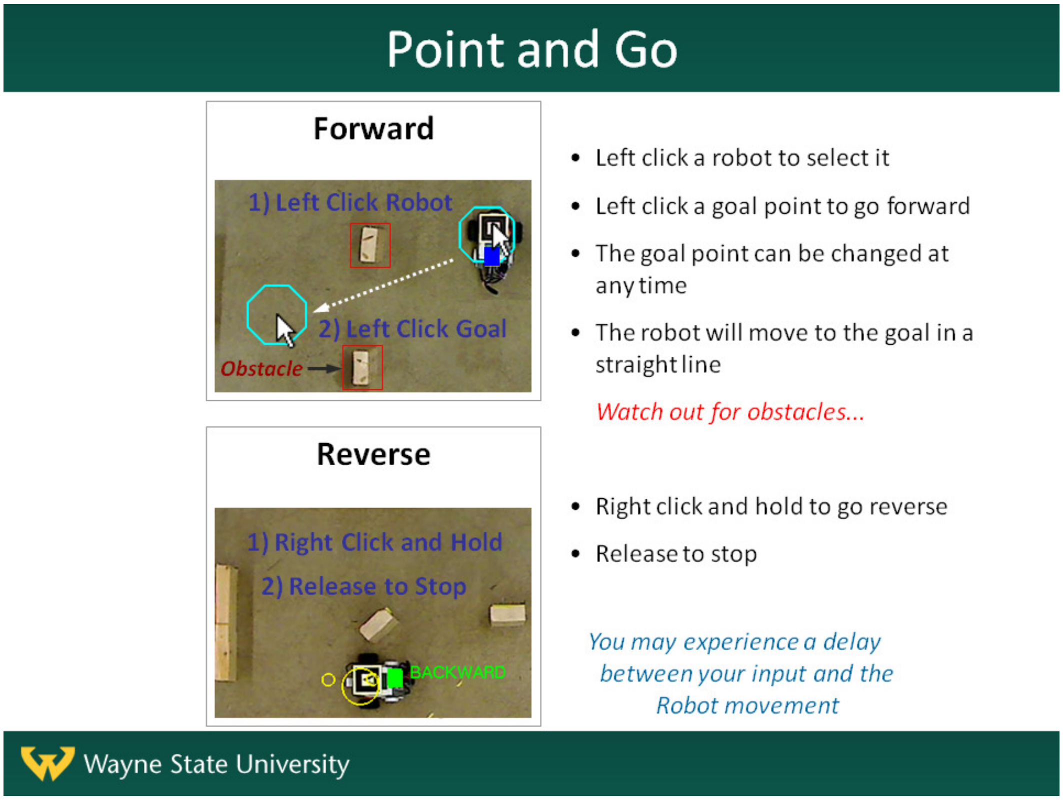


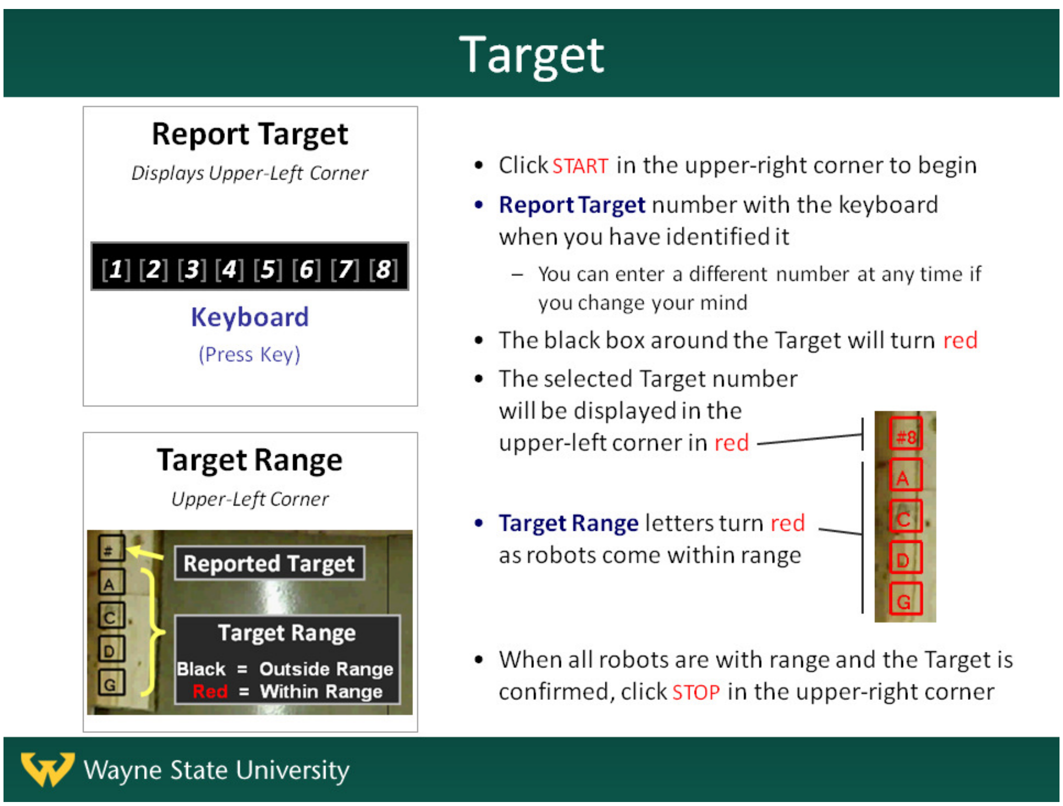
Self-Paced Instructional Presentation Slides- Point-and-Go



