USARSim: Simulation for the Study of Human-Robot Interaction

Michael Lewis
University of Pittsburgh
Jijun Wang
University of Pittsburgh
Stephen Hughes
Roanoke College

ABSTRACT: The PackBots being used by the U.S. military in Afghanistan and the urban search and rescue (USAR) robots that worked the World Trade Center site are just two recent examples of mobile robots moving from the laboratory to the field. What is significant about these new applications is that they invariably involve some form of human-robot interaction (HRI) rather than the full robot autonomy that has motivated most prior research. Conducting HRI research can be extremely difficult because experimentation with physical robots is expensive and time consuming. Few roboticists have experience or interest in conducting human experimentation, and researchers in human factors or human-computer interaction often lack experience in programming robots or access to robotic platforms. In this paper, we describe a high-fidelity, open-source simulation intended for HRI researchers of varying backgrounds and provide reference tasks and environments to facilitate collaboration in order to share the results. The architecture and capabilities of the game engine–based USARSim simulation are described. Its use for HRI research is illustrated through case studies describing experiments in camera control for remote viewing and integrated display of attitude information.

Introduction

MOBILE ROBOTICS IS MATURING FROM AN AREA OF ACADEMIC RESEARCH AND "ONE-OF-A-KIND" special projects, such as interplanetary rovers or robots, to enter containments at Three Mile Island and Chernobyl to a small but growing number of applications used by the general population. As smaller mass-produced robots, such as the domestic Rhoomba or military Packbot, enter general use, many of the conventional assumptions about their operators and tasks no longer apply. The user interface, behaviors, and robot-exercising artificial tasks appropriate for a robot's creator are no longer acceptable when the user has scant knowledge of the robot's design or limitations and is focused on tasks for which the robot is a tool rather than the *raison d'etre*. This difficulty is compounded by the growing realization that many, if not most, of the important applications of robotics will involve some degree of human control.

Today, the emerging science of human-robot interaction (HRI) stands in approximately the same place that human-computer interaction (HCI) stood 30 years ago.

ADDRESS CORRESPONDENCE TO: Michael Lewis, University of Pittsburgh, 135 N. Bellefield Ave., Pittsburgh, PA 15260; ml@sis.pitt.edu. Visit the *JCEDM* Online Companion at http://cedm.webexone.com.

Journal of Cognitive Engineering and Decision Making, Volume 1, Number 1, Spring 2007, pp. 98–120

The makers and users of the technology are the same people. The user interface and interactions are of secondary interest and have not yet been the subject of sustained study. The technology has experienced great advances in utility and reliability and is poised to expand into a wider market.

The problems of HRI fall somewhere between the humancentric interests of human factors researchers and the automation-centric concerns of artificial intelligence (AI) and intelligent-agent researchers. Although there is a substantial body of knowledge about the difficulties of teleoperation and manual control (Sheridan, 1992), much less is known about methods for overcoming the limitations of the human perceptual system that severely handicap remote viewing operations. Even more interesting questions revolve around recent developments involving adjustable autonomy (Hearst, Allen, Horvitz, & Guinn, 1999) and mixed-initiative control (Dorais, Bonasso, Kortenkamp, Pell, & Schrekenghost, 1999), which combine human and automated behaviors.

Although the notion that a fully autonomous entity can replace human presence is appealing, human observation and supervision remain essential components of robotic activity. Contrary to the findings in traditional research in automation, operators will need to be prepared to deal with robots that assert their own agendas rather than strictly follow simple orders.

Finally, as the level of autonomy increases, it will be possible to simultaneously interact with multiple robots. In addition to magnifying the difficulties discussed previously, this will expose the operator to a whole new set of coordination issues for teams of robots.

Urban Search and Rescue

Urban search and rescue (USAR) is emerging as the canonical HRI research task. Unlike its companion demonstration domain, RoboCup soccer (Noda, Suzuki, Matsubara, Asada, & Kitano, 1998), USAR robotics has consistently emphasized human interactivity. Robotic soccer employs a flat, known, well-marked environment in order to focus research on robot coordination, planning, and execution, whereas USAR presents an unknown, obstacle-ridden environment that can be challenging to robotic exploration even with the best of human assistance.

In 1999, the National Institute of Standards and Technology (NIST) (Jacoff, Messina, & Evans, 2001) introduced physical USAR reference test arenas (Figure 1), which are "designed to represent, at varying degrees of verisimilitude, challenges associated with collapsed structures" (Jacoff et al., p. 259). The USAR arenas provided a controlled environment for comparing the effectiveness of different robotic designs, control and mapping algorithms, and team regimes. In 2002, the RoboCup Federation adopted these USAR arenas to host RoboCup Rescue league competitions (Jacoff et al., 2001).

The Yellow, Orange, and Red arenas pose search tasks of progressive difficulty ranging from an office-like environment to a difficult-to-navigate rubble heap. A much larger reference environment that was built around an abandoned Nike silo at NIST's Gaithersburg, Maryland, campus was recently added.

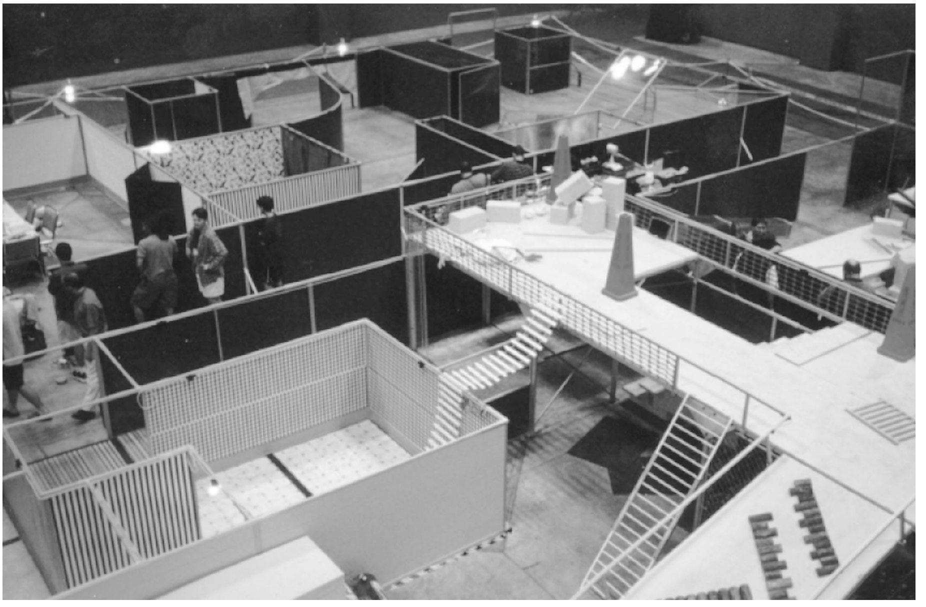

Figure 1. NIST Urban search and rescue arenas.

Although the arenas were originally designed to advance research in autonomous robotics, no team has yet successfully operated robots autonomously. Recent NIST efforts (Yanco, Drury, & Scholtz, 2004) have turned to identifying features of interfaces and HRI strategies that lead to successful performance. Experience with robots in actual USAR operations has been more limited and is primarily attributable to reports by Robin Murphy and her students' experiences at the World Trade Center (Casper, 2002; Murphy, 2003) in disaster response training exercises (Burke, Murphy, & Coovert, 2004), and in the USAR arenas (Casper, 2002).

Simulation and HRI

HRI is an excellent candidate for simulator-based research because of the relative simplicity of the systems being modeled, the behavioral fidelity possible with current physics engines, and the capability of modern graphics cards to approximate camera video. Many of the HRI studies recently reported have relied on simulation (Chadwick, Gillan, Simon, & Pazuchanics, 2004; Hughes & Lewis, 2004; Nielsen et al., 2004; Ricks, Nielsen, & Goodrich, 2004; Olsen & Wood, 2004; Wang, Lewis, & Hughes, 2004).

