

Human-Humanoid Robot Interaction through Virtual Reality Interfaces

Murphy Wonsick and Taşkın Padır
Institute for Experiential Robotics
Northeastern University
360 Huntington Ave
Boston, MA 02115
{wonsick.m, t.padir}@northeastern.edu

Abstract—In the last decade, there have been great advancements in virtual reality (VR) resulting in its availability for everyday consumers. As VR becomes more ubiquitous, there is an opportunity to utilize this technology to create intuitive operator interfaces for interaction with complex dynamic systems, such as humanoid robots. As evidenced in the DARPA Robotics Challenge (DRC), current interfaces for humanoids primarily use a standard computer setup with monitor, keyboard, and mouse requiring operators to process 3D data with 2D devices. And although these interfaces can be very capable in operating a robot, they are often complex and require expert operators as well as extensive training. However, this paradigm can be changed with VR by allowing operators to visualize and interact with 3D data in a 3D environment, allowing for a more natural interaction. In this paper, we present our work on converting a typical interface to a virtual reality interface for NASA's humanoid robot, Valkyrie. We compare our standard computer interface and our VR interface for Valkyrie, as well as the shared control planners and system architecture that make our interfaces possible. The goal of this work is to better understand the utility of virtual reality interfaces and how they can be employed in human-supervised robot applications so that we may move towards more intuitive and easy-to-use interfaces for control and interaction.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. BACKGROUND	2
3. SYSTEM ARCHITECTURE	2
4. 2D INTERFACE	3
5. VR INTERFACE	3
6. DISCUSSION AND FUTURE WORK	5
7. CONCLUSION	6
ACKNOWLEDGMENTS	6
REFERENCES	6
BIOGRAPHY	7

1. INTRODUCTION

Supervisory-control interfaces are an important aspect for robot operations in extreme environments, such as disaster response, nuclear decommissioning, and space exploration, as you often want a human-in-the-loop to help evaluate and make decisions for critical objectives of a task at hand [1]. However, the majority of current interfaces for such environments rely on computer monitors to display information and keyboards, mice, or joysticks to interact with the system. These approaches force operators to process and interact with 3D data with 2D devices, which can increase the workload for

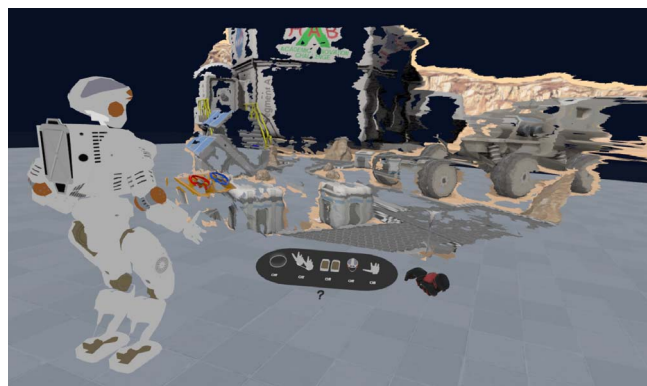


Figure 1: VR interface designed for human-humanoid interaction.

operators and are often cumbersome to use.

Virtual reality (VR) provides an opportunity to change this paradigm by allowing operators to visualize and interact with 3D data in a 3D environment. In fact, VR is already being used in robot-assisted surgery and there have been numerous studies that investigate using 3D views over 2D views in this context and have shown that 3D views outperform 2D views [2–4]. However, although VR has demonstrated utility in operator processing of 3D data, its usage has still yet to be widely adopted in other robot interfaces. This is likely due to the fact that, even though virtual reality technology has been around since the 1960s [5], it is just in the past decade that there have been rapid advancements in the field to where VR devices with immersive visualization and 6 DOF tracking are now commercially available and relatively affordable. With these advancements, it is now more feasible to create VR interfaces for interacting with robots. This capability allows for the opportunity to create more natural and intuitive interfaces over traditional interfaces for supervisory-control.

In this paper, we present our work on developing supervisory-control virtual reality interfaces for humanoid robots using NASA's humanoid robot, Valkyrie, as our validation platform. To accomplish this, we first created a simplified traditional 2D interface that was modeled after several of the interfaces developed for the DARPA Robotics Challenge (DRC). Next, we developed a virtual reality interface, as seen in Figure 1, that utilizes the same planners and control methods as the traditional interface so that both interfaces provide the same functionality and only vary in interaction techniques and data visualization. The ultimate goal of this research is to better understand virtual reality interfaces for humanoid robots so that we may move toward more intuitive



Figure 2: Team WPI-CMU operator room at the DRC.

and easy-to-use interfaces.