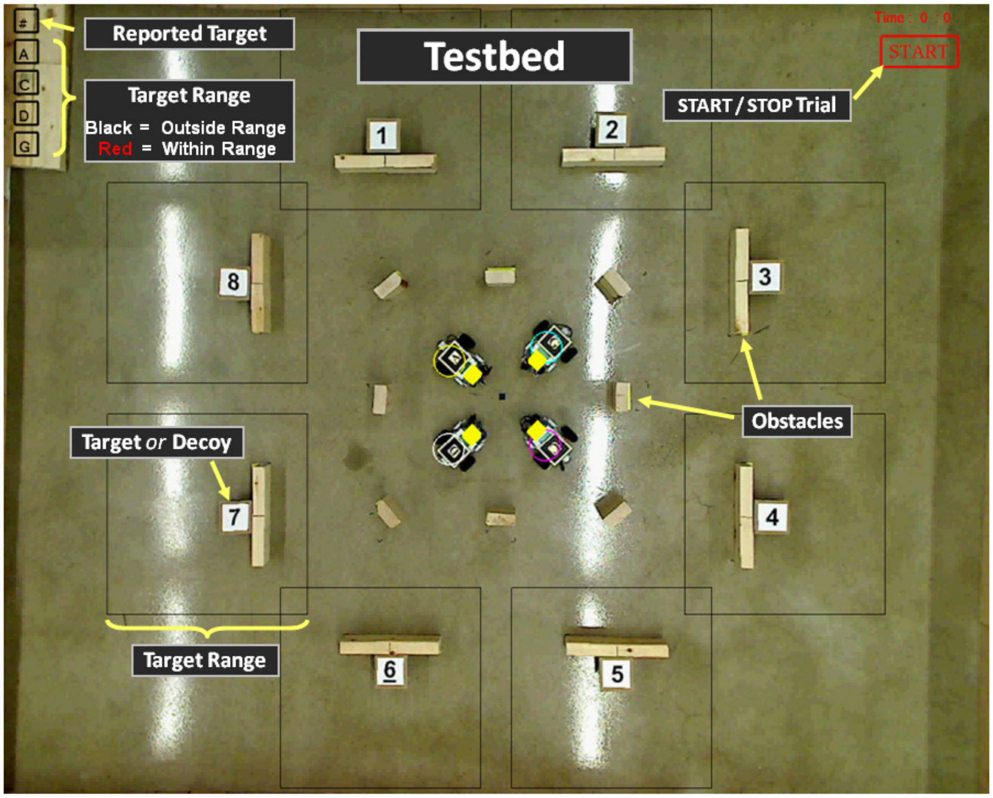


Self-Paced Instructional Presentation Slides- Point-and-Go (Cont.)





Self-Paced Instructional Presentation Slides- Point-and-Go (Cont.)



Point and Go Practice

Practice Tasks

Click START to begin

1) **Locate and Report the Target**
2) **Move Robots within Target Range**

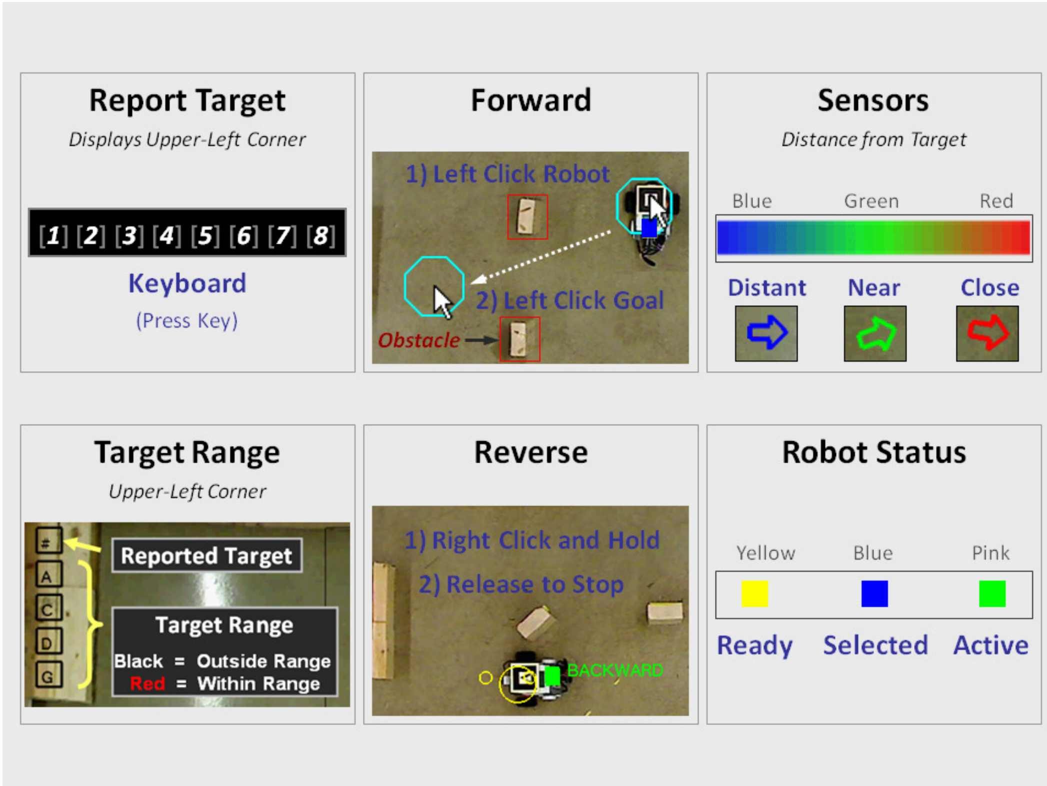
Click STOP when done

You must complete both tasks within the specified time

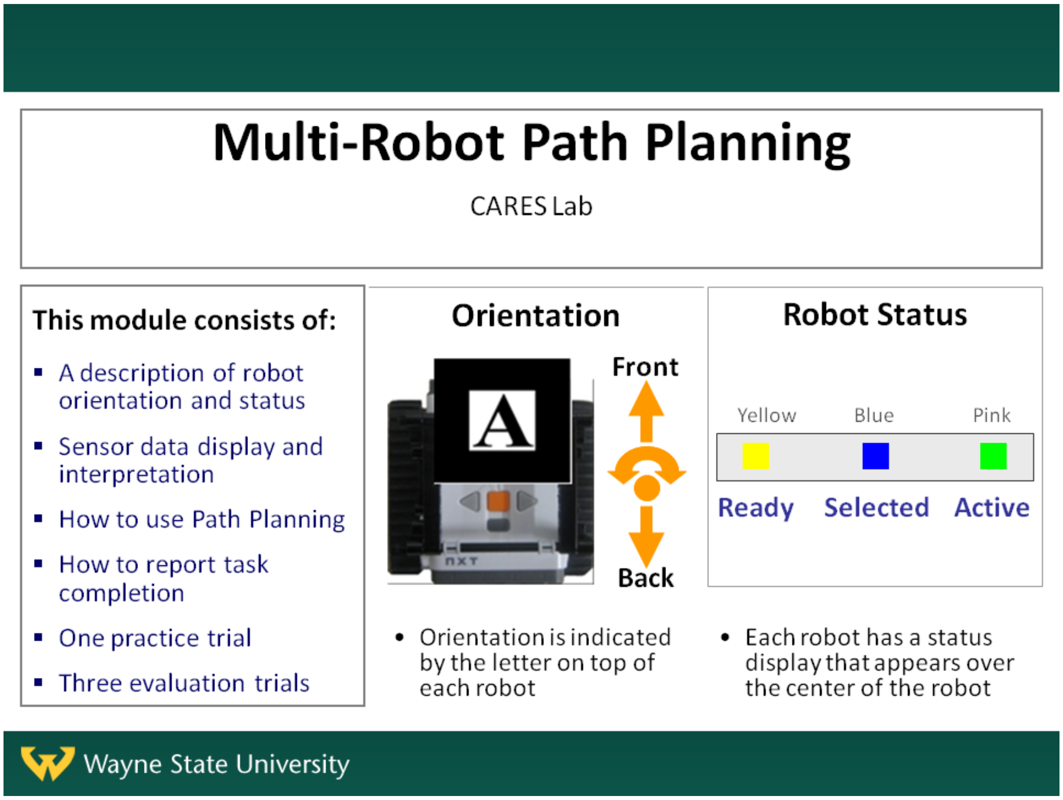
You have an unlimited number of attempts

