Multi-modal Human Robot Interaction in a Simulation Environment

Dept. of CIS - Senior Design 2012-2013

Justin Cockburn
jusco@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Mubbasir Kapadia
mubbasir@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Yonas Solomon
soyonas@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Norman Badler
badler@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT

Human Robot Interaction (HRI) has been the subject of extensive research with many different proposed control modalities including gestures, computer vision, and teleoperation. The increasing use of robotic assets in the military, industry, and even household settings has led to a growing need for developing more intuitive and flexible HRI interfaces. In order to address this demand, an interactive simulation that facilitates the objective evaluation of multi-modal HRI interaction is presented. An HRI interface is implemented and tested on existing software framework in Unity [13]. Evaluation of the system is based on mission parameters and user input. Audio commands and gesture-based controls direct robots in the simulation environment, granting these robots a higher level of autonomy. The robotic agents are tasked with correctly interpreting and behaving in response to foreign sounds and directional commands. In doing so, a more intuitive means of controlling and monitoring robots is created and new ways of developing more user-friendly HRIs are demonstrated.

1. INTRODUCTION

HRI refers to the ways in which humans can communicate with robots. HRI is a fast-paced field, as it involves developing more robust methods of interaction. One approach to accomplishing this is by increasing robot autonomy, or the percentage of time the robot operates without direct input from the operator. The ideal robot is able to receive instructions as naturally as a human can, as this provides the most intuitive interface. However, the downside to increasing autonomy is dealing with the potential for error in robot perception. An interface needs to be developed so that robots do exactly as the operator requests, without burdening the operator with cumbersome low-level commands. This system surmounts the challenges of introducing autonomy in robots, without sacrificing the accuracy presently offered by low-level interaction.

Robotic systems play an integral role in industries stemming from medicine to military technology. Significant resources are appropriated to control interface training and telecommunication with devices that are dependent on operators. For instance, the soldier at the control terminal in Figure 1 must focus wholly on operating a robot rather than completing another team-related task. Robots performing operations in the field are controlled and observed by military personnel. Robotic autonomy is essential in a military setting, where robots are utilized in numerous surveillance and combat scenarios.

Intuitive and natural interaction between robots and operators increases efficiency and effectiveness in the field. Since verbal commands and physical gestures are already used to convey information on the field, communicating with robots in a similar fashion is ideal. As opposed to handling low-level maneuvering, higher level commands provide a sleeker method of control. A combination of both developments improves and expands the use of robotic agents in various environments.

The software framework consisting of Sound Propagation and Perception for Autonomous Agents in Dynamic Environments (SPREAD) [10] and Agent Development and Prototyping Testbed (ADAPT) [2] built on Unity [13], a game engine and Integrated Development Environment, has been developed at the Human Modeling and Simulation Lab (HMS) at the University of Pennsylvania. This framework offers a unique opportunity to simulate autonomous agent interactions in a definable world. This multifaceted tool can provide an ideal platform for HRI development and evaluation.

This simulation environment is utilized to help tackle the challenge of interacting with robots. First, an android application is designed to read in gestures and mapped audio input from the user. Then the input data is parsed and interpreted by the interface. Finally, the output low level commands are conveyed to the simulated robot. The simulated robot then navigates through the tactical situation that it is presented. By establishing a scenario that a military robot could potentially find itself in, the performance of such an interface can be effectively evaluated. The two major contributions are an application that receives multimodal commands and triggers autonomous behaviors and a simulation environment that demonstrates its capabilities.

Section 2 provides background information pertaining to this HRI system. Section 3 describes the details of the system model. Section 4 explains the design decisions behind
the implementation of this system. Section 5 presents the results obtained by evaluating the system. Section 6 provides details about ethical issues that could arise from the use of the system. Section 7 offers ideas on future development. Section 8 relays the conclusions reached after implementing this HRI system.

Figure 1: Teleoperated military robot [2]

This approach to improving HRI systems is unique in that it relies on the environment to make autonomous decisions in combat situations. Rather than dictating lower level commands to robots, the focus is on simulating a robot that can act as a member of the unit rather than a cumbersome tool. This is accomplished by modeling the robot's perception of sound and visual cues on existing software framework in Unity.