The paper is organized as follows. We will first discuss the prior research in the areas of traditional interfaces and virtual reality interfaces in Section 2. Next, we quickly describe the hardware and system architecture used for both interfaces in Section 3. Then we will describe our 2D interface and planners used that lay the foundation for the VR interface in Section 4. Following, we will present the current iteration of our VR interface in Section 5. Finally, we will discuss the next steps for this work and other areas to still explore to support the continued development of virtual reality interfaces for humanoid robots in Section 6.

2. BACKGROUND

The DARPA Robotics Challenge (DRC) was aimed at accelerating progress in human-supervised autonomy for robots used in humanitarian aid and disaster relief, which led to improvements in supervisory-control interfaces, specifically for humanoid robots [6, 7]. Although the interfaces developed for the DRC enabled the operators to complete the challenge tasks, they were very complex and often required multiple operators to help process all the data. Figure 2 shows team WPI-CMU’s operator room during the DRC, which utilized a total of five passive and active operators and four different workstations. Most other teams had similar setups. Additionally, almost all of the teams exclusively used traditional input devices and data displays, including monitors, keyboard, mice, game controllers, switch boards, etc [6, 8, 9]. VR has the potential to help reduce the complexity of these interfaces by immersing the operator in a 3D environment to visualize and interact with 3D data rather than traditional interfaces which only allow for visualization and interaction on a 2D level. However, at the time of the DRC, VR devices were just starting to become available to developers. Team ViGIR did use a VR headset for data visualization to assist in situational awareness but had no interaction capabilities [10].

Since the DRC, VR has become widely commercially available which has led to an increase in the development of VR interfaces for robots [11]. Several works of research have compared VR interfaces with traditional interfaces and overall have shown that they reduce task completion time, lower operator workload, and are considered more usable

and likable among users [12–14]. However, most of the research so far has focused on developing direct teleoperation interfaces where an operator controls a robot manipulator, by either directly or indirectly tracking human movement to signal where the robot’s end-effector should go, or navigates a mobile robot, by using a joystick or similar devices [12, 13]. These types of interfaces require the human to be always involved with the robot operations and ultimately do not take advantage of autonomous behaviors or planners the robot may have, such as motion planners or waypoint navigation. It has been found though that by operating a robot with positional control, i.e., placing waypoints for the robot to traverse through, over direct teleoperation control, where the operator’s movements are linked to the robot’s movement, can lead to an increase in accuracy and faster completion times [15]. Furthermore, there is little development so far on VR interfaces for bipedal humanoid robots. [16] defines a system architecture that utilizes a VR headset, VR controllers, and an omnidirectional treadmill to create a fully immersive teleoperation interface to operator a humanoid robot, but their system is a telepresence one with the goal of providing a real-time sensation to the human of being in another place and not one where the robot and operator work together. The aim of this work is to continue development in VR interfaces for operating robots by presenting our VR interface for Valkyrie that utilizes supervisory control elements and is based on a traditional 2D interface modeled after the ones seen in the DRC.

3. SYSTEM ARCHITECTURE

Valkyrie is a 32 degree of freedom (DOF) humanoid robot designed by NASA’s Johnson Space Center and was originally designed to compete in the DRC Trials in December 2013 [17]. The robot consists of 5 major mechanical sub-assemblies: two 7 DOF arms, two 6 DOF legs, and a torso that consists of a 3 DOF waist and 3 DOF neck. Additionally, each arm is equipped with a 6 DOF tendon driven hand consisting of three 1 DOF fingers and a 3 DOF thumb. Valkyrie has a single vision sensor of a Carnegie Robotics Multisense SL sensor unit combining a rotating Hokuyo LIDAR and a stereo camera pair that is located in the head. Interfacing with Valkyrie’s controllers and sensors is all done through ROS.

2D Interface

The 2D interface is a computer-based interface comprised of a monitor, keyboard, and mouse. It was developed using RViz, a visualization tool for ROS, and custom Qt Widgets, also known as Panels in RViz. RViz was chosen as the interface backbone since ROS is widely used in the robotics community making it familiar to many and for its built-in data visualization tools. RViz was also used for several of the interfaces designed for DRC [7, 18, 19].