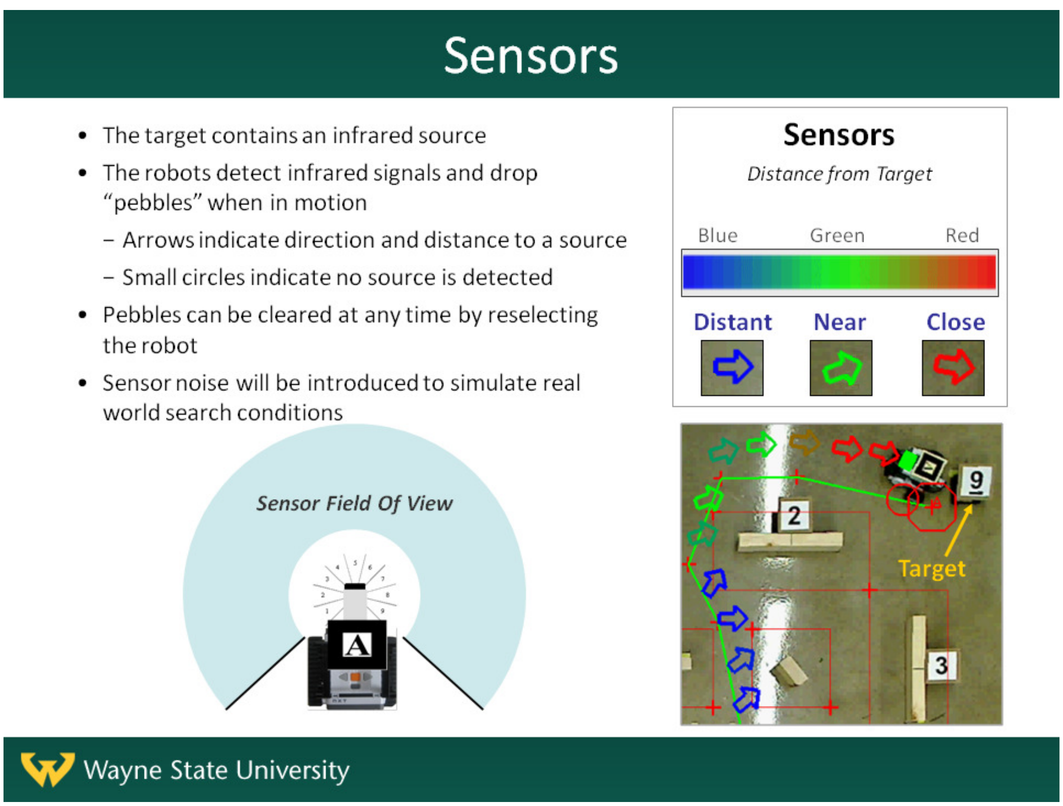
Wayne State University

Self-Paced Instructional Presentation Slides- Point-and-Go (Cont.)