2. RELATED WORK

Extensive work has been done in the area of human robot interaction. A detailed inter-disciplinary study by Burke et al. [4] of HRI reveals research issues in this area to be considered. The study highlights the need for representing situational context as well as social models in HRI. In addition, Yanco and Drury [14] describe a classification of HRI concepts. These classifications have provided a basis for the development of an HRI system. Task type is applicable to developing a control interface as it describes a task a robot is expected to complete. In this HRI system, the task type requires an interface that can allow the operator to complete other physical tasks simultaneously. Another aspect mentioned is task 'criticality', a term to describe how critical a task is. In military applications the criticality of the task would be high, as the robot must perform exactly as the leader instructs. The ratio of people to robots is also stated as an important area to consider. The interface used in this system is required to respond to multiple agents in the field. Another relevant section of the taxonomy discussed here is the level of autonomy and intervention, the fraction of time the robot independently operating and the fraction of time the human is controlling the robot respectively. Considering these aspects have contributed significantly to the design of the interface.

Papers have also been written regarding the assessment of control modalities. Steinfeld et al. [12] discuss metrics evaluating HRI systems. Using navigation, perception, manipulation, social interaction, and performance as benchmarks, the effectiveness of HRI systems can be objectively evaluated. These metrics are used to evaluate the implementation of a control interface.

In preparation for the development of a military oriented robot control interface, existing interfaces used by the military have been considered. Teleoperation, controlling a machine from a distance, is often the method of HRI used by military units [5]. Though it provides accurate and predictable interaction, teleoperation requires the attention of one or more individuals in order to operate a robotic asset. The solution presented here improves on this technology.

Control modalities are another important aspect in this area of study. Gesturing is one approach that has been explored [7]. In this approach, accelerometer values from a smart phone were mapped to low level commands given to the robot. Using gestures as a means of communicating with robots on the battlefield falls in line with the gestures already used in field commands. A user study [7] was conducted in which subjects controlled the robot by physically gesturing with the smart phone. It was concluded that gestures did not provide the best mapping to low level commands as the gestures required intensive input by the user in order to complete navigation tasks. An alternative solution to this issue is mapping gestures to higher level commands, which is explored later.

In addition to singular modalities, there has also been research on multiple modalities in HRI. An existing application of Multimodal HRI that has been developed involves the utilization of the Augmented Reality Interface for Teleoperation (ARITI) software framework [3]. Multiple forms of input are combined in order to provide a more efficient output. This allows the operator to use multiple devices to interact with the robot, without being occupied by the details of device management. The Smart Core, which unifies incoming data from multiple inputs, is tasked with handling low level objectives, such as moving a robot arm. This provided motivation for the system design discussed later for handling multiple control modalities.

Correa et al. [6] explore multimodal HRI in a system developed to allow supervisors to command military forklifts through tablet sketch and speech commands. In this framework, the robot was directed to lift and transport cargo, as well as report information back to the supervisor. These actions were implemented as high level commands that allow the supervisor to take control if something went wrong. This involves significant interaction with the environment, as the safety of nearby humans has to be considered. The requirements of this interface mirrors that of a military field robot and is a model for this level of criticality.

Another existing application of multi-modal interaction created by Holzapfel, Nickel, and Stiefelhagen [8] utilizes speech and 3D gestures monitored by video in order to command a robot. Algorithms were used to parse audio data and interpret visual gestures. The evaluation of the architecture, which consisted of giving a humanoid robot a series of commands, showed that the combination of visual gestures and speech reduced false-detection of gestures. This methodology was built upon in the implementation presented hereafter.