VR Interface

To develop the VR interface, we use the Unity game engine and for hardware, we use an HTC Vive Pro headset with two Vive controllers and a waist tracker. Currently though, the Vive and Unity both run best on the Windows operating system and not on Linux, which is what Valkyrie, the 2D interface, and all the supporting ROS code uses. Therefore, to bridge the gap between the VR environment developed in Unity and ROS, we utilize an open-source software library called ROS# [20]. This allows for the same motion and footstep planner to be used between the two interfaces so that each interface uses the same back-end and therefore only vary

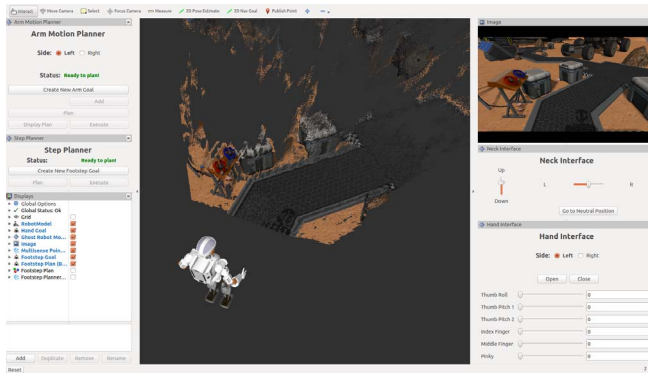


Figure 3: 2D interface developed using Rviz.

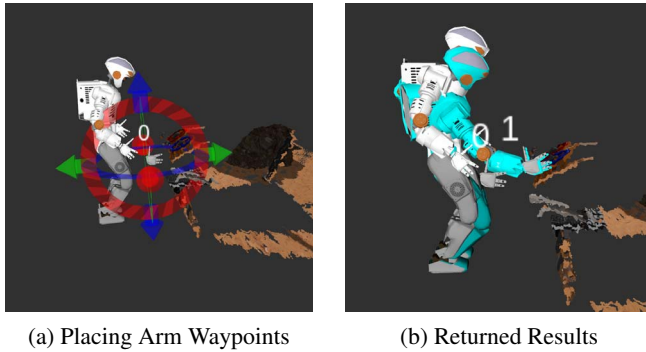


Figure 4: 2D interface arm motion planning.

in the actual front-end.

4. 2D INTERFACE

The 2D interface, as seen in Figure 3, is designed to mimic our humanoid interface designed for the DRC Finals [7], but with a reduced number of capabilities, inputs, and sensor data to allow for both a single operator and to reduce operator training time. The goal is to create a comparable interface to ones used at the DRC, but user-friendly enough to allow for non-experts to quickly learn and operate. The interface can be broken down into 4 custom panels: Arm Motion Planner, Step Planner, Neck Interface, and Hand Interface. Additionally, there are three built-in RViz panels: the main visualization window, a Displays panel, and an Image panel that contains a camera feed.

The main window consists of the built-in 3D data visualization of RViz and, for our specific interface, includes a point-cloud from Valkyrie's Multisense SL sensor located in the head, a robot model of Valkyrie's current joint state and planned joint state from the arm motion planner, and interactive markers to interface with the arm motion and step planners. The Displays panel allows users to select what is actively visualized in the main window and the Image panel for our interface is a camera feed from one of the stereo cameras in the Multisense SL.

Arm Motion Planner Panel

The Arm Motion Planner panel utilizes a whole body inverse kinematics solver that finds motion plans based on a set of cost and constraints [21, 22]. Cost and constraints can be categorized as either kinematic, collision avoidance, or ZMP.

To simplify the interface, operators only need to define the desired end-effector positions and a predefined default set of cost and constraints are used, i.e., balancing constraint or velocity cost, etc. To define the end-effector positions, operators can add interactive markers to the scene that represent waypoints for the end-effector to traverse through. These interactive markers display a single end-effector, marker visual depends on which side of the robot is being operated and are able to translate and rotate in 3D using arrows and circular scroll bars that surround the visual marker, as seen in Figure 4a. The markers are sequenced to allow for a variety of paths. Once an operator is done placing markers, they can request a plan. Plans return as successful if they do not violate any of the predefined costs. If they violate a cost, the operator is informed and must adjust the markers and try again. Once a successful plan is returned, the operator can then visualize the joint state of the robot at each placed marker to ensure it gives the desired outcome. Figure 4 shows an example of the arm motion planning process. Once satisfied with the plan, operators can then send the plan to the robot for execution.