Researchers specializing in robotics who are outside computer science or mechanical engineering departments are unlikely to have access to the experimental robots needed to conduct research. Yet researchers from disciplines such as HCI, psychology, or human factors often have the greatest interest in issues affecting HRI. Even if robots and maintainers were made available to all interested HRI researchers, the inevitable problems with batteries, dropped leads, and misaligned sensors that accompany experimental robotics could contaminate data and endanger the replicability of results. Expense, unreliability, and difficulties in running participants in parallel – especially in multirobot experiments – make physical robotics inappropriate for the large samples, repeated trials, and varied conditions that are needed for HRI research. It is crucial that these conditions be tightly controlled because the object of study in HRI is the behavior of the human with the robot, environment, and task determining the experimental conditions. High-fidelity simulation is therefore not only the most practical but may also be the best-suited tool for HRI research.

The critical feature for HRI-oriented simulation is that it accurately reflects the range of available information, behavior, and user experience encountered in actual robot operation. This means an HRI simulation must supply both sufficient perceptual fidelity to make the operator realistically aware of the remote environment and sufficiently accurate modeling of the robot and its automation to relate robot behavior to that same environment. In terms of Terry Fong's (2001) example of a robot detouring around tall grass, the simulator must provide graphics sufficiently accurate for the operator to recognize tall grass and modeling of sensors and automation sufficiently accurate for the robot to display obstacle avoidance.

Although many robotic simulators are available, most of them have been built as ancillary tools for developing and testing control programs to be run on research robots. Simulators built before 2000, including those by Konolige and Meyers (1996) and Lee, Ruspini, and Khatib (1994), typically have low-fidelity dynamics for approximating the robot's interaction with its environment. More recent simulators, including ÜberSim (*http://www.cs.cmu.edu/~robosoccer/ubersim*), a soccer simulator; Gazebo (Gerkey, Vaughan, & Howard, 2003); and the commercial Webots (*http://cyberbotics.com*) use the open source, Open Dynamics Engine (ODE) physics engine to approximate physics and kinematics more precisely. ODE, however, is not integrated with a graphics library, which forces developers to rely on low-level libraries, such as OpenGL. This limits the complexity of environments that can practically be developed and effectively precludes use of many of the specialized rendering features of modern graphics processing units. Both high-quality graphics and accurate physics are needed for HRI research because the operator's tasks depend strongly on remote perception (Woods, Tittle, Feil, & Roesler, 2004). Remote perception requires accurate simulation of camera video and interaction with automation, which requires accurate simulation of sensors, effectors, and control logic.

USARSim

USARSim is a high-fidelity simulation of urban search and rescue robots and environments intended as a research tool for the study of HRI and multirobot coordination. USARSim supports HRI by accurately rendering user interface elements (particularly camera video), accurately representing robot automation and behavior, and accurately representing the remote environment that links the operator's awareness with the robot's behaviors. The current version of USARSim consists of environmental models (levels) of the NIST Yellow, Orange, and Red Arenas and the Nike site, which serve as standardized disaster environments for mobile robot studies, robot models of commercial and experimental robots, and sensor models. USARSim also provides users with the capability to build their own environments and robots. Its socket-based control API enables users to test their own control algorithms and user interfaces without limitations in programming language or the operating system.

A Game Engine–Based Design

Typically, real-time "out the window" or "through the camera" simulations have been difficult, time consuming, and expensive to build, requiring specialized hardware and personnel. The cost of developing such simulation has grown so huge that even in the gaming industry, developers can no longer rely on recouping their entire investment from a single game. This has led to the emergence of game engines – modular simulation code – which are used for families of similar games. Separation of game logic and rules from simulation dynamics and environmental data allows the core code to be reused for more general simulation. Aside from affordability, today's game engines also offer advanced graphical displays, realistic environments, accurate physics, and dramatically reduced development times (Lewis & Jacobson, 2002).

USARSim uses Epic Games' UnrealEngine2 (*http://www.unrealtechnology.com/html/homefold/home.shtml*) to provide a high-fidelity simulator at low cost. Unreal is one of the leading engines in the so-called first-person shooter genre and is widely used in the gaming industry. It is also gaining a strong following in the academic community as more researchers use it in their work. Recent academic projects have included creating virtual reality displays (Jacobson & Lewis, 2005), studying AI techniques (Magerko, Laird, Assanie, Kerfoot, & Stokes, 2004), studying pedagogical agents (Johnson et al., 2004), and creating synthetic characters (Best, Lebiere, & Scarpinatto, 2002).

In addition to the egocentric perspective, there are several other features of UnrealEngine2 that make it particularly appealing for HRI research, including cost, fidelity, and wide availability. Figure 2 shows UnrealEngine components and the expandable library of robot-themed models, environments, and control interfaces added to create the USARSim simulation.

Environments and Models

Robot models. USARSim currently provides detailed models of 13 robots. These include widely used research robots: the Activmedia Pioneer P2AT and P2DX, iRobot ATRV-Jr (*http://www.activrobots.com*), Sony Aibo and QUIO, the commercial Foster-Miller Talon, and a variety of experimental robots that were developed for USAR research. Figure 3 shows some of these simulated and real robots.

These models were constructed by building the components of the robot and defining how these parts were connected using joints that serve as mechanical primitives for the Karma physics engine. Because the physics engine is mechanically accurate, the resulting movement of the aggregate robot is highly realistic. Karma uses a variety of computational strategies to simplify, speed up, and exclude noninteracting

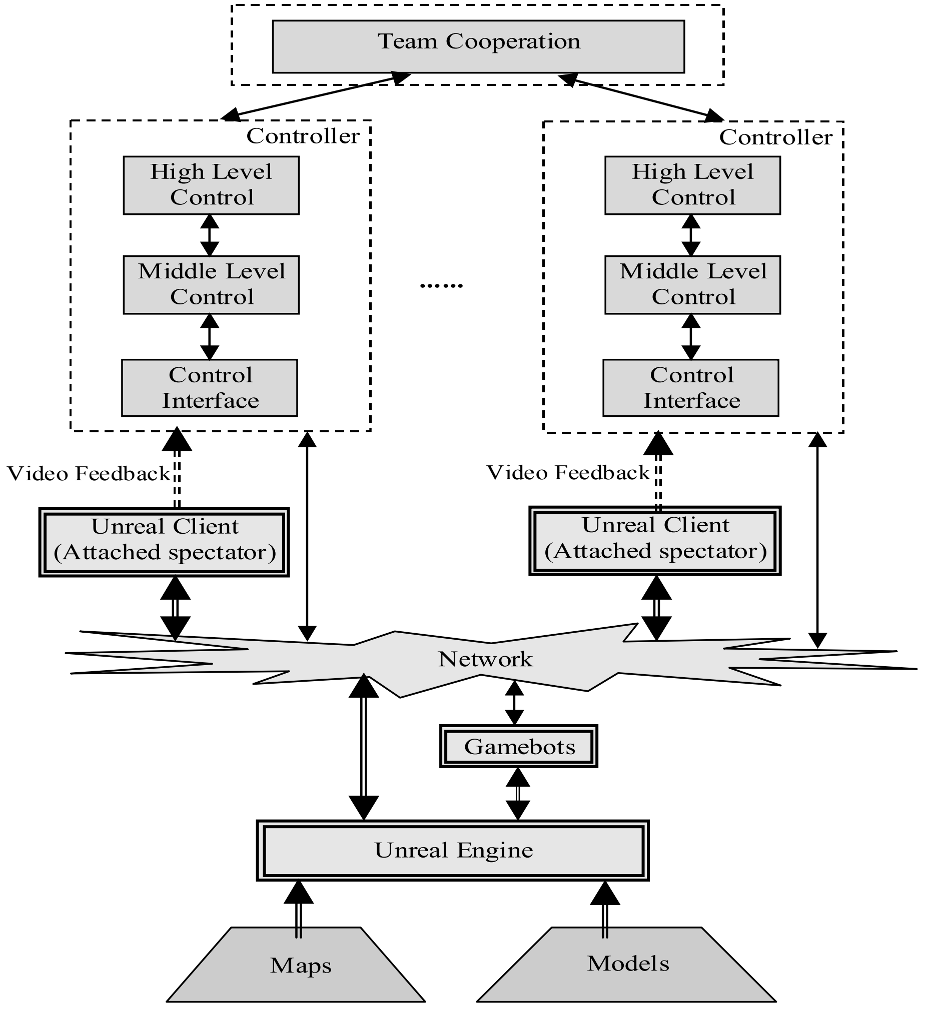

Figure 2. System architecture.

objects to achieve animation-level speed without sacrificing physical fidelity. Although current physics processing is sufficient to accurately simulate wheeled robots, it begins to break down as the number of constrained degrees of freedom increases. A scrum of five Aibos, for example, pushes the current engine to its limits (Zaratti, Fratarcangeli, & Iocchi, 2006). To simulate tracked robots, such as the Tarantula, an array of wheels must be substituted for the tracks. The next version of the game engine (UnrealEngine3), due in the summer of 2007, should resolve these problems by incorporating hardware-accelerated physics (Ageia Technologies, 2006) that are expected to improve physics processing by two to three orders of magnitude.