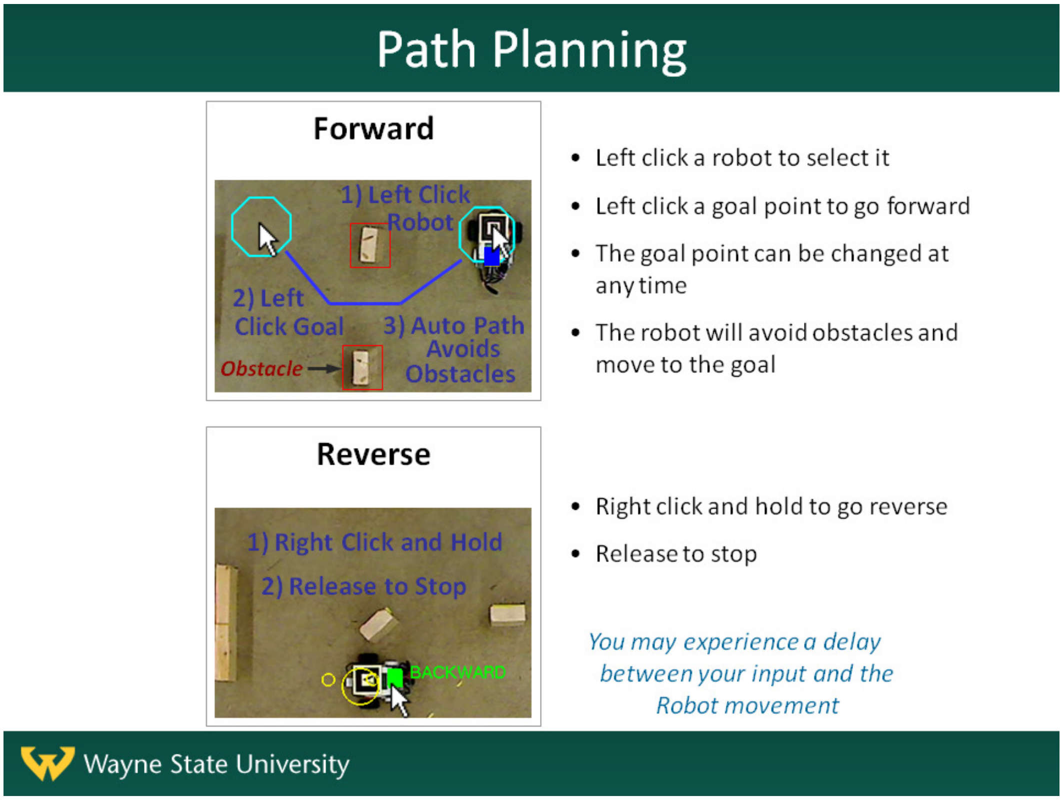


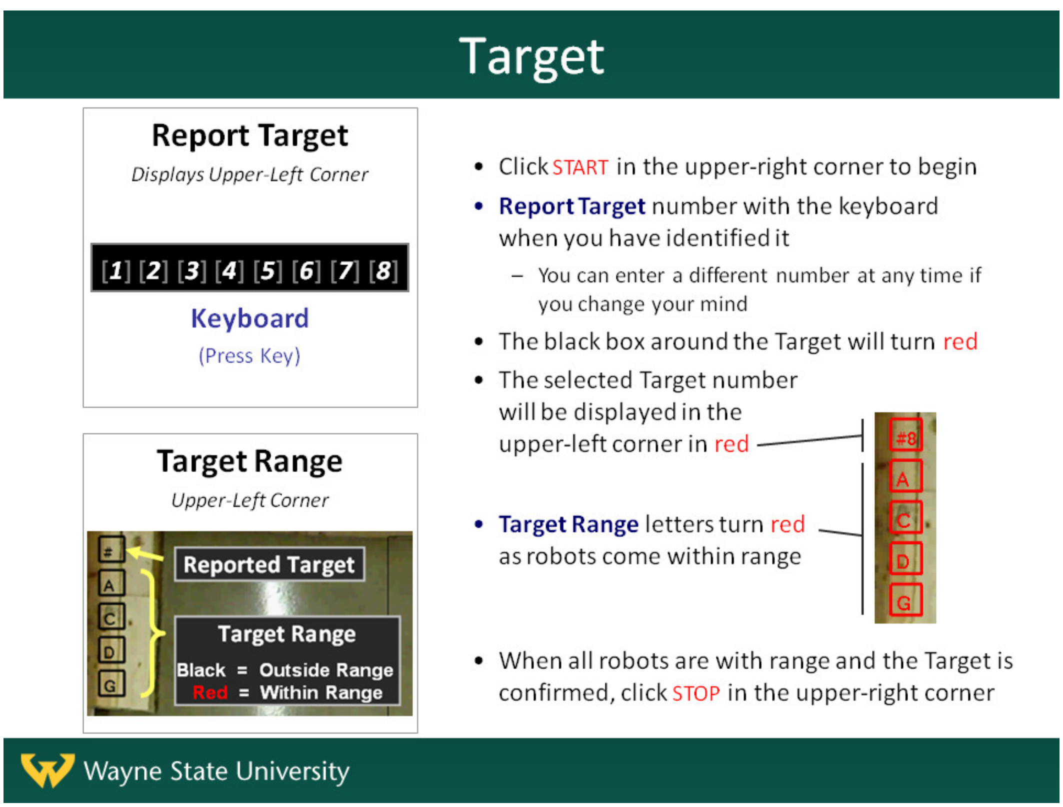
Self-Paced Instructional Presentation Slides- Path Planning



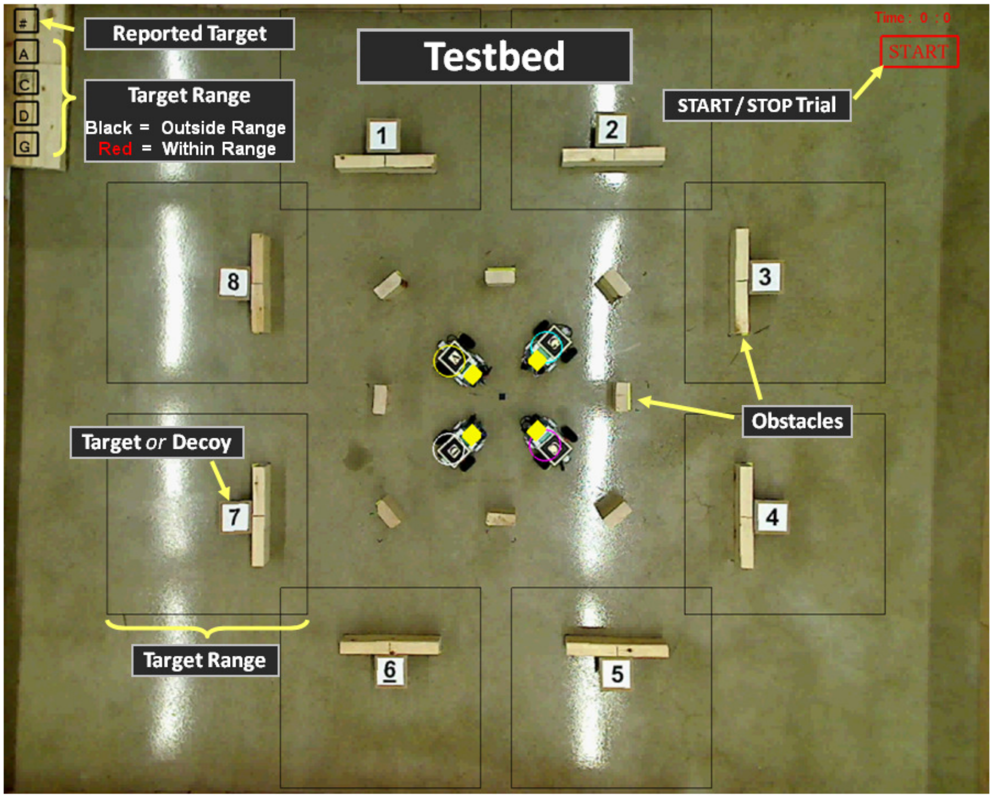


Self-Paced Instructional Presentation Slides- Path Planning (Cont.)





Self-Paced Instructional Presentation Slides- Path Planning (Cont.)