Step Planner Panel

The Step Planner panel allows operators to designate goal positions for the robot to navigate to. It works as a single interactive marker that displays both feet and only allows the operator to interact in 2D, which is displayed as two sets of arrows for translating on a plane and a single circular scroll bar for rotation on the plane surrounding the visual of the pair of feet. Operators can position the desired final goal and a plan is generated using an A* search-based footstep planner. Operators are notified that the planner is actively planning by updating the status text, which is also updated once the planner either finds a plan or is unsuccessful in finding a plan by timing out. Once a successful plan is returned, interactive markers are automatically generated that display each planned footstep. Operators can select these individual footsteps and make minor adjustments to them as necessary. When finished making adjustments, operators are then able to send the plan to the robot for execution.

Neck Interface and Hand Interface Panels

The Neck Interface and Hand Interface panels both work in the same manner with a series of joint sliders and buttons with some predefined joint positions. The Neck Interface panel allows operators to move the head of the robot right-left and up-down with sliders and a predefined neutral position, where the robot is looking straight forward. The Hand Interface panel has sliders for each individual finger to open-close along with two predefined open hand and closed fist grasp. Sliders were added over additional grasp types to allow the operator to position the hand at any intermediate pose if needed.

5. VR INTERFACE

The aim of the VR interface is to provide the same exact functionality as the 2D interface, but to take advantage of the interaction methods and visualization capabilities that VR has to offer. Figure 1 shows a bird's-eye view of the VR interface. Like the 2D interface, there is a robot model that matches the current state of the Valkyrie along with the point-cloud from Valkyrie's Multisense.

The VR interface can be broken down into the same four sections as the 2D interface: Arm Motion Planner, Footstep Planner, Neck Control, and Hand Control. Rather than pan-

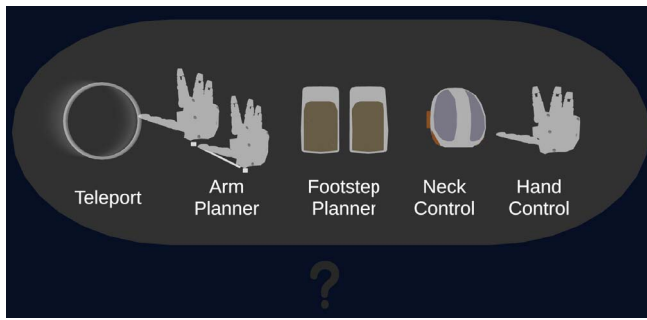
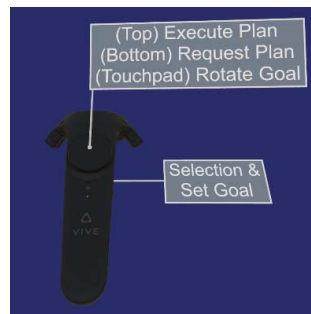


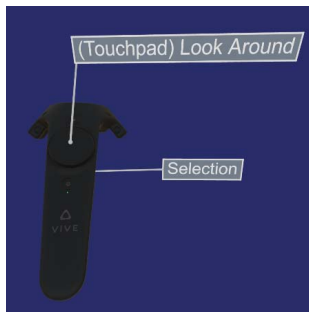
Figure 5: VR Control Panel acts similar a toolbelt by being located at the operator's waist and contains interactive icons to switch the controllers between the different modes, i.e. change out the tools.



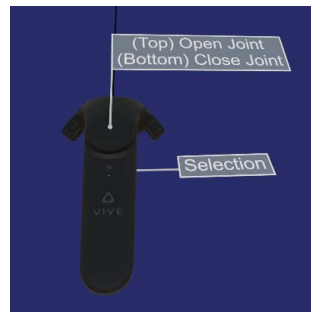
(a) Arm Planner Controls



(b) Footstep Planner Controls



(c) Neck Control Controls



(d) Hand Control Controls

Figure 6: VR Controls for the major modes. This help text can be toggled on/off by using the help button on the Control Panel.

els, each section can be considered a mode that the operator can switch a controller into, and each controller can be in a different mode. For example, the right controller can be in the Arm Motion Planner mode to help plan arm trajectories while the left controller can be in the Neck Control mode to help move the neck to look around. The buttons on the controller change based on the mode it is currently in. Figure 6 shows the button functions for each of the major modes. The operator also has the ability to switch between the modes at any time. In order to accomplish this, the operator is equipped with a control panel that contains interactive icons for mode switching, information on what mode each controller is in, and status update information for the planners. Figure 5 shows the default view of the control panel with no controllers activated. This virtual control panel is located on the operator's waist through use of a physical VR tracker attached to the operator, similar to a toolbelt. The