USAR environments. USARSim includes detailed models of the NIST reference

Figure 3. Some robots in USARsim (actual and simulated).

arenas and a replica of the fixed Nike site reference environment. A significantly larger disaster environment, including both indoor and outdoor areas, was built for the Virtual Robot USAR competition at RoboCup 2006 in Bremen. Another large environment will be made available following RoboCup 2007, which is to be held in Atlanta on July 1–10. To achieve high-fidelity simulation, Pro/Engineer (PROE, *http://www.ptc.com*) models of the real arenas were imported into Unreal and decorated with texture maps that were generated from digital photos of the actual environments. This ensures geometric compatibility and correspondence between camera views from the simulation and the actual arena. A collection of virtual panels, frames, and other parts used to construct the portable arenas are included with USARSim. Using the UnrealEd tool, it is possible to rearrange these elements to quickly develop alternative USAR layouts similar to how the arenas are reconfigured during USAR contests.

Control Interfaces

On the other side of the UnrealEngine are the control interfaces that allow the operator to direct and observe the robot's behavior. USARSim was developed as a tool for observing and testing alternative designs for this automation and its related user interface. The basic concept is that the simulation provides accurate models of robots, environments, and camera video while experimenters bring their own interface and automation strategies to be tested. Figure 4 shows an *example interface* that comes with the simulation. The example interface was designed as an illustration of the sensor data and control options that are available and the code illustrating how

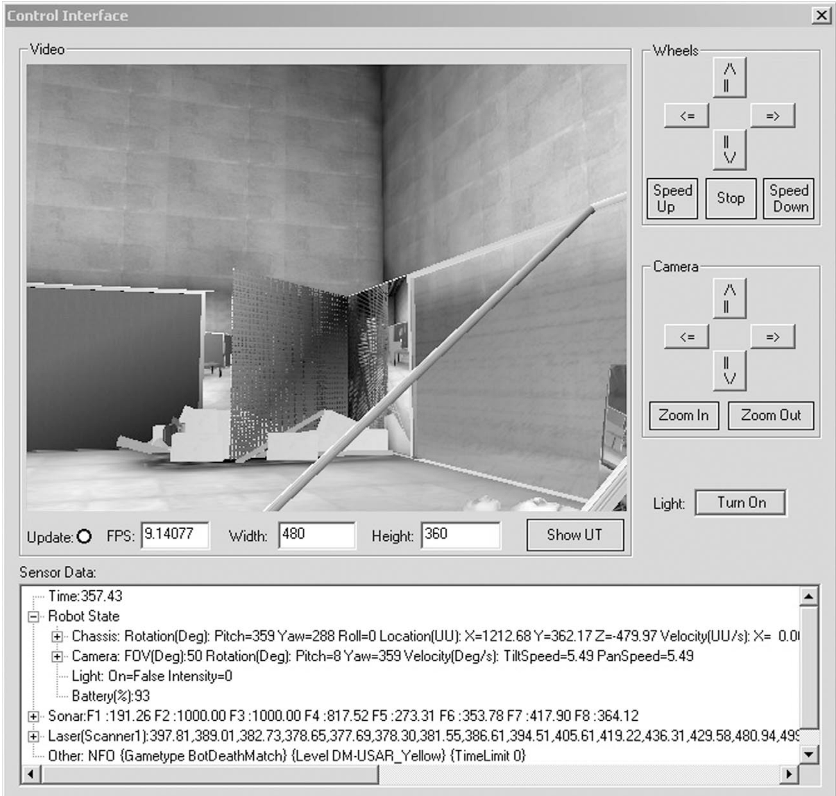

Figure 4. A sample interface researchers can use to integrate their interfaces and control logic.

these features may be accessed. The camera window demonstrates the image server's facilities for selecting field of view, window size, and frame rate. The controls on the right demonstrate control over the robot's movement and camera. Available sensor data can be accessed through the exploding lists at the bottom of the panel.

Sensor models. Sensors are a critical part of HRI because they provide the basis for simulating automation and also link the operator to the remote environment. USARSim simulates sensors by programmatically manipulating objects in the UnrealEngine. For example, sonar and laser sensors can be modeled by querying the engine for the distance given the sensor's position and orientation to the first object encountered.

To achieve high-fidelity simulation, noise and data distortion are added to the sensor models by introducing random error and tailoring the data using a distortion curve. Three kinds of sensors are simulated in USARSim:

- Proprioceptive sensors, including battery state and headlight state

- Position estimation sensors, including location, rotation, and velocity sensors
- Perception sensors, including sonar, laser, sound, and pan-tilt-zoom cameras

USARSim defines a hierarchical architecture to build sensor models. A sensor class defines a type of sensor. Every sensor is defined by a set of attributes stored in a configuration file. For example, perception sensors are commonly specified by range, resolution, and field of view. To get a sensor with specified capability, one can either directly configure a sensor class or derive a new sensor from an existing sensor class. Once the sensor is configured, it can be added to a robot model simply by including a line in the robot's configuration file. A sensor is mounted on a robot specified by a name, position where it is mounted, and the direction that it faces.

Simulating video. Cameras provide the most powerful perceptual link to the remote environment. The scenes viewed from the simulated camera are acquired by attaching a *spectator*, a special kind of disembodied player, to the camera mount on the robot. USARSim provides two ways to simulate camera feedback. The most direct is to use the UnrealClient as video feedback, either as a separate sensor panel or embedded in the user interface. Although this approach is the simplest, the UnrealClient provides higher frame rates and image quality than are likely to be achieved in a real robotic system, and its images are not accessible to the image processing routines often used in robotics. The second method involves intermittently capturing scenes from the UnrealClient and using these pictures as video feedback – an approach that is very close to how a real camera works.

USARSim includes a separate image server that runs alongside the UnrealClient. This server captures pictures in raw or JPEG format and sends them over the network to the user interface. Using this image server, researchers are able to tune the properties of the camera, specifying the desired frame rate, image format, and communication properties such as distortion or noise to match the camera and conditions being simulated.

Integration with other tools. Although USARSim was conceived primarily as an HRI research tool that would enable users to test their own automation logic and user interfaces, we have also integrated it with other widely used robotic research tools to increase its usefulness. In the USARSim package, we provide integration with the popular Pyro (*http://pyrorobotics.org*) and Player (*http://playerstage.sourceforge.net/player/player.html*) robotic middleware. These utilities provide an abstraction layer allowing control programs to be written that run across robotic platforms or in simulation. We also provide integration with CaveUT (Jacobson & Lewis, 2005), panoramic multiscreen display software, and UTSAF (Prasithsangaree, Manojlovich, Hughes, & Lewis, 2004), a bridge to DIS-based simulations. Both tools use modifications to the UnrealEngine that were developed in our laboratory.

Public software. Developed for undergraduate instruction in robotics, Pyro consists of a Python library, environment, graphical user interface (GUI), and low-level drivers. It abstracts robot control to a high level to enable students to build platform-independent robot programs. USARSim is wrapped as a Pyro plug-in to integrate with the system. Services such as sensor visualization and camera control are provided through the USARSim plug-in. The left picture in Figure 5 shows an example of controlling USARSim through an earlier version of Pyro.