The system discussed henceforth expands upon these ideas of autonomy and multi-modality. The multi-modal communication interface is evaluated by performing a user study.
on a mission in a simulation environment and collect a set of quantitative metrics. In this way it simulates the ease of interaction in a realistic environment and more effectively measure the performance of the communication interface. Utilizing Frazier’s [7] user study of low-level commands as a base, it is determined whether higher level commands provide better functionality. The study’s evaluation of gestures commands is employed to eliminate errors that can occur during gesture recognition. The SPREAD framework developed by Badler, Huan, and Kapadia [10] is used to represent the dispersion and attenuation of audio signals.

3. SYSTEM MODEL

This system consists of the development of a combination of communication modalities including audio commands and gestures to provide an intuitive interface for sending commands to a semi-autonomous robot. A set of missions has been designed for a human-robot team in a virtual marketplace setting where the soldier team is tasked with patrolling the region as shown in Figure 2 to determine if there is any suspicious activity. The lead soldier is controlled by a human user using a traditional control interface such as directional keys. In addition, the human user issues commands to the semi-autonomous robot using gestures and audio triggers. Gestures are captured using a smart phone interface [7]. The human user issues audio commands by selecting from a set of predefined audio cues that are propagated in the virtual environment and perceived by the robot. These human commands map to high-level behavior directives for robot control.

The system consists of an application for handling audio and gesture-based commands. This application allows for the management of communication modalities. A human user has the ability to perform predefined audio commands along with gestures in order to manage the behavior of the military team as it attempts to accomplish a mission. In order to facilitate this communication, the application first requires an interface for the Android device, which processes the user’s input. Then, this data is transmitted to Unity via Bluetooth where it is parsed and interpreted.

The goal of the interpreter is to provide the best response to user input, while still enabling the robot to respond to environmental audio cues. The interpreter receives this information in the form of signed bytes and manifests them within the simulated environment. The interpreter receives the user’s selection of targeted agent and communication modality and completes these selections within the scenario manager. Gesture commands are propagated into the scenario as gestures given by the user’s character. Audio commands are manifested as simulated sounds within the scenario.

In addition to this interface, this system includes a simulated environment created using the ADAPT simulation platform. It is made to resemble a populated marketplace, which includes various independent virtual agents such as buyers and vendors. The capability of the robot agent is of primary significance. The motivation behind the various missions the robot must complete is to demonstrate that a robot agent can independently and correctly interpret high-level commands from a human agent without hindering the performance of the team.

This system involves a series of modules ranging from recognition to implementation that work in conjunction to provide both an intuitive interface for the user and increased autonomy in the simulated robot. The success of the system relies on the interpretation of the user input.

4. SYSTEM IMPLEMENTATION

4.1 Interface

The Android smartphone interface was designed to issue multi-modal commands to the simulated robot and is operated using buttons on the touchscreen. In more detail, the user is presented with the options of selecting a receiver (Robot or Soldier) and modality (Sound or Gesture). For example, if he selects ‘Robot’ as the receiver and ‘Sound’ as the modality, he can choose a sound signal to relay to the robot in Unity, which initiates the signal through the SPREAD interface. In addition, the user may select ‘Gesture’ as the modality in which case he can perform a gesture, which is then mapped to a command as shown in Figure 5. Gesture commands, processed by the Android Gesture Recognition Framework on the smartphone application [9], are propagated to the virtual model in the scenario. The user can then tilt the smartphone, making use of its accelerometer, in order to position a target in the Unity marketplace environment. Screenshots from the smartphone application interface can be seen in Figure 4.

![Figure 5: Depiction of the 'follow' gesture.](image)

This communication is achieved by transmitting the data from the Android device, which outputs the signed byte result of the user’s commands, via Bluetooth ports to Unity, which is running on a Bluetooth compatible machine. A script within the Unity marketplace simulation then translates this incoming input from the smartphone based upon predetermined mappings to high level commands for the agents(s) to execute in the marketplace. The system makes use of the open-source Android application Cellbots [1], which provides functionality for collecting user targeting input.