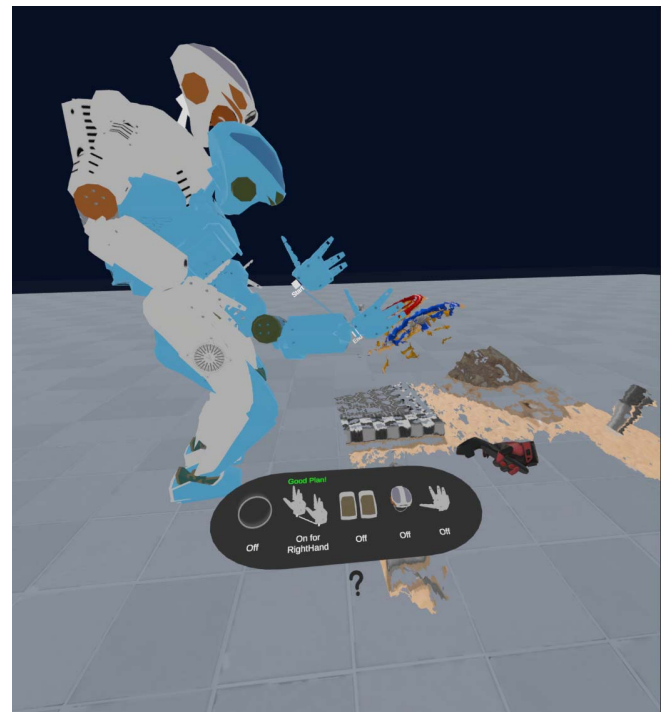


Figure 7: Arm Motion Planner mode. The Valkyrie hand models represent the operator defined trajectory waypoints and the blue Valkyrie model shows the final position of the returned plan.

purpose of this is to have the control panel always within the operator's reach and in a known constant location as they navigate around the scene, but yet still somewhere that does not obstruct the operator's view. Additionally, since VR adds the capability of having the operator navigate around the scene there is a Teleport mode to allow the user to quickly move about the VR environment. Furthermore, since VR controls are not as commonly used and have the ability to change as the operator changes between modes, there is a help button on the control panel that will bring up the current button commands of the controllers, which can be seen in Figure 6.

Arm Motion Planner Mode

The Arm Motion Planner mode functionally works in the same fashion as the 2D interface with the only major difference between the two is the actual interaction elements. In the 2D interface, the operator can place interactive markers that represent waypoints that they want the robot to plan to traverse through. This method requires markers to be generated in some predefined location and then moved to the desired location. However, in VR it is designed so that markers are generated at the location the controller is currently at. This overall results in less movement of the waypoints. Waypoints can still be adjusted in VR like in the 2D interface by hovering over the waypoint with the controller, holding down the selection button, and then moving it around until it is in the desired location. Adjustments are made much easier this way in VR since the controllers move in 3D space and are not confined to the arrows and circular scroll bars of the interactive markers. Also, the last created waypoint can be removed by squeezing the grip buttons on the controller allowing for quick editing. The remaining flow of control directly matches the 2D interface. The operator can request a

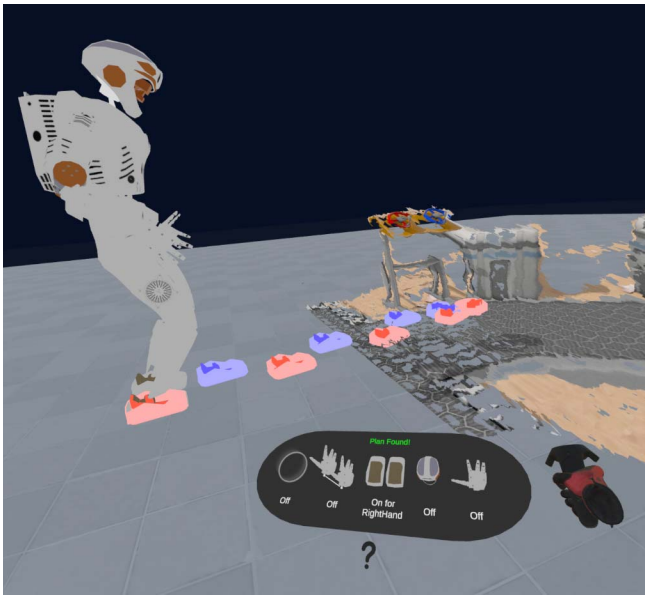


Figure 8: Footstep Planner mode. Example of returned footstep plan, the red and blue foot markers represent the planned steps for the right and left foot, respectively.

plan, display a plan, and then execute the plan if desired using the buttons on the trackpad, as seen in Figure 6a. Planning status and results returned from the planner are displayed on the control panel. If a successful plan is found after being requested, the plan is visualized with a blue translucent model of the robot that displays the robot's planned joint state for each defined waypoint. The benefit of the visualization in VR over 2D is the ease of which the VR operator can move around to ensure the plan is collision-free and matches the operator's trajectory intentions.