Player is a network server for robot control that provides an interface to the robot's sensors and actuators over the TCP/IP network. Player is a research-oriented tool that is substantially more complex than Pyro but provides platform independence through abstraction in much the same way. The virtual device drivers provided in USARSim driver set include position, sonar, laser, and ptz (camera) drivers. The pictures are at the right in Figure 5.

Validation

From an engineering perspective, the validity of a simulation can be measured in terms of the completeness of the model. The structure of the environment is deemed accurate if all the components are present and proportionally scaled. Likewise, physical properties of the environment and the robots can be modeled to behave consistently with the laws of physics, allowing other objective measurements. However, building simulated robots and environments from specification sheets is only the first step in assuring a simulation's fidelity. Ultimately, consistent operator behaviors and overall user experiences will determine the effectiveness of a simulation for HRI studies.

Informally, we observed that the tasks that caused difficulties in the real environment also caused problems in the simulation (Wang, Lewis, & Gennari, 2003).

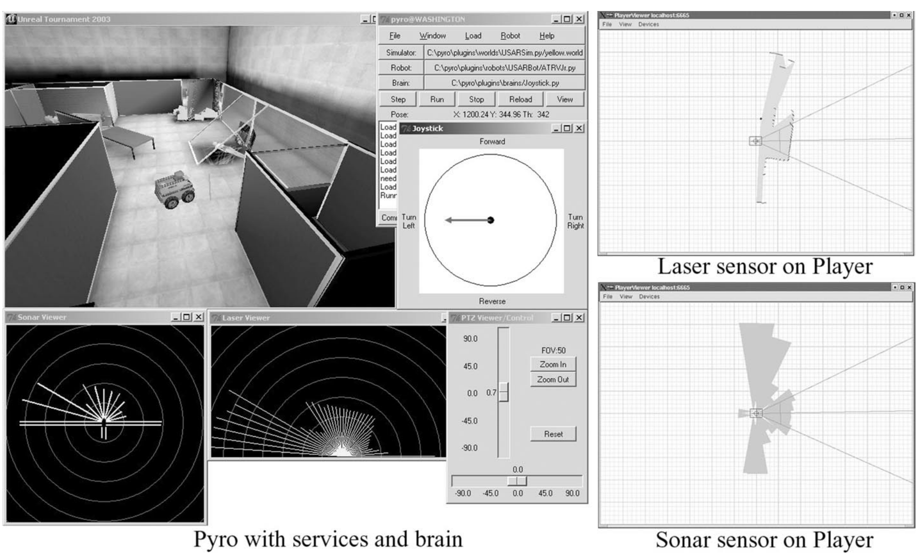

Figure 5. Pyro and Player with USARSim.

It is very difficult, for example, to drive the Corky robot up the ramp of either the real or the simulated Orange Arena. We are conducting more systematic tests to document the behavior of the simulation and explore the extent to which human control corresponds between the simulated and real robots. Initial tests have been completed for the PER (Personal Exploration Rover) and are in progress for the Pioneer P2-AT (simulation)/P3-AT (real) robots.

Providing a comprehensive validation of human behavior with USARSim presents a complex problem because any attempt is necessarily limited by the particular interface and control opportunities offered to the operator. For example, human operators may unwittingly compensate for environmental factors that would derail automated systems.

Realizing the impossibility of complete validation, we are relying on a two-stage approach. In the first stage, we conduct tests comparing the performance of elementary behaviors and sensor readings for real and simulated robots. In the second stage, we compare standard HRI tasks for particular interfaces and definitions of automation. Positive results give us some assurance that the simulation is physically accurate (Stage 1) and evidence that it remains consistent for at least some interface and automation definitions (Stage 2).

Stage 1 evaluation of the PER included collecting baseline measures to establish times, distances, and errors associated with movements from point to point over three types of debris (wood floor, scattered papers, lava rocks). These data were used to adjust the speed of the simulated PER and refine its performance when moving over these surfaces. The corresponding Stage 2 testing consisted of a 2 × 2 experimental evaluation. We assigned 20 paid participants to manipulate either the physical robot or the simulation using either direct teleoperation or waypoint (specified distance) interfaces.

In the initial three exposures to each debris field, participants had to drive approximately 3 m along an unobstructed path to an orange traffic cone. Hypothetically, this could be achieved with a single command. However, operators had to counteract the effects of the debris, which could delay or alter the heading of the robot.

In three subsequent trials, obstacles were added to the environments, forcing the driver to negotiate at least three turns to reach the objective destination (see Figure 6 for a visual comparison between the real and simulated video feed).

Although the data collected from five subjects per condition is insufficient for statistical analyses, it does provide a qualitative picture of the simulation's performance.

Terrain effects. The effect of the paper surface on the PER's operation was minimal. In Stage 1 testing, we found that after 30 trials on a 1-m straight course, the PER was typically within 5 cm of the intended destination. Alternatively, the rocky surface had a considerable impact; participants fell short of the goal by an average of 40 cm.

The relative performance trends on these surfaces were also found in Stage 2 results, reflected by increases in the odometry (shown in Figure 7) and number of turn commands issued by the operator. A parallel spike in these metrics is recorded in the simulator data, suggesting a similar experience.

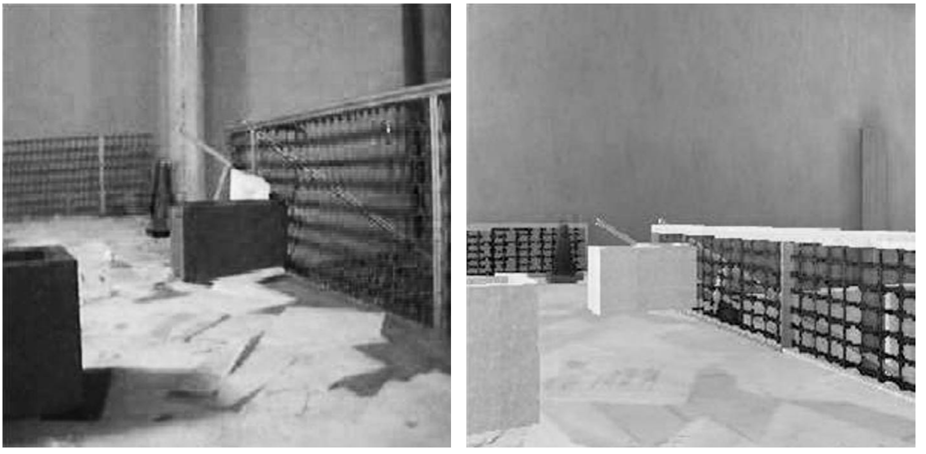

Figure 6. Real (left) and simulated (right) obstructed paper scenarios.

Learning rates. The participants in the study were novice operators who had received only a brief description of the robots' capabilities and performed a quick pretrial test drive. Similar learning trends on repeated tasks were observed between the simulation and the real environment. Participants generally improved performance across a range of metrics from the first trial in a debris field to the third trial.

Figure 8 shows these trends for the idle-time percentage. This metric tracks the lapse of time between commands and offers particular insight into the operator's

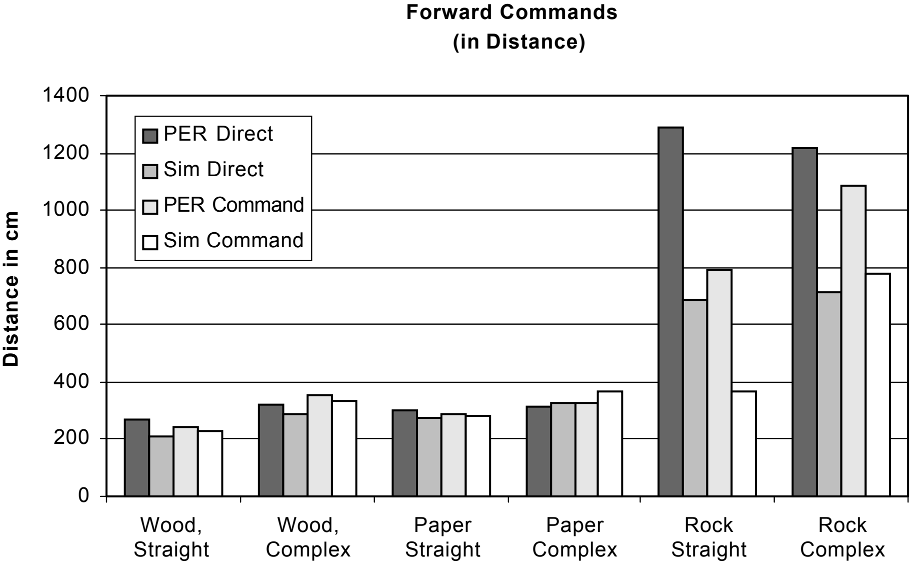

Figure 7. Odometry for Stage 2 validation.