The output of the interpreter is relayed to the simulation running in Unity, which is used as a testing environment for displaying the effectiveness of capturing and performing audio and gesture-based commands. In order to materialize these low level commands into an action in the marketplace simulation, it is necessary to use the ADAPT framework. ADAPT is an open-source software framework developed at the HMS lab. It provides tools for creating complex virtual environments and functionality for pathfinding, steering, and character animation. This is essential for simulating virtual humans and robots [2]. Low level commands output...
by the interpreter are mapped to ADAPT functions that allow the robot to move around and pick up items in simulation. Figure 6 contains a layout of how robot and multi-actor behaviors are implemented through ADAPT. By using behavior trees a series of these low level instructions is assembled for autonomous behavior.

4.2 Behavior Authoring

To maintain the high-level behavioral repertoire of the robot and other virtual agents, the use of behavior trees has been essential. The ADAPT simulation platform is augmented by procedurally generating a diverse virtual populace and author complex avatar behaviors using Parameterized Behavior Trees (PBTs) [11]. PBTs support authoring multi-actor behaviors in a single tree structure, which greatly reduces the authoring burden for more complex interactions between multiple actors. An event-centric authoring paradigm facilitates complex multi-actor interactions without the need for heavy scripting. A user-controlled soldier-robot team is introduced into the simulation which elicits appropriate behavioral responses in the populace and trig-
<table>
<thead>
<tr>
<th>Behavior</th>
<th>Description</th>
<th>Communication Modality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look Here</td>
<td>Face the user</td>
<td>Periodic Sequence Sound</td>
</tr>
<tr>
<td>Follow Me</td>
<td>Follow the user</td>
<td>Impulse Sequence Sound</td>
</tr>
<tr>
<td>Stay Here</td>
<td>The robot stops moving</td>
<td>‘Swing Down’ Gesture</td>
</tr>
<tr>
<td>Follow Me</td>
<td>Follow the user</td>
<td>‘Upright Circular’ Gesture</td>
</tr>
<tr>
<td>Go To</td>
<td>Go to a point indicated by the user</td>
<td>‘Swing Left’ Gesture.</td>
</tr>
<tr>
<td>Investigate</td>
<td>Approach a point or object, take pictures, and return to the user</td>
<td>‘Swing Right’ Gesture.</td>
</tr>
<tr>
<td>PickUp</td>
<td>Pick up an item.</td>
<td>‘Slide Right’ Gesture.</td>
</tr>
</tbody>
</table>

Table 1: High level commands issued to robot

![Flow of the smartphone application](image)

Figure 4: Flow of the smartphone application.

gers events which are authored to define specific training missions. In Figure 9(a) a simple behavior tree for the robot is illustrated. Evaluation begins at the root node, which in this tree is a sequence node symbolized by a dashed arrow. While other types of transition nodes exist, sequence nodes continue performing each action in order from left to right until an action fails. At that point, the tree has finished execution. So, the first behavior the robot performs is to go to the location of the designated object. Then, it takes a picture of the object and rotate a certain amount of degrees in order to position itself for a picture from another angle. Upon successful completion of investigating the object, the robot attempts to perform Follow, which determines the distance the robot is from the team leader. If this distance is sufficiently large, meaning Follow returns successfully, the robot returns to the team leader. Finally, it enters a loop in which it stays in its current location until it is directed to do otherwise. A list of the behaviors that have been implemented in the fashion can be seen in Table 1.

4.3 Event Coordination

In order to author complex missions for the soldier-robot team to perform in the marketplace, a mechanism is needed to trigger event trees in a dynamic fashion based on the presence and interaction of the soldier-robot team with the populace. To achieve this, a centralized event coordinator is utilized. It automatically instantiates behaviors and events in the populace to conform to the specific mission directives. A mission director defines a mapping of events which must occur when certain conditions in the simulation are satisfied. These conditions include: (1) user input, (2) the spatial location of the soldier and robot, (3) when the soldier or robot performs a particular interaction (e.g., investigate suspicious vendor stall), (4) temporal conditions, or (5) the success or completion of ongoing events.

![Generic mission definition](image)

Figure 7: Generic mission definition.