Footstep Planner Mode

The Footstep Planner mode works similarly to the Arm Motion Planner mode and its 2D interface counterpart. To place the footstep goal location though, the operator projects the goal marker, similar to how teleportation is accomplished. Additionally, while projecting the goal marker, the operator can also use the trackpad on the controller to rotate the marker at the same time. The button controls are shown in Figure 6b. After the desired goal location is set, the operator can request a plan and a text display of the planning status and results are shown on the control panel. When the footstep planner returns a successful plan, it is automatically displayed in the scene with colored markers, red for the right foot and blue for the left foot, and example of this can be seen in Figure 8. Footsteps can also be adjusted in the same manner as how waypoints are adjusted when in the Arm Motion Planner mode.

Neck Control Mode

Rather than using sliders to control the neck directions, we utilize the trackpad on the VR controllers, which can output the same two directional output as (up/down and left/right) as two sliders can. Figure 6c shows the button inputs when in this mode. The benefit of using the trackpad over sliders is that neck can quickly be moved by the operator to the desired position since it can be moved in two directions at one time, i.e., up and to the right or down and to the left.

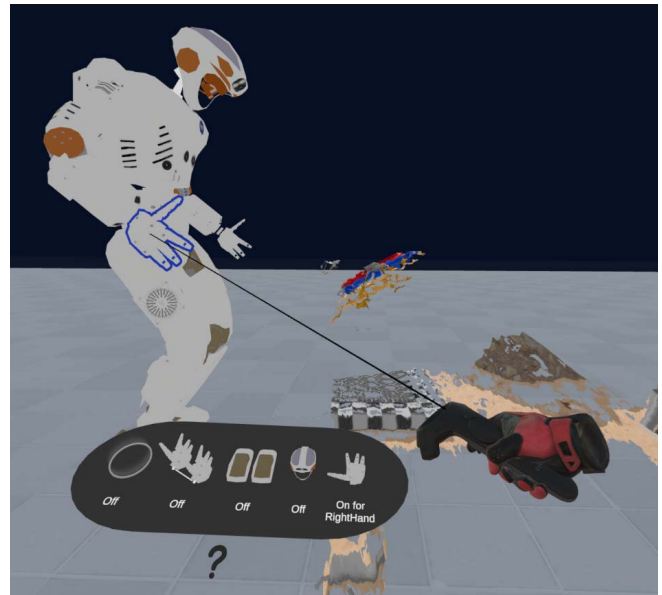


Figure 9: Hand Control mode. The laser pointer points to the palm of the right hand and will close the fingers in a power grasp when the trackpad is clicked.

Hand Control Mode

Like the Neck Control mode, we also elected to not use sliders to control individual joints in the Hand Control mode. Instead, we attach a laser pointer to the end of the controller to allow an operator to point to individual finger joints and then use the trackpad to open/close the finger. This was done over sliders since it does not require the operator to have knowledge of joint or finger names or map between the slider and the finger they want to move. To replicate the general open/close grasp buttons on the 2D interface, the same laser idea is applied, but to the palm of the hand. Figure 9 shows an example of this mode in action.

6. DISCUSSION AND FUTURE WORK

The goal of this work is to provide a baseline for future research studies to support identifying ways to create more intuitive and natural interfaces for humanoid robots. One important research area we would like to explore is comparing traditional supervisory-control 2D interfaces with VR ones for humanoid robots. There has been prior work in comparing traditional interfaces with VR interfaces for other types of robots, such as robot manipulators, mobile robots, and drones, and the results have shown that VR interfaces provide great promise to improve human-robot interaction. We envision that this trend will continue for humanoid robots, but we do acknowledge that there are potential downsides to using VR interfaces over traditional 2D interfaces. For example, VR requires a larger amount of infrastructure to support an interface compared to traditional 2D interfaces, which can be accomplished on a single computer or tablet. Table 1 presents the major pros and cons for each interface. Furthermore, it is still unknown how operators perform when using VR interfaces for robot operation over a long period of time, especially for complex tasks. Most of the current research in the field have operators operating robots for short periods of time for simple tasks. However, there are scenarios, such as disaster relief, where operators will need