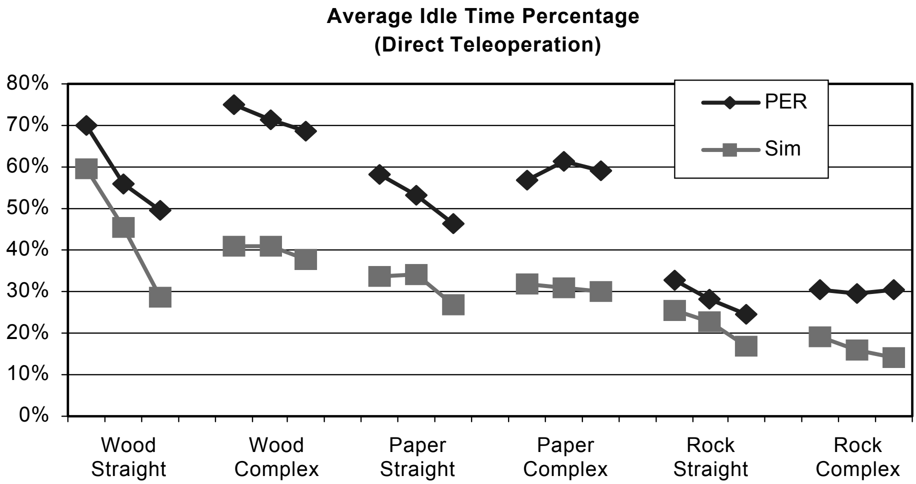

Figure 8. Between command times for Stage 2 validation.

decision making. Given that the simulation was set to match the frame rate for a physical camera, the time between commands can most likely be attributed to the operator's processing the changes in the display and deciding on the next command.

It is encouraging to see the idle time decrease within each set of trials, as well as an apparent downward trend throughout the course of the entire session. This indicates that the operators became more familiar with the control of the robot and the simulation.

Time to complete. Analysis of the average time to execute the task reveals the expected result that the complex environments took longer to complete than their straight, unobstructed counterparts. As shown in Figure 9, the command mode interaction seemed to be hampered more by the complex environment than direct teleoperation. We suggest that this is caused by the operator's burden of distance estimation, which becomes a more integral part of the task in complex settings. This difference highlights the impact of the interface on the validation process, and we are encouraged to see comparable shifts in performance from the simulation along this metric.

Proximity. One metric in which the simulation and the physical robot consistently differed was the proximity to the cone acquired by the operator. Participants were given the instruction to "get as close to the cone as possible without touching it." Operators using the physical robot reliably moved the robot to about 35 cm from the cone, whereas the USARSim operators were generally within 80 cm of the cone. It seems unlikely that the simulation would have elicited more caution from the operators, so this result suggests that there could be a systematic distortion in depth perception, situation awareness, or strategy. In both cases, the cone filled the camera's view at the end of the task.

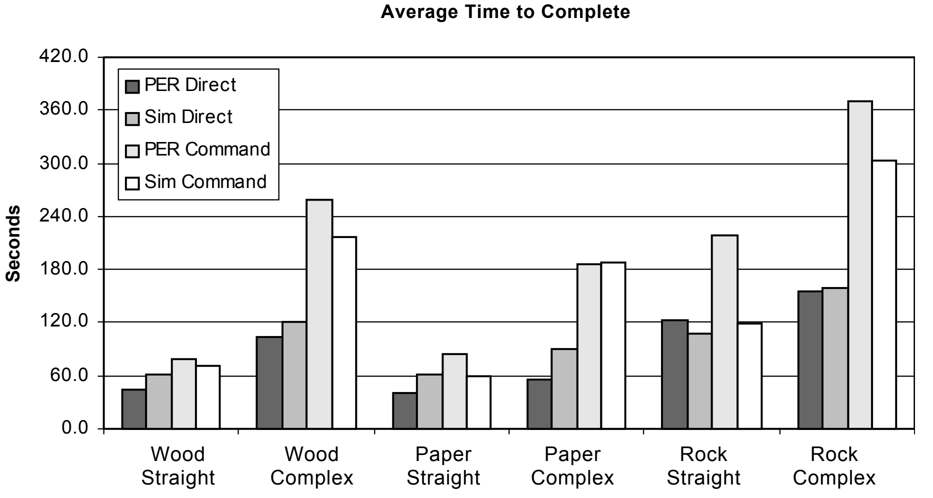

Figure 9. Average completion time for Stage 2 validation.

To draw valid conclusions from robotic simulations, it is important to know the metrics that are consistent with the operation of the actual robot and those that are not. By collecting validation data for all entities within the simulation, we hope to create a tool with which researchers can select manipulations and metrics that are likely to yield useful results. From this initial evaluation, for example, one might expect that debris would affect interaction with the PER model much the same as it would if encountered in the field. However, if the researcher was interested in using this model and interface to study strategies for ordnance removal, then the proximity data for the current implementation should raise questions.

Researchers at the International University Bremen have conducted additional validation studies of USARSim sensor models, including the Hokuyu laser range finder that is used for localization and mapping (Carpin, Birk, Lewis, Wang, & Jacoff, 2005), and the comparison of results from a variety of machine vision techniques that are applied to camera video and synthetic video from the USARSim's image server (Carpin, Stoyanov, Nevatia, Lewis, & Wang, 2006). It is our hope that new models that have been added to the simulation will be accompanied by validation data. As our library of models and validation data expands, we hope to begin incorporating more rugged and realistic robots, tasks, and environments. Tame tasks performed by wheeled robots in portable arenas are only a hint of the potential that high-fidelity HRI simulation offers for human control of robots in large-scale hazardous environments for which we could not otherwise gain experience.

Case Studies

USARSim provides a powerful tool that allows for the assessment of fundamental design decisions for HRI without the expense of constructing and programming physical robots. To illustrate its use, we briefly describe experiments in two aspects of remote viewing: perception of attitude from camera video (Wang et al., 2004) and the effectiveness of various camera configurations (Hughes & Lewis, 2004). Two other HRI research areas that are likely to benefit from simulator-based research, adjustable autonomy, and multirobot control are also briefly discussed.

Case 1: Perception of Attitude

Situational awareness is particularly critical to teleoperation activities. The problem of getting lost while relying on video feeds was most recently documented in Darken, Kempster, and Peterson (2001). Drury, Scholtz, and Yanco (2003) extended the notion of awareness to the field of human-robotics interaction: "HRI awareness is the understanding that the human has of the location, activities, status, and surroundings of the robot; and the knowledge that the robot has of the human's commands necessary to direct its activities and the constraints under which it must operate" (Drury et al., 2003, p. 913). The breakdown of perceptual modalities, lack of vestibular and proprioceptive cues, and lack of direct interaction with the environment can significantly detract from the operator's efforts to maintain awareness of the robot (Tittle, Roesler, & Woods, 2002).

One crucial element of awareness is understanding the robot's attitude defined by the pitch or roll of a robot. If the degree of roll is too great, the robot may roll over onto its side and become inoperable. Although it may be easy to see from the outside when a robot is approaching these limits, it is surprisingly difficult when trying to teleoperate a robot from an onboard camera (Drury et al., 2003).

The lack of context for robot attitude in camera-supplied video creates an illusion of flatness. This fixed-camera illusion occurs because a robot's camera returns an image that is normal to the robot's frame. If the robot is sitting perpendicular to a slope, the camera image will therefore appear flat. These camera-linked illusions of flatness are a likely source of frequent, unanticipated rollovers (McGovern,1991) and the difficulties in realizing that a robot had actually rolled over, as reported by Casper (2002).