Path Planning Practice

Practice Tasks

Click START to begin

1) **Locate and Report the Target**
2) **Move Robots within Target Range**

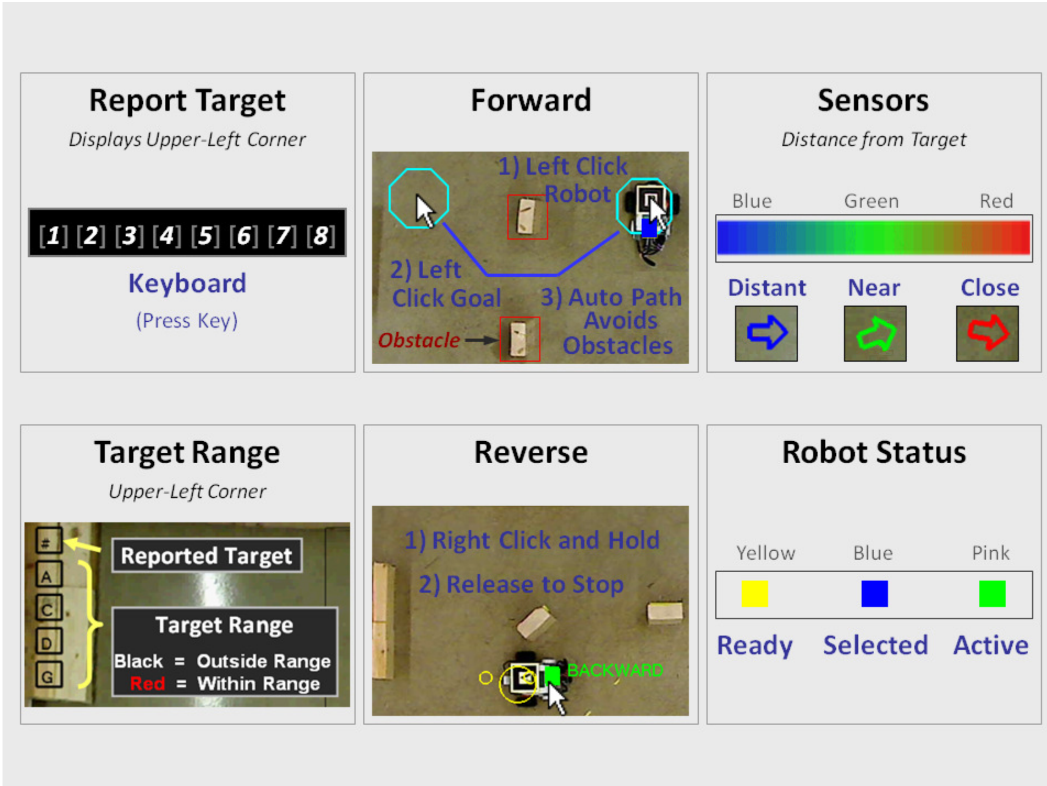
Click STOP when done

You must complete both tasks within the specified time

You have an unlimited number of attempts

Wayne State University

Self-Paced Instructional Presentation Slides- Path Planning (Cont.)



Pre-Test Questionnaire for the project

Multi-Robot AR Overwatch Interface Evaluation — **WSU ECE Dept. CARES Lab** — **v2.1**

Participant Questionnaire — **Subject:** 1 — Save | Clear

Unpopulated fields are marked Yellow. Areas shaded Blue are reserved for Lab use.

Section 1

Age ____ Gender (M / F) ____ Occupation ____

For each question, marking your response with an X

On average, up to how many hours a week do you drive an automobile?

0	4	7	11	14	More

On average, how often do you play video games?

Almost Never	Occasionally	Frequently	Almost Daily

Rate your level of experience with remote control devices (e.g. RC cars):

	1	2	3	4	5	6	7	
None								Extensive

Rate your level of experience controlling mobile ground robots:

	1	2	3	4	5	6	7	
None								Extensive

Rate your level of experience controlling multiple mobile robots:

	1	2	3	4	5	6	7	
None								Extensive

STOP HERE - The next section will be completed after the evaluations

Post-Test Questionnaire for the project

Section 2

Joystick For each question, mark your response with an X

How easy was the Joystick to learn and use?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

How effective was the Joystick?

	1	2	3	4	5	6	7	
Very Ineffective								**Very Effective**

How easy was it to remember Joystick commands?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

How easy was it to prevent or correct mistakes with the Joystick?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

Overall, how satisfied were you with the Joystick controls?

	1	2	3	4	5	6	7	
Very Unsatisfied								**Very Satisfied**

Comments:

Post-Test Questionnaire for the project (Cont.)

Section 3

Point and Go For each question, mark your response with an X

How easy was Point and Go to learn and use?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

How effective was Point and Go?

	1	2	3	4	5	6	7	
Very Ineffective								**Very Effective**

How easy was it to remember Point and Go commands?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

How easy was it to prevent or correct mistakes with Point and Go?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

Overall, how satisfied were you with the Point and Go controls?

	1	2	3	4	5	6	7	
Very Unsatisfied								**Very Satisfied**

Comments:

Post-Test Questionnaire for the project (Cont.)

Section 4

Path Planning For each question, mark your response with an X

How easy was Path Planning to learn and use?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

How effective was Path Planning?

	1	2	3	4	5	6	7	
Very Ineffective								**Very Effective**

How easy was it to remember Path Planning commands?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

How easy was it to prevent or correct mistakes with Path Planning?

	1	2	3	4	5	6	7	
Very Difficult								**Very Easy**

Overall, how satisfied were you with the Path Planning controls?

	1	2	3	4	5	6	7	
Very Unsatisfied								**Very Satisfied**

Comments:

Section 5

Joystick / Point and Go / Path Planning

Which do you prefer and why?

REFERENCES

[1] QinetiQ. *TALON Robot*. Available: http://www.qinetiq-na.com/products-talon.htm

[2] iRobot. *PackBot* Available: http://www.irobot.com/gi/ground/510_PackBot

[3] B. Yamauchi, "PackBot: A Versatile Platform for Military Robotics," in *Proceedings of SPIE 5422*, ed, 2004, pp. 228-237.

[4] H. G. Nguyen, "Robotics for law enforcement: applications beyond explosive ordnance disposal," *Proceedings of SPIE, the international society for optical engineering,* vol. 4232, pp. 433-54, 2001.

[5] R. R. Murphy, "Trial by fire [rescue robots]," *Robotics & Automation Magazine, IEEE,* vol. 11, pp. 50-61, 2004.

[6] H. Goldenberg, H. Najjaran, N. Kircanski, and A. A. Goldenberg, "Landmine detection using an autonomous terrain-scanning robot," *Industrial Robot: An International Journal,* vol. 32, pp. 240-247, 2005.

[7] *Multi Autonomous Ground-robotic International Challenge (MAGIC 2010).* Available: http://www.dsto.defence.gov.au/MAGIC2010/

[8] M. J. Del Signore, A. Czop, and K. Hacker, "Cooperative robotics: bringing autonomy to explosive ordnance disposal robots," Orlando, FL, USA, 2008, pp. 69621J-9.

[9] H. Wang, S. Y. Chien, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara, "Human teams for large scale multirobot control," presented at the Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 2009.

[10] Y. Nevatia, T. Stoyanov, R. Rathnam, M. Pfingsthorn, S. Markov, R. Ambrus, and A. Birk, "Augmented autonomy: Improving human-robot team performance in Urban search and rescue," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 2103-2108.

[11] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *Computer Graphics and Applications, IEEE,* vol. 21, pp. 34-47, 2001.

[12] R. Azuma, "A Survey of Augmented Reality," presented at the Presence, 1997.