	Pros	Cons
VR	• Interaction is in 3D, operators can move objects through real-world hand movements • Immersive viewpoint, which can allow for better situational awareness • Can navigate scene by using natural movements (i.e., moving head or walking) • Requires less workload on the operator	• Hardware intensive, requires VR setup (headset, controllers, tracking stations), dedicated computer to run the interface • Can induce motion sickness in some individuals • No personal situational awareness
2D	• Only requires a single computer • Keyboards, mice, and monitors are familiar devices • Operator maintains personal situational awareness	• Interaction is in 2D, requires operators to understand how to translate and rotate objects using rings and arrows • Must use mouse to navigate scene, requires understanding the controls on the mouse • Workload intensive

Table 1: Pros and cons of each interface.

to be operating the robot for potentially hours to accomplish the mission. Therefore, it is important to investigate how elements like accuracy, efficiency, and operator fatigue in VR compare to traditional interfaces.

Additionally, we would like to continue development on the VR interface and better understand and identify the best practices when designing VR interfaces for operating robots. The interaction methods we selected for the current VR interface went through a few iterations. However, no formal study was conducted and there were other interaction techniques considered that could potentially lead to a better user experience. For example, in the Arm Motion Planner mode, rather than drop waypoints to define a trajectory, we could have the operator move their arm through the desired trajectory, or rather than use the Hand Control mode, we could use VR gloves to control the fingers instead of using the controller to open/close individual joints. We would also like to investigate the effects of using two VR controllers versus one and whether the operator should be allowed to operate in only one mode at a time using both controllers or keep both controllers as separate tools that can each be in their own mode. For example, when a mode is selected have both controllers switch to that mode so that the operator is not responsible to remember which mode each controller is in. This also could allow for additional interactions to be used as both controllers would be active in a single mode.

Finally, both of our interfaces were designed to be simplified down to the minimum elements needed to accomplish basic tasks, such as pick-and-place operations in a static and relatively clutter free environment. The purpose of this was to allow for both interfaces to be quickly understood and usable for most individuals with little to no training. However, there is room for additional development to make both of our interfaces more robust and capable to a wide variety of environments and situations.

7. CONCLUSION

In this paper, we presented our supervisory-control VR interface for Valkyrie, NASA's humanoid robot that is based off our 2D traditional interface inspired by the humanoid interfaces designed for the DRC. Details on both interfaces were provided with focus on the interaction methods each one

uses and how utilizing the 3D environment VR provides can be beneficial to operators over standard 2D displays. Future work includes further development of the VR interface along with user-studies so that we may move towards more intuitive and easy-to-use interfaces for human-robot interaction.

ACKNOWLEDGMENTS

This research is supported by the National Aeronautics and Space Administration under Grant No. NNX16AC48A issued through the Science and Technology Mission Directorate, by the National Science Foundation under Award No. 1544895, 1928654, 1935337, and 1944453.

REFERENCES

- [1] L. Takayama, W. Ju, and C. Nass, "Beyond dirty, dangerous and dull: what everyday people think robots should do," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2008, pp. 25–32.
- [2] Y. S. Tanagho, G. L. Andriole, A. G. Paradis, K. M. Madison, G. S. Sandhu, J. E. Varela, and B. M. Benway, "2d versus 3d visualization: impact on laparoscopic proficiency using the fundamentals of laparoscopic surgery skill set," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 22, no. 9, pp. 865–870, 2012.
- [3] P. Storz, G. F. Buess, W. Kunert, and A. Kirschniak, "3d hd versus 2d hd: surgical task efficiency in standardised phantom tasks," *Surgical endoscopy*, vol. 26, no. 5, pp. 1454–1460, 2012.
- [4] S. B. Bhayani and G. L. Andriole, "Three-dimensional (3d) vision: does it improve laparoscopic skills? an assessment of a 3d head-mounted visualization system," *Reviews in urology*, vol. 7, no. 4, p. 211, 2005.
- [5] T. Mazuryk and M. Gervautz, "Virtual reality-history, applications, technology and future," 1996.
- [6] A. Norton, W. Ober, L. Baraniecki, E. McCann, J. Scholtz, D. Shane, A. Skinner, R. Watson, and H. Yanco, "Analysis of human-robot interaction at the darpa robotics challenge finals," *The International Jour-*

Journal of Robotics Research, vol. 36, no. 5-7, pp. 483–513, 2017.