Wang et al. (2004) reported on using USARSim to compare displays in which attitude information is separated from the camera view with one in which attitude and camera view are integrated through presenting the camera view in a gravity-referenced orientation. This interface comparison is well suited for study using a robotic simulator because the simulator allows one to repeatedly run robots over rough terrain that is likely to induce rollovers without risking damage to expensive equipment. Though readily available, the sensors needed for real-time estimation of pose are also expensive, and programming would be required to reorient the camera views. Using the simulator, one can easily test one's hypotheses about the effects of gravity-referenced displays on operator performance without expensive implementation or risk to equipment.

The instruments used to display attitude were a pitch/roll indicator and a gravity-referenced camera (GRC), shown in Figure 10. The fixed-camera (FC) screen displayed the roll angle and pitch angle of the robot on bar charts at the bottom left

Fixed Camera *Gravity-Referenced Camera*

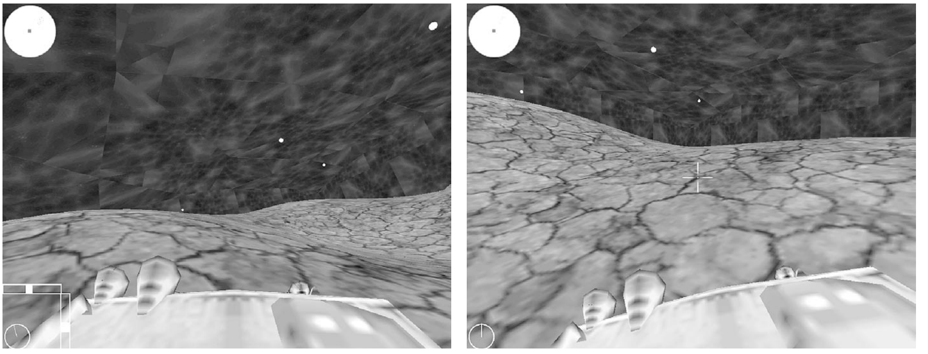

Figure 10. The fixed camera (left) displays attitude on an indicator at the bottom of its screen. The gravity-referenced camera conveys roll by showing the scene from true vertical.

corner. The GRC simulated a camera view that conveyed roll by maintaining a gravity-referenced view but without any other pitch or roll indication. From the GRC camera, the operator views the scene from a true horizontal perspective while the landscape and visible portion of the robot body appear to rotate as the vehicle rolls.

Lewis, Wang, and Hughes randomly assigned 26 participants to the FC or the GRC condition. The ordering of environments was also randomly assigned. Participants using the GRC were faster in completing their circuits (ANOVA, $F_{1,24} = 7.031$, $p = .014$) and spent less time backing out of impassable areas (ANOVA, $F_{1,24} = 6.11$, $p = .021$). They also subjected their robots to less extreme attitudes, measured as the sum of squared roll (ANOVA, $F_{1,24} = 6.35$, $p = .019$). Participants using the GRC reported using their view of the vehicle body and nearby landscape to estimate attitude (indoor $p = .015$, outdoor $p = .002$). FC subjects reported using a variety of indications to estimate attitude, but only one rated the pitch/roll indication as a primary source of attitude information.

This experiment shows that camera video is particularly compelling for those performing teleoperation tasks, making it difficult for them to incorporate information from a separated attitude indicator. As hypothesized, the use of a gravity-referenced display containing a partial view of the vehicle's body appears to repackage attitude information in a way that was easy for the operators to assimilate.

Case 2: Camera Placement and Control

The strongest perceptual link to the remote environment often comes through a video feed supplied from a camera mounted on the robot (Fong & Thorpe, 2001). These video feeds are often generated by cameras that are mounted low to the ground and restricted to a very narrow field of view, offering unfamiliar viewing angles and distorting the sense of scale (Murphy, 2004). Moreover, there is appreciable variability in the strategies for the generation and manipulation of the video stream, which can have a dramatic influence on the overall experience.

Designers who wish to balance the appeal of direct control with the complications of remote perception may need to consider the full range of available techniques. McGovern (1991) provided accounts of robotic systems that included independently controlled cameras, cameras that were dependent on the steering mechanisms, and multiple fixed cameras. Issues of camera placement and control are well suited for testing in simulation because the factors affecting camera control, navigation, and search can plausibly be presumed to be the same for both simulated and actual robots. Testing in simulation avoids the problems of finding mounting points for multiple cameras, altering application logic, and dealing with bandwidth limitations for wireless video.

Hughes and Lewis (2004) used USARSim to compare the use of an independent, controllable camera with a fixed camera in search tasks. They assigned 39 participants to one of the conditions and asked them to explore a complex environment with the task of locating as many target objects as possible. Targets were identified on two levels of specificity: Objects were to be initially identified by class, and then confirmed by a discriminating feature. For example, targets were described to the searchers as red cubes with a yellow letter marked on one face. This design forced the explorers to engage in global navigation to locate objects from a distance and engage in local navigation to inspect the target more closely to identify the discriminating feature.

The data collected from this experiment reveal that the use of an independent, controllable camera resulted in participants finding 10% more of the available targets than with the fixed camera ($t(37) = 1.75$, $p < .05$). The major shortcoming of the fixed camera was its inability to efficiently perform inspection activities – acquiring a useful point of view within a limited range. The operators needed to periodically toggle the orientation of the robot between facing the object and moving around the object. Knowing when to turn to face the object required that the controller have a good sense of both the overall configuration and scale of the environment. For many applications of robotic teleoperation, it is unlikely that either is the case.

In addition to the cognitive burdens that the fixed-camera approach will likely introduce, there is also the problem of making repeated physical adjustments to the orientation of the robot. Not only is the probability increased that the robot will get stuck or be obstructed, but designers should be concerned about the amount of energy that is required to repeatedly pivot the entire robot back and forth.

This experiment also provides some qualitative evidence that operators generally do not travel long distances looking in a direction different from the motion vector. This information could justify the inclusion of a “homing” button that automatically returns the camera to the forward-facing position.

These examples illustrate the use of USARSim to study HRI. The camera configuration experiment depended primarily on USARSim’s accurate simulation of camera video and ability to simulate camera control. The attitude experiment relied on a combination of physics to simulate rollovers and camera control to simulate CRC and FC conditions. The relative ease with which these experiments could be set up and run contrasts sharply with the efforts that would have been needed to collect these data using actual robots.

Adjustable Autonomy and Mixed Initiative

To effectively control the robot, the operator must simultaneously manage the cognitive tasks of understanding the layout of the environment and knowing where to look to find relevant information, along with mastering the physical interaction that is required to position the robot in locations that offer meaningful viewpoints. This is not a trivial assignment. The effort applied to manipulating the robot may compete with the task of extracting relevant information from the environment.

From the beginning, mobile robotic systems have interleaved operator inputs with automatic actions in a less structured way than the large-scale systems widely studied in human factors. In mobile robotics, the operator and automation are often simultaneously involved in the same task. In teleoperation, for example, a robot might have safeguards with acceleration and velocity limits and overcurrent and obstacle interlocks. The operator's role of actively controlling a robot that can actively assert control to protect itself is significantly different from the "set it and wait" turn-taking found with conventional automation.

Because the roles of the human operator and robotic automation tend to be so tightly coupled and mutually supportive, Terry Fong (2001) has argued that the HRI problem should be viewed as one of *collaborative control* rather than the traditional views of a robot and its programmer (AI) or an operator controlling a robot (human factors). Instead of these master-slave relationships, the mixed-initiative interaction model dictates that the execution of subtasks is "opportunistically negotiated," depending on who is best equipped to meet an objective (Hearst, Allen, Horvitz, & Guinn, 1999). This approach can leverage the human operators' perceptual discrimination, cognitive flexibility, spontaneity, and ingenuity, allowing them to focus their efforts on judgments and decision making that exceed the current capability of automation. USARSim provides an ideal tool to study these interactions because it combines detailed models of robot kinematics, sensors, and the environment that enable one to program realistic control algorithms with high-quality video simulation for the operator interface.