Figure 7 illustrates a generic mission definition. Events are hierarchical control structures rather than scripted action sequences, and may succeed or fail based on the actions of the user controlled soldier-robot team. This ensures that
every run of the same mission produces a different simulation where the outcome is contingent upon user input, facilitating a dynamic simulation environment which exercises the soldier-robot interactions in a variety of different situations, without the need for heavy authoring.

4.4 Sound Propagation

In order to simulate sound perception for the robot agent, the SPREAD model [10] for sound propagation was utilized. The SPREAD model represents audio data as sound packets. Audio data is segmented into packets by a short-time Fourier Transform. The packets disperse from that source and attenuate as a result of distance and other environmental factors. Figure 8 depicts a visual manifestation of the SPREAD model. Audio commands are pre-recorded non-linguistic audio triggers that are mapped to behavior primitives. The simulated robot not only observes vocal commands given by agents, but also observes and reports on observational sounds. For example, if the robot is in the alleyway ahead of the military unit and it hears hurried footsteps quickly diminishing in volume, it is able to report back to the military personnel that it hears someone running. This autonomy provides useful insight in this environment.

4.5 Soldier-Robot Team

The soldier-robot team consists of a team captain, a robot, and one or more other soldiers, who are tasked with carrying out a specific mission in the given environment.

**Team Captain.** The team captain is controlled by a trainee subject using a traditional keyboard and mouse interface. The user controls the movement of the team captain using a traditional WASD control scheme. The commands issued by the user result in appropriate animations on the subject’s avatar: the team captain performs a gesture to indicate a particular command or points at a specific location or object to go to.

**Robot.** The robot is a semi-autonomous agent. It monitors commands issued by the team captain which then trigger appropriate behaviors. A directional command interface where the user has fine-grained control of the robot is very burdensome, preventing the user from carrying out additional tasks simultaneously. The robot is equipped with autonomous capabilities to independently carry out high level commands. Table 1 defines a set of behaviors that we defined for demonstrating the missions described in Section 4.6. Additional behaviors can be easily added using the authoring interface described earlier.

**Other soldiers.** The other soldiers can be controlled (1) directly by other users using a similar control interface, (2) indirectly by defining an extended set of behaviors that can be triggered by the team captain’s commands, or (3) by conditions in the environment.

Additional specializations can be easily introduced by defining new behavior trees that are instantiated when their trigger condition is satisfied. The villagers are made to react to the presence of soldiers and robot by introducing a set of appropriate behaviors that are triggered when the villager is near them. For example, we introduce an Avoid behavior where villagers maintain a safe distance from the passing robot or soldier. Alternatively, villagers might be happy to see the soldiers and approach them and initiate a conversation. The presence of the robot might invoke the curiosity of the villagers by capturing their attention or even having them follow the robot for a short distance. Table 2 enumerates some behaviors that were authored for villagers in the marketplace. Mission-specific behaviors where the villagers act in a particular manner (e.g., a crowd gathering or a procession) can be easily added without requiring any change to the existing behavior library. The supplementary document provides details of additional authored behaviors.

4.6 Missions

To demonstrate the effectiveness of our framework, missions were authored and demonstrated, exercising different types of interactions between the team captain, his soldier and robot team, and the populace. The mission with which the functionality of the interface is assessed involves a global event that guides the user throughout the mission. This global event directs the user to complete tasks that demonstrate the different features of the interface and simulation. The global event is a behavior tree that displays information for the user, waits for the user to complete tasks, and triggers events among the marketplace populace. For example, the user is tasked with directing the robot to investigate an alleyway. The behavior tree first calls a function that displays this instruction to the user. The tree will then wait until the robot has reached its target, before triggering the occurrence of a suspicious event. The tree will then wait until the suspicious event has been completed before prompting the user to react. The mission implemented by this global event tree demonstrates functionality, after which point the user is prompted to experiment within the scenario.