- [7] M. DeDonato, F. Polido, K. Knoedler, B. P. Babu, N. Banerjee, C. P. Bove, X. Cui, R. Du, P. Franklin, J. P. Graff *et al.*, “Team wpi-cmu: Achieving reliable humanoid behavior in the darpa robotics challenge,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 381–399, 2017.
- [8] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin *et al.*, “Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals,” *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829, 2017.
- [9] J. Pratt, “Darpa robotics challenge (drc) using human-machine teamwork to perform disaster response with a humanoid robot,” Florida Institute for Human and Machine Cognition Inc. Pensacola United States, Tech. Rep., 2017.
- [10] A. Romy, S. Kohlbrecher, A. Stumpf, O. von Stryk, S. Maniopoulos, H. Kress-Gazit, P. Schillinger, and D. C. Conner, “Collaborative autonomy between high-level behaviors and human operators for remote manipulation tasks using different humanoid robots,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 333–358, 2017.
- [11] M. Wonsick and T. Padir, “A systematic review of virtual reality interfaces for controlling and interacting with robots,” *Applied Sciences*, vol. 10, no. 24, p. 9051, 2020.
- [12] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex, “Comparing Robot Grasping Teleoperation Across Desktop and Virtual Reality with ROS Reality,” in *Robotics Research*, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Cham: Springer International Publishing, 2020, pp. 335–350.
- [13] “Bringing Adaptive and Immersive Interfaces to Real-World Multi-Robot Scenarios: Application to Surveillance and Intervention in Infrastructures,” *IEEE Access*, vol. 7, pp. 86 319–86 335, 2019.
- [14] M. Maciaś, A. Dąbrowski, J. Fraś, S. Karczewski Michałand Puchalski, S. Tabaka, and P. Jaroszek, “Measuring Performance in Robotic Teleoperation Tasks with Virtual Reality Headgear,” in *Automation 2019*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Cham: Springer International Publishing, 2020, pp. 408–417.
- [15] R. Hetrick, N. Amerson, B. Kim, E. Rosen, E. J. de Visser, and E. Phillips, “Comparing Virtual Reality Interfaces for the Teleoperation of Robots,” in *2020 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 2020, pp. 1–7.
- [16] M. Elobaid, Y. Hu, G. Romualdi, S. Dafarra, J. Babic, and D. Pucci, “Telexistence and teleoperation for walking humanoid robots,” in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2019, pp. 1106–1121.
- [17] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater *et al.*, “Valkyrie: Nasa’s first bipedal humanoid robot,” *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.
- [18] S. Kohlbrecher, A. Romy, A. Stumpf, A. Gupta, O. Von Stryk, F. Bacim, D. A. Bowman, A. Goins, R. Balasubramanian, and D. C. Conner, “Human-robot teaming for rescue missions: Team vigir’s approach to

the 2013 darpa robotics challenge trials,” *Journal of Field Robotics*, vol. 32, no. 3, pp. 352–377, 2015.

- [19] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zajac, J. A. Bagnell, J. Brindza, C. Dellin *et al.*, “Chimp, the cmu highly intelligent mobile platform,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.
- [20] M. Bischoff, “ROS#,” Dec. 2019. [Online]. Available: <https://github.com/siemens/ros-sharp/releases/tag/v1.6>
- [21] X. Long, M. Wonsick, V. Dimitrov, and T. Padir, “Task-oriented planning algorithm for humanoid robots based on a foot repositionable inverse kinematics engine,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 1114–1120.
- [22] —, “Anytime multi-task motion planning for humanoid robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4452–4459.

BIOGRAPHY



Murphy Wonsick is a PhD candidate in the Electrical and Computer Engineering department at Northeastern University. Murphy has received two Bachelors of Science degrees from Florida Institute of Technology, one in Computer Science and the other in Mathematical Sciences, and a Master of Science in Robotics Engineering from Worcester Polytechnic Institute. Murphy’s current areas of interest include human-robot interaction, heterogeneous human-robot teams, virtual reality, augmented reality, and user experience and interface design.



Taşkın Padir is an Associate Professor in the Electrical and Computer Engineering Department at Northeastern University. He received his PhD and MS degrees in electrical and computer engineering from Purdue University. He is the Director of Robotics and Intelligent Vehicles Research Laboratory (RIVeR Lab). He is also the Founding Director of the Institute for Experiential Robotics at Northeastern. His research interests include supervised autonomy for humanoid robots, shared autonomy for intelligent vehicles, and human-in-the-loop control systems with applications in exploration, disaster response, personalized in-home care, and nuclear decommissioning.