Multirobot Control

Many hypothesized applications of mobile robotics require multiple robots. Although current implementations are largely limited to static domains, such as RoboCup soccer (Noda et al., 1998), envisioned applications including interplanetary construction or cooperating uninhabited aerial vehicles (UAVs) will require close coordination and control between human operator(s) and cooperating teams of robots in uncertain environments. The complexity of the operator's task is substantially increased because operators must continually shift attention among robots under their control, maintain situation awareness for both the team and individual robots, and exert control over a complex system.

In the simplest case, an operator controls multiple independent robots, interacting with each as needed. Control performance at this task has been investigated as a matter of neglect tolerance (Crandall, Goodrich, Olsen, & Nielsen, 2003) or bottlenecks (Nickerson & Skiena, 2005). Multirobot control at NIST's USAR competitions has been predominantly of this type. To perform dependent tasks such as formation movements or cooperative activities, robots need an additional coordination control layer that introduces further complexity into the operator's task.

Programming a team's behaviors prior to its mission is the most common form of control (Endo, MacKenzie, & Arkin, 2004; Miller & Parasuraman, 2003) and typically characterizes a mission into phases in which different team and individual behaviors are active. During the course of the mission, a number of forms of control may be possible, depending on the control layer with which the operator interacts. Under some conditions, an operator might choose to detach a robot from its team and control it directly, or the team might require a human input, such as an authorization to attack.

In control through redirection (Scerri et al., 2004), the operator might modify the team's goals by doing things such as moving a waypoint or adding a region of interest. Alternatively, the operator might control the team's behavior by interacting with its coordination mechanism to substitute a selected plan; for example, pressing a button to shift from offense to defense in Roboflag (Parasuraman et al., 2005). Less directly, the operator might affect the team's behaviors by altering individual behavioral parameters such as changing the value of a robot's *wanderlust* (Endo et al., 2004). Finally, the parameters governing the coordination algorithms themselves might be altered to modulate team performance (Scerri et al., 2004), changing the distance threshold for accepting a role assignment, for example.

The availability of any of these avenues of control is determined by the design of the automation. At present, mission planning, plan substitution, and redirection appear to be the most widely used. Unlike individual robot control, which has been studied for many years (Sheridan, 1992), intensive research on the control of multiple robots is only about 10 years old (Adams, 1995; Ali, 1999; Parker, 1995) and lacks widely accepted techniques or approaches to HRI.

Because the role of the human operator is so strongly tied to the algorithms that are chosen to provide coordination and the avenues of control that are granted, it is essential that these problems be approached as a (human + automation) × robot problem rather than restricting HRI considerations to the interface. The USARSim simulation was designed expressly to support this capability, taking advantage of the UnrealEngine2 client-server architecture. The advantages of simulation for the study of HRI are multiplied for multirobot control when issues of cost and logistics increase along with the number of robots.

Conclusions

Traditionally, human factors researchers have worried about the trade-off between cost and fidelity of simulation. Taking the brute force approach of simulating everything in highest attainable fidelity was prohibitively expensive, whereas lower-fidelity simulation left open the question of whether the approximation actually captured the characteristics of concern from the real system. The commoditization of simulation through mass-market video games has stood this dilemma on its head.

Because operator stations for controlling mobile robots most often use either a laptop computer or similar interface, end-to-end simulation for mobile robot control is now possible at very low cost and without sacrificing fidelity.

The ease of modeling new robots and environments that comes with reliance on commercial tools allows the simulation to be easily extended, which increases its usefulness to other researchers beyond the single configuration/scenario found in many research simulations. USARSim also benefits from involvement with RoboCup competitions, where it is used by the RoboCupRescue League (*http://www.robocuprescue.org/rescuerobots.html*) for the Virtual Robot USAR competition, a contest emphasizing human interaction with autonomously coordinating robot teams. As additional users develop and contribute models and environments, the simulation becomes richer and more likely to contain the types of robots, sensors, and terrain needed for a variety of HRI research interests. By benefiting at two levels from contributions of larger groups, USARSim can provide a more stable and well-supported research platform than could be developed by a single research group in isolation.

There are four major advantages to using a simulator such as USARSim for HRI research:

1. Use of a common tool provides the opportunity to replicate and compare research findings, which accelerates progress of the science. There is also the opportunity to share models and control code, as well as other research developments, allowing diverse groups to collaborate via artifacts.
2. Low-cost, high-fidelity simulation offers researchers who are interested in HRI, but unable to afford or maintain a robotics lab, the opportunity to work in the area. It also provides economies for current robotics researchers, particularly in expensive areas such as multirobot control.
3. Simulation allows tighter experimental control, such as starting robots from identical locations, exposing them to the same environmental events, or generating terrain of precisely-determined characteristics. It can also prevent unintentional factors, such as a misaligned camera or dead battery, from invalidating experimental results.
4. Simulation allows researchers to investigate situations that are difficult to observe in a lab setting, such as fire or collapse or the effects of noise and dropouts from a radio model. A related capability is testing the operational impact of new technology, such as 3-D laser scanning, which may be too expensive to obtain or is not yet available.

We feel these advantages are compelling and hope to see increased use of USARSim for HRI research. The current version of the simulator is available for download at *http://www.sourceforge.net/projects/usarsim*.

Acknowledgments

This work was supported by NSF grant NSF-ITR-0205526 and AFOSR PRET F49640-01-1-0542, John Tangney program manager.

References

Ageia Technologies. (2006, March). Advanced gaming Physics: Defining the new reality in PC hardware. *http://www.ageia.com/pdf/wp_advanced_gaming_physics.pdf*, accessed January 8, 2007.

Adams, J. (1995). *Human management of a hierarchical system for the control of multiple mobile robots*. Ph.D. dissertation, University of Pennsylvania, Philadelphia.

Ali, K. (1999). *Multiagent telerobotics: Matching systems to tasks*, Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA.

Best, B., Lebiere, C., & Scarpinatto, C. (2002). Modeling synthetic opponents in MOUT training simulations using the ACT-R cognitive architecture. *Eleventh Conference on Computer Generated Forces and Behavior Representation*, Orlando: Institute for Simulation and Training, University of Central Florida.

Bowman, D., Koller, D., & Hodges, L. (1997). Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. *Virtual Reality Annual International Symposium* (pp. 45–52). Albuquerque, NM: IEEE.

Burke, J., Murphy, R., & Coovert, M. (2004). Moonlight in Miami: An ethnographic study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Journal of Human-Computer Interaction, 19*, 85–116.

Carpin, S., Birk, A., Lewis, M., Wang, J., & Jacoff, A. (2005). High fidelity tools for rescue robotics: Results and perspectives. *Robocup 2005: Robot Soccer World Cup IX*, Springer, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin/Heidelberg.

Carpin, S., Stoyanov, T., Nevatia, Y., Lewis, M., & Wang, J. (2006). Quantitative assessments of USARSim accuracy. CD of the *Proceedings of PerMIS 2006.*

Casper, J. L. (2002). *Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center*, Masters' thesis, University of Southern Florida, Tampa, FL.

Chadwick, R., Gillan, D., Simon, D., & Pazuchanics, S. (2004). Cognitive analysis methods for control of multiple robots: Robotics on $5 a day. *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting* (pp. 688–692). Santa Monica, CA: Human Factors and Ergonomics Society.

Crandall, J., Goodrich, M., Olsen, D., & Nielsen, C. (2005). Validating human-robot interaction schemes in multi-tasking environments. *Transactions on Systems, Man, and Cybernetics–Part A*: *35*(4), Special Issue on Human-Robot Interaction.

Darken, R., Kempster, K., & Peterson, B. (2001). Effects of streaming video quality of service on spatial comprehension in a reconnaissance task. CD of the *Proceedings of the Meeting of I/ITSEC, St. Petersburg, FL*, distributed by NTSA, Arlington, VA.

Dorais, G., Bonasso, R., Kortenkamp, D., Pell, B., & Schrekenghost, D. (1999). Adjustable autonomy for human-centered autonomous systems on Mars. *Proceedings of the AAAI Spring Symposium on Agents with Adjustable Autonomy*, Menlo Park, CA: AAAI Press.