[13] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Found. Trends Hum.-Comput. Interact.,* vol. 1, pp. 203-275, 2007.

[14] M. Endsley, "Level of automation effects on performance, situation awareness and workload in a dynamic control task," *Ergonomics,* vol. 42, pp. 462-492, 1999.

[15] J. P. Luck, "An investigation of real world control of robotic assets under communication latency," *1st Annual Conference on Human-Robot Interaction,* pp. 202-9, 2006.

[16] J. Wang and M. Lewis, "Mixed-initiative Multirobot Control in USAR," *Human Robot Interaction, Nilanjan Sarkar (Ed.), ISBN: 978-3-902613-13-4, I-Tech Education and Publishing, Available from: http://www.intechopen.com/articles/show/title/mixed-initiative_multirobot_control_in_usar,* 2007.

[17] J. W. Crandall and M. L. Cummings, "Identifying Predictive Metrics for Supervisory Control of Multiple Robots," *Robotics, IEEE Transactions on,* vol. 23, pp. 942-951, 2007.

[18] J. A. Adams, "Human management of a hierarchical system for the control of multiple mobile robots," University of Pennsylvania, 1995.

[19] J. A. Adams, "Multiple robot / single human interaction: effects on perceived workload," *Behaviour & Information Technology,* vol. 28, pp. 183 - 198, 2009.

[20] A. M. Naghsh, J. Gancet, A. Tanoto, and C. Roast, "Analysis and design of human-robot swarm interaction in firefighting," in *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, 2008, pp. 255-260.

[21] L. Alboul, H. S. Abdul-Rahman, P. S. Haynes, J. Penders, and J. Tharin, "An approach to multi-robot site exploration based on principles of self-organisation," in *3rd International Conference on Intelligent Robotics and Applications, ICIRA 2010, November 10, 2010 - November 12, 2010*, Shanghai, China, 2010, pp. 717-729.

[22] L. Alboul, J. Saez-Pons, and J. Penders, "Mixed human-robot team navigation in the GUARDIANS project," in *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, 2008, pp. 95-101.

[23] J. Saez-Pons, L. Alboul, J. Penders, and L. Nomdedeu, "Multi-robot team formation control in the GUARDIANS project," *Industrial Robot: An International Journal,* vol. 37, pp. 372-383, 2010.

[24] F. Mondada, G. C. Pettinaro, A. Guignard, I. W. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo, "Swarm-Bot: A New Distributed Robotic Concept," *Auton. Robots,* vol. 17, pp. 193-221, 2004.

[25] M. Dorigo, "SWARM-BOT: an experiment in swarm robotics," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 2005, pp. 192-200.

[26] E. Sahin, T. H. Labella, V. Trianni, J.-L. Deneubourg, P. Rasse, D. Floreano, L. Gambardella, F. Mondada, S. Nolfi, and M. Dorigo, "Swarm-bot: Pattern formation in a swarm of self-assembling mobile robots," in *2002 IEEE International Conference on Systems, Man and Cybernetics, October 6, 2002 - October 9, 2002*, Yasmine Hammamet, Tunisia, 2002, pp. 145-150.

[27] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous Self-Assembly in Swarm-Bots," *Robotics, IEEE Transactions on,* vol. 22, pp. 1115-1130, 2006.

[28] E. Tuci, R. Gross, V. Trianni, F. Mondada, M. Bonani, and M. Dorigo, "Cooperation through self-assembly in multi-robot systems," *ACM Trans. Auton. Adapt. Syst.,* vol. 1, pp. 115-150, 2006.

[29] V. Trianni, S. Nolfi, and M. Dorigo, "Cooperative hole avoidance in a swarm-bot," in *Intelligent Autonomous systems, April 10, 2005 - April 13, 2005*, 2006, pp. 97-103.

[30] S. Nouyan and M. Dorigo, "Chain based path formation in swarms of robots," in *Ant Colony Optimization and Swarm Intelligence - 5th International Workshop, ANTS 2006, Proceedings, September 4, 2006 - September 7, 2006*, Brussels, Belgium, 2006, pp. 120-131.

[31] J. F. Roberts, J. C. Zufferey, and D. Floreano, "Energy management for indoor hovering robots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 1242-1247.

[32] M. Bonani, St\, \#233, p. Magnenat, P. R\, tornaz, and F. Mondada, "The Hand-Bot, a Robot Design for Simultaneous Climbing and Manipulation," presented at the Proceedings of the 2nd International Conference on Intelligent Robotics and Applications, Singapore, 2009.

[33] C. Pinciroli, R. O'Grady, A. L. Christensen, and M. Dorigo, "Self-organised recruitment in a heteregeneous swarm," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 2009, pp. 1-8.

[34] J. McLurkin, "Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots," phdS.M. thesis, Massachusetts Institute of Technology, 2004.

[35] J. McLurkin, "Distributed algorithms for multi-robot systems," presented at the Proceedings of the 6th international conference on Information processing in sensor networks, Cambridge, Massachusetts, USA, 2007.

[36] J. McLurkin and D. Yamins, "Dynamic Task Assignment in Robot Swarms," *Proceedings of Robotics: Science and Systems, June,* vol. 8, 2005.

[37] J. McLurkin and J. Smith, "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots," *7th International Symposium on Distributed Autonomous Robotic Systems (DARS),* 2004.

[38] J. Rothermich, M. Ecemiş, and P. Gaudiano, "Distributed Localization and Mapping with a Robotic Swarm

Swarm Robotics." vol. 3342, E. Sahin and W. Spears, Eds., ed: Springer Berlin / Heidelberg, 2005, pp. 58-69.

[39] D. Cruz, J. McClintock, B. Perteet, O. A. A. Orqueda, C. Yuan, and R. Fierro, "Decentralized cooperative control - A multivehicle platform for research in networked embedded systems," *Control Systems Magazine, IEEE,* vol. 27, pp. 58-78, 2007.

[40] H. Yanco and J. Drury, "Rescuing interfaces: A multi-year study of human-robot interaction at the AAAI Robot Rescue Competition," *Autonomous Robots,* vol. 22, pp. 333-352, 2007.

[41] J. L. Marble, "Evaluation of supervisory vs. peer-peer interaction with human-robot teams," *Proceedings of the 37th Annual Hawaii International Conference on System Sciences,* pp. 9-9, 2004.

[42] J. L. Marble, D. J. Bruemmer, and D. A. Few, "Lessons learned from usability tests with a collaborative cognitive workspace for human-robot teams," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, 2003, pp. 448-453 vol.1.