5. RESULTS

Upon completing the use case for this interface and testing environment, we began to measure the accuracy of the components. A series of test runs of the example mission were completed, during which time each input by the user was recorded. After the collection of 50 test inputs, the accuracy of the interface was defined by the percentage of instances where the robot had the correct response to the user’s input. Communicating to the robot using the gesture modality resulted in an accuracy of 80% by this definition. This reflects the ability of the gesture recognition software to recognize user input. This is also a result of the requirement that the robot should have the soldier agent within its visual range. Communicating with the robot using the audio commands resulted in an accuracy of 68% by the definition given above. This reflects the ability of the SPREAD...
Figure 9: Example Behavior Trees used for authoring behaviors in the marketplace.

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wander</td>
<td>Roam the marketplace, stop at stalls and look at goods of interest.</td>
</tr>
<tr>
<td>Conversation</td>
<td>Arrive at a common meeting point and have a conversation with other villagers. Different conversation behaviors can be authored to convey concord, or disagreement.</td>
</tr>
<tr>
<td>Haggle</td>
<td>Interaction between vendor and villager to agree upon a price to purchase goods.</td>
</tr>
<tr>
<td>Keep Safe Distance</td>
<td>Cautiously observe soldier or robot as it passes by and keep a safe distance.</td>
</tr>
<tr>
<td>Interact</td>
<td>Wave to passing soldier and strike a conversation.</td>
</tr>
<tr>
<td>Follow</td>
<td>Follow soldier or robot.</td>
</tr>
<tr>
<td>Disperse</td>
<td>Disperse from a specific area in the marketplace.</td>
</tr>
</tbody>
</table>

Table 2: Behaviors defined for the villagers in the marketplace.

framework to recognize the signal propagated based on the positioning of the characters within the virtual scenario.

Both of these results reflect the challenges faced when switching from a low-level, direct form of communication to a more intuitive and high-level one. The robot’s environment constantly affects its ability to hear the team leader within this situation. Though the user has less demanding means with which to communicate with the robot, the user has to consider the robot’s ability to process that command given its relative position. These results also demonstrate the usefulness of the SPREAD framework and the gesture recognition software in simulating user input. The scenario can reflect challenges with utilizing this interface in a real environment.

6. ETHICS

Since the original application of this system lies within a military setting, ethical issues could occur from its use. One issue involves an invasion of privacy as the robot could be used to surreptitiously obtain images of individuals who may or may not be guilty of any crime. Another issue could arise from the misinterpretation of commands to the robot leading to unintended consequences as a result of its actions. Naturally, this could occur because the robot has no sense of moral judgment as it executes commands. Essentially, the robot in this system suffers from the same flaws any machine without a concern for moral behavior has. These ethical challenges can be mitigated by including safeguards into the robot’s autonomous behaviors. Before a behavior is executed, the robot will first have to ensure that civilians around it will not be adversely affected.

7. FUTURE WORK

In the future, this system could be adapted to a physical robot for further experimentation with convenient control modalities. A robot with autonomous behaviors such as moving to a point and picking up items can be communicated to with the smartphone interface. A robot such as the PR-2 can be controlled via gestures and audio commands given through the smartphone.

The Unity environment described in this system provides a testing platform for the development of future human-robot interfaces. The provision of an interactive, functional, purposeful virtual populace is of paramount importance to completely immerse and engage human subjects in military training simulations and other serious game applications.

One technical challenge that could be further addressed includes incorporating natural language processing so that a user has the ability to deliver vocal commands rather than selecting from a list of predefined inputs.

8. CONCLUSIONS

This application and simulation environment are useful assets in the development of HRI. By simulating an au-
tonomous robot, interfaces using more convenient modalities can be created. The creation of a realistic interface aids in effective training and testing of human-robot interfaces. This framework provides a strong foundation for military training simulations and human soldier-robot team experiments, and can be easily adapted to other interactive applications. The system makes it possible to investigate important research directions including the selection of the right set of robot behaviors, and defining an appropriate tactical command vocabulary for human-robot interaction.

8.1 Contribution

- Development of an application that uses combination of communication modalities, namely audio commands and gestures, to provide an intuitive interface for sending commands to an autonomous robot.
- Authoring of a virtual marketplace scenario to facilitate the development of human robot interfaces.

9. REFERENCES