Drury, J., Scholtz, J., & Yanco, H. (2003). Awareness in human-robot interactions. *IEEE Conference on Systems, Man and Cybernetics* (pp. 912–918).Washington DC: IEEE.

Endo, Y., MacKenzie, D., & Arkin, R. (2004). Usability evaluation of high-level user assistance for robot mission specification. *Transactions on Systems, Man and Cybernetics, Part C, 34*(2), 168–180.

Fong, T. (2001). *Collaborative control: A robot-centric model for vehicle teleoperation.* Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA.

Fong, T., & Thorpe, C. (2001). Vehicle teleoperation interfaces. *Autonomous Robots, 11* (pp. 9–18). Kluwer Academic Publishers, The Netherlands.

Gerkey, B., Vaughan, R., & Howard, A. (2003). The Player/Stage project: Tools for multi-robot and distributed sensor systems. *Proceedings of the International Conference on Advanced Robotics* (ICAR 2003) (pp. 317–323). University of Coimbra, Portugal: IEE.

Hearst, M., Allen, J., Horvitz, E., & Guinn, C. (1999). Trends & controversies: Mixed-initiative interaction. *IEEE Intelligent Systems, 14*(4), 14–23.

Hughes, S., & Lewis, M. (2004). Robotic camera control for remote exploration. *ACM CHI 2004 Conference on Human Factors in Computing Systems* (pp. 511–517). Vienna, Austria: ACM.

Jacobson, J., & Lewis, M. (2005). Game engine virtual reality with CaveUT. *Computer*, *38*(4), 79–82.

Jacoff, A., Messina, E., Evans, J. (2001). Experiences in Deploying Test Arenas for Autonomous Mobile Robots, CD of *Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop*, in association with IEEE CCA and ISIC, Mexico City, Mexico, Sept. 4, 2001.

Johnson, L., Marsella, S., Mote, N., Viljhalmsson, H., Narayanan, S., & Choi, S. (2004). Tactical language training system: Supporting the rapid acquisition of foreign language and cultural skills. *Proceedings of InSTIL/ICALL2004*. *http://sisley.cgm.unive.it/ICALL2004/papers/005Johnson.pdf*. Accessed January 8, 2007.

Konolige, K., & Myers, K. (1996). The Saphira architecture for autonomous mobile robots. In D. Kortenkamp, R. P. Bonasso, & R. Murphy (Eds.), *AI-based mobile robots: Case studies of successful robot systems.* Cambridge, MA: MIT Press.

Lee, P., Ruspini, D., & Khatib, O. (1994). Dynamic simulation of interactive robotic environment. *International Conference on Robotics and Automation* (pp. 1147–1152). San Diego, CA: IEE.

Lewis, M., & Jacobson, J. (2002). Game engines in research. *Communications of the Association for Computing Machinery, 45*, 27–48.

Magerko, B., Laird, J., Assanie, M., Kerfoot, A., & Stokes, D. (2004). AI characters and directors for interactive computer games. CD of *Proceedings of the 2004 Innovative Applications of Artificial Intelligence Conference*. Menlo Park, CA: AAAI Press.

McGovern, D. (1991). Teleoperation of land vehicles. In S. Ellis (Ed.), *Pictorial communications in virtual and real environments* (pp. 182–195). New York: Taylor & Francis.

Miller, C., & Parasuraman, R. (2003). Beyond levels of automation: An architecture for more flexible human-automation collaboration. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* (pp. 182–186). Santa Monica, CA: Human Factors and Ergonomics Society.

Murphy, R. (2003). Gaps in rescue robotics. CD *IEEE Workshop on Safety, Security, and Rescue Robotics*. Tampa, FL.

Murphy, R. (2004). Human-robot interaction in rescue robotics. *Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 34*(2), 138–153.

Nickerson, J., & Skiena, S. (2005). Attention and communication: Decision scenarios for teleoperating robots. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. Waikaloa, HI: IEEE.

Nielsen, C., Ricks, B., Goodrich, M., Bruemmer, D., Few, D., & Walton, M. (2004). Snapshots for semantic maps. *Proceedings of the 2004 IEEE Conference on Systems, Man, and Cybernetics.* Hague, Netherlands: IEEE.

Noda, I., Suzuki, S., Matsubara, H., Asada, M., & Kitano, K. (1998). Robocup '97: The first world cup soccer games and conferences. *AI Magazine, 19*(3), 49–59.

Olsen, D., Jr., & Wood, S. (2004). Fan-out: Measuring human control of multiple robots. *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems* (pp. 231–238). Vienna, Austria: ACM.

Parasuraman, R., Galster, S., Squire, P., Furukawa, H., & Miller, C. (2005). A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with RoboFlag. *IEEE Transactions of Systems, Man, and Cybernetics Part A, 35*(4), 481–493.

Parker, L. (1995). *Heterogeneous multi-robot cooperation*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Prasithsangaree, P., Manojlovich, J., Hughes, S., & Lewis, M. (2004). UTSAF: A multi-agent-based software bridge for interoperability between distributed military and commercial gaming simulation, *Simulation*, *80*(12), 647–657.

Ricks, B., Nielsen, C., & Goodrich, M. (2004). Ecological displays for robot interaction: A new perspective. *Proceedings of IROS 2004.* Sendai, Japan: IEEE.

Scerri, P., Liao, E., Guorning, L., Sycara, K., Yang, X., & Lewis, M. (2004). Coordinating very large groups of wide area search munitions. In D. Grundel, R. Murphey, & P. Pardalas (Eds.), *Theory and algorithms for cooperative systems.* Singapore: World Scientific.

Sheridan, T. B. (1992). *Telerobotics, automation and human supervisory control*. Cambridge, MA: MIT Press.

Tittle, J., Roesler, A., & Woods, D. (2002). The remote perception problem. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting* (pp. 260–264). Santa Monica, CA: Human Factors and Ergonomics Society.

Wang, J., Lewis, M., & Gennari, J. (2003). USAR: A game-based simulation for teleoperation. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* (pp. 493–497). Santa Monica, CA: Human Factors and Ergonomics Society.

Wang, J., Lewis, M., & Hughes, S. (2004). Gravity-referenced attitude display for teleoperation of mobile robots. *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting* (pp. 2662–2666). Santa Monica, CA: Human Factors and Ergonomics Society.

Woods, D., Tittle, J., Feil, M., & Roesler, A. (2004). Envisioning human-robot coordination in future operations. *Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 34*(2), 210–218.

Yanco, H., Drury, J., & Scholtz, J. (2004). Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Journal of Human-Computer Interaction, 19*(1–2), 117–149.

Zaratti, M., Fratarcangeli, M., & Iocchi, L. (2006). A 3D simulator of multiple legged robots based on USARSim. CD from *Robocup 2006: Robot Soccer World Cup X, Springer, Lecture Notes in Artificial Intelligence.* Bremen, Germany: Springer-Verlag, Berlin/Heidelberg.

Michael Lewis is a professor in the School of Information Sciences at the University of Pittsburgh, A200 IS Building, 135 North Bellefield Ave., Pittsburgh, PA 15260, ml@sis.pitt.edu. He has a Ph.D. and M.S. in engineering psychology from Georgia Institute of Technology and a B.A. in experimental psychology from the University of California, Santa Barbara. His research focuses on human interaction with intelligent automation, particularly mixed-initiative control for agent and robotic systems. Dr. Lewis's research has been supported by grants from AFOSR, AFRL, NASA, NIST, NSF, and ONR, among others, and has been reported in 130+ technical papers in the areas of human factors and computer science.

Jijun Wang is a Ph.D. student in the School of Information Sciences at the University of Pittsburgh. He holds E.E. and M.E. degrees from Tsinghua University. Mr. Wang is the primary developer of the USARSim simulation, which he is using in his research on human control of robot teams.

Stephen Hughes is an assistant professor at Roanoke College. He has a Ph.D. from the University of Pittsburgh in information science, an M.S. from the University of Indiana, and a B.S. from Dickinson College in computer science. His research involving viewpoint control in virtual environments and robotics has been published in *Human Factors* and *IEEE Transactions on Systems, Man, and Cybernetics*, among other outlets.