[43] C. M. Humphrey, C. Henk, G. Sewell, B. W. Williams, and J. A. Adams, "Assessing the scalability of a multiple robot interface," presented at the Proceedings of the ACM/IEEE international conference on Human-robot interaction, Arlington, Virginia, USA, 2007.

[44] I. C. Envarli and J. A. Adams, "Task lists for human-multiple robot interaction," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, 2005, pp. 119-124.

[45] T. Fong, C. Thorpe, and B. Glass, "PdaDriver: A Handheld System for Remote Driving," 2003.

[46] T. Fong and C. Thorpe, "Vehicle Teleoperation Interfaces," *Auton. Robots,* vol. 11, pp. 9-18, 2001.

[47] T. Fong, C. Thorpe, and C. Baur, "Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Remote Driving Tools," *Auton. Robots,* vol. 11, pp. 77-85, 2001.

[48] T. Fong, "Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation," Doctoral Dissertation, Robotics Institute, Carnegie Mellon University, 2001.

[49] T. W. Fong and C. Thorpe, "Robot as partner: vehicle teleoperation with collaborative control," in *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, 2002.

[50] T. Fong, S. Grange, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," in *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, 2001, pp. 237-242.

[51] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *Industrial Electronics, IEEE Transactions on,* vol. 50, pp. 699-704, 2003.

[52] T. Fong, C. Thorpe, and C. Baur, "Collaboration, Dialogue, Human-Robot Interaction," in *Robotics Research*. vol. 6, R. Jarvis and A. Zelinsky, Eds., ed: Springer Berlin / Heidelberg, 2003, pp. 255-266.

[53] S. T. Hayes, E. R. Hooten, and J. A. Adams, "Multi-touch interaction for tasking robots," presented at the Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction, Osaka, Japan, 2010.

[54] M. Micire, J. L. Drury, B. Keyes, and H. A. Yanco, "Multi-touch interaction for robot control," presented at the Proceedings of the 14th international conference on Intelligent user interfaces, Sanibel Island, Florida, USA, 2009.

[55] M. Micire, E. McCann, M. Desai, K. M. Tsui, A. Norton, and H. A. Yanco, "Hand and finger registration for multi-touch joysticks on software-based operator control units," in *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, 2011, pp. 88-93.

[56] M. Micire, M. Desai, J. L. Drury, E. McCann, A. Norton, K. M. Tsui, and H. A. Yanco, "Design and validation of two-handed multi-touch tabletop controllers for robot teleoperation," presented at the Proceedings of the 16th international conference on Intelligent user interfaces, Palo Alto, CA, USA, 2011.

[57] M. J. Micire, "Multi-Touch Interaction for Robot Command and Control," University of Massachusetts Lowell, 2010.

[58] M. Micire, M. Desai, A. Courtemanche, K. M. Tsui, and H. A. Yanco, "Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces," presented at the Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, Banff, Alberta, Canada, 2009.

[59] J. Kato, D. Sakamoto, M. Inami, and T. Igarashi, "Multi-touch interface for controlling multiple mobile robots," presented at the Proceedings of the 27th

international conference extended abstracts on Human factors in computing systems, Boston, MA, USA, 2009.

[60] L. A. Nguyen, M. Bualat, L. J. Edwards, L. Flueckiger, C. Neveu, K. Schwehr, M. D. Wagner, and E. Zbinden, "Virtual Reality Interfaces for Visualization and Control of Remote Vehicles," *Auton. Robots,* vol. 11, pp. 59-68, 2001.

[61] K. Schwehr, A. Derbes, L. Edwards, L. Nguyen, and E. Zbinden, "Designing visualization software for ships and robotic vehicles," in *Proceedings of SPIE-IS and T Electronic Imaging - Visualization and Data Analysis 2005, January 17, 2005 - January 18, 2005*, San Jose, CA, United states, 2005, pp. 226-237.

[62] J. Wright, A. Trebi-Ollennu, F. Hartman, B. Cooper, S. Maxwell, J. Yen, and J. Morrison, "Terrain Modelling for In-situ Activity Planning and Rehearsal for the Mars Exploration Rovers," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 2005, pp. 1372-1377.

[63] N. Mollet, L. G. Brayda, R. Chellali, and J. G. Fontaine, "Virtual Environments and Scenario Languages for Advanced Teleoperation of Groups of Real Robots: Real Case Application," in *Advances in Computer-Human Interactions, 2009. ACHI '09. Second International Conferences on*, 2009, pp. 310-316.

[64] K. Baizid, Z. Li, N. Mollet, and R. Chellali, "Human Multi-robots Interaction with High Virtual Reality Abstraction Level," in *Intelligent Robotics and Applications*. vol. 5928, M. Xie, Y. Xiong, C. Xiong, H. Liu, and Z. Hu, Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 23-32.

[65] J. R. Vallino, "Interactive Augmented Reality," PhD thesis, University of Rochester, New York, 1998.

[66] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-Robot Collaboration: A Literature Review and Augmented Reality Approach in Design," *International Journal of Advanced Robotic Systems, ISSN: 1729-8806, InTech, Available from: http://www.intechopen.com/articles/show/title/human-robot_collaboration__a_literature_review_and_augmented_reality_approach_in_design,* 2008.

[67] S. A. Green, "An Augmented Reality Human-Robot Collaboration System," PhD thesis, University of Canterbury, 2008.

[68] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Augmented Reality for Human-Robot Collaboration," *Human Robot Interaction, Nilanjan Sarkar (Ed.), ISBN: 978-3-902613-13-4, I-Tech Education and Publishing, Available from: http://www.intechopen.com/articles/show/title/augmented_reality_for_human-robot_collaboration,* 2007.

[69] K. Chintamani, A. Cao, R. D. Ellis, and A. K. Pandya, "Improved Telemanipulator Navigation During Display-Control Misalignments Using Augmented Reality Cues," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 40, pp. 29-39, 2010.

[70] B. Giesler, T. Salb, P. Steinhaus, and R. Dillmann, "Using augmented reality to interact with an autonomous mobile platform," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2004, pp. 1009-1014 Vol.1.

[71] D. A. Wang, F. Bello, and A. Darzi, "Augmented reality provision in robotically assisted minimally invasive surgery," *International Congress Series,* vol. 1268, pp. 527-532, 2004.

[72] T. H. J. Collett and B. A. MacDonald, "Developer oriented visualisation of a robot program," presented at the Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, Salt Lake City, Utah, USA, 2006.

[73] T. H. J. Collett and B. A. MacDonald, "Augmented reality visualisation for player," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 3954-3959.

[74] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone Robotics," *Autonomous Robots,* vol. 11, pp. 319-324, 2001.

[75] D. Mike, "World Embedded Interfaces for Human-Robot Interaction," 2003, pp. 125b-125b.

[76] D. Claros, M. D. Haro, M. Dominguez, C. D. Trazegnies, C. Urdiales, and F. Sandoval, "Augmented reality visualization interface for biometric wireless sensor networks," presented at the Proceedings of the 9th international work conference on Artificial neural networks, San Sebastián, Spain, 2007.

[77] J. E. Young, E. Sharlin, and J. E. Boyd, "The Use of Harr-like Features in Bubblegrams: a Mixed Reality Human-Robot Interaction Technique," ed: University of Calgary, 2006.

[78] J. E. Young, E. Sharlin, and J. E. Boyd, "Implementing Bubblegrams: The Use of Haar-Like Features for Human-Robot Interaction," in *Automation Science and*

Engineering, 2006. CASE '06. IEEE International Conference on, 2006, pp. 298-303.

[79] S. A. Green, X. Q. Chen, M. Billinghurst, and J. G. Chase, "Collaborating with a Mobile Robot: An Augmented Reality Multimodal Interface," ed: University of Canterbury. Human Interface Technology Laboratory., 2008.

[80] S. A. Green, J. G. Chase, C. XiaoQi, and M. Billinghurst, "Evaluating the Augmented Reality Human-Robot Collaboration System," in *Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference on*, 2008, pp. 521-526.

[81] S. A. Green, "Evaluating the augmented reality human-robot collaboration system," *International journal of intelligent systems technologies and applications,* vol. 8, p. 130, 2010.

[82] N. Mollet, R. Chellali, and L. Brayda, "Virtual and Augmented Reality Tools for Teleoperation: Improving Distant Immersion and Perception," in *Transactions on Edutainment II*. vol. 5660, Z. Pan, A. Cheok, W. Müller, and A. Rhalibi, Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 135-159.

[83] N. Mollet and R. Chellali, "Virtual and Augmented Reality with Head-Tracking for Efficient Teleoperation of Groups of Robots," in *Cyberworlds, 2008 International Conference on*, 2008, pp. 102-108.

[84] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological Interfaces for Improving Mobile Robot Teleoperation," *Robotics, IEEE Transactions on,* vol. 23, pp. 927-941, 2007.

[85] C. W. Nielsen, D. A. Few, and D. S. Athey, "Using mixed-initiative human-robot interaction to bound performance in a search task," in *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, 2008, pp. 195-200.

[86] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *In: IEICE Trans. Information Systems,* vol. E77-D, pp. 1321-1329, 1994.

[87] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*, 1999, pp. 85-94.

[88] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *Robotics and Automation, IEEE Journal of,* vol. 3, pp. 323-344, 1987.

[89] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on,* vol. 12, pp. 566-580, 1996.

[90] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.

[91] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Systems Science and Cybernetics, IEEE Transactions on,* vol. 4, pp. 100-107, 1968.

[92] N. J. Nilsson, "A Mobile Automaton: An Application of Artificial Intelligence Techniques," in *in Proc. IJCAI*, 1969, pp. 509-520.

[93] A. Cao, K. K. Chintamani, A. K. Pandya, and R. D. Ellis, "NASA TLX: software for assessing subjective mental workload.," *Behavior Research Methods,* vol. 41, pp. 113-117, 2009.

[94] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," in *Advances in Psychology*. vol. Volume 52, A. H. Peter and M. Najmedin, Eds., ed: North-Holland, 1988, pp. 139-183.

ABSTRACT

AN AUGMENTED REALITY INTERFACE FOR MULTI-ROBOT TELE-OPERATION AND CONTROL

by

SAM LEE

December 2011

Advisors: Dr. Abhilash Pandya

Major: Electrical Engineering

Degree: Doctor of Philosophy

Teams of heterogeneous robots with different dynamics or capabilities can perform a variety of tasks such as multipoint surveillance, cooperative transport and explorations in hazardous environments. However, the operation of these teams of robots by a human operator is a major challenge, particularly in search and rescue applications. This research created a seamlessly controlled multi-robot system comprised of ground robots of semi-autonomous nature for source detection tasks. The system combines augmented reality interface capabilities with human supervisor's ability to control multiple robots. The thesis studies a preliminary Human Factors evaluation of this system in which several interface conditions are tested for source detection tasks. Results show that the novel Augmented Reality multi-robot control (Point-and-Go and Path Planning) reduced mission completion times compared to the traditional joystick control for target detection missions.

AUTOBIOGRAPHICAL STATEMENT

SAM YUNG-SEN LEE

Sam Lee is a doctoral candidate at the department of Electrical and Computer Engineering at Wayne State University. He received his MS degree in Electrical Engineering from University of Detroit Mercy in 2003.

His primary research interests are in multi-robot systems, cooperative control of heterogeneous robots and human-robot interfaces. At WSU, his research has focus on Augmented Reality interfaces to improve tele-operation performance for multi-robot control.

Selected publications:

- **S. Lee**, N. P. Lucas, A. Cao, A. Pandya, and R. D. Ellis, "An Augmented Reality UAV-Guided Ground Navigation Interface Improve Human Performance in Multi-robot Tele-operation," presented at the 2011 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Modeling and Simulation, Testing and Validation (MSTV) Mini-symposium, Dearborn, MICHIGAN, 2011.
- **S. Lee**, S. Hunt, A. Cao, and A. Pandya, "Virtual Interface with Guarded Teleoperation Control of Multiple Heterogeneous Robots," presented at the 2010 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Modeling and Simulation, Testing and Validation (MSTV) Mini-symposium, Dearborn, MICHIGAN, 2010.
- **S. Y.-S. Lee**, S. Hunt, A. Cao, and A. Pandya, "Combined virtual and real robotic test-bed for single operator control of multiple robots", Proc. SPIE 7692, 769208 (2010); doi:10.1117/12.850166
- **Y.-S. Li**, S. Hunt, C. Popovici, S. Walter, G. Witus, R. D. Ellis, G. Auner, A. Cao, and A. Pandya, "Development of an extendable arm and software architecture for autonomous and tele-operated control for mobile platforms", Proc. SPIE 6962, 69621U (2008); doi:10.1117/12.778059
- S. Hunt, **Y.-S. Li**, G. Witus, S. Walter, R. D. Ellis, G. Auner, A. Cao, and A. Pandya, "Integration of new communications and mast subsystems on an Omni-Directional Inspection Robot (ODIS)", Proc. SPIE 6736, 67360A (2007); doi:10.1117/12.738078