Multimodal Teleoperation of Heterogeneous Robots within a Construction Environment

Dylan Wallace[1], Yu Hang He[1], Jean Chagas Vaz[1], Leonardo Georgescu[1], and Paul Y. Oh[2]

***Abstract*— Automation in construction continues to be a topic of interest for many in industry and academia. However, the dynamic environments presented in construction sites prove these tasks to be difficult to automate reliably. This paper proposes a novel method of teleoperation for multiple heterogeneous robots within a construction environment. The system is achieved by creating a virtual reality interface that allows an operator to control multiple robots both synchronously and asynchronously. Feedback is provided from an array of RGBD cameras, force sensors, and precise odometry data. The DRC-Hubo and Spot robot platforms are used for implementation and experimentation. Experiments include useful tasks for construction including item manipulation and item delivery of tools and components. Results demonstrate the feasibility of implementing the system in a construction environment, including trajectory comparisons, task learning curves, and successful multi-robot collaboration.**

***Index Terms*— Robots in Construction, Telerobotics, Virtual Reality, Humanoids, Quadrupeds, Human-in-the-loop.**

I. INTRODUCTION

A study by the McKinsey & Company consultancy found that compared to the manufacturing and agriculture industries, the construction industry has suffered in value-added growth and global productivity due to lack of innovation in construction practices and methodologies [1]. This lack of innovation has caused the construction industry to experience a severe labor shortage in the past decade [2]. In an effort to increase construction efficiency and encourage wage growth, automation can be utilized across the field of construction in conjunction with new technologies such as modular construction, large-scale 3D printing, and smart building information management (BIM). However, this automation is still an open problem when the dynamic environments of traditional construction sites are considered.

While automation has made substantial progress in the past decade, current robotic platforms show difficulty to automate tasks and roles in construction in a way that would allow for dynamic adaptation. The team previously competed in the 2015 DARPA Robotics Challenge (DRC) in which robots had to perform complex manipulation and locomotion semi-autonomously. The results demonstrated that the team's approach was largely dependent on pre-planned trajectories within ideal conditions [3], and even under these conditions movements were slow. The team concluded that more work was needed to ensure that manipulation by these semi-automated humanoid robots could handle tasks more dynamically. Introducing a human-in-the-loop control method for these construction robots allows them to perform safely in a dynamic construction environment, especially when working with other human workers.

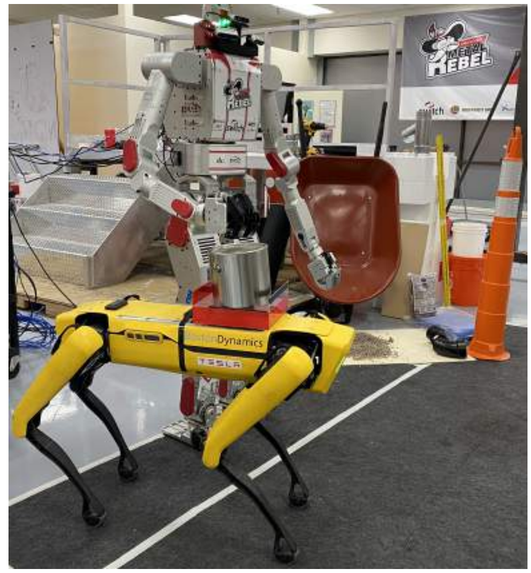

Fig. 1: Collaborative task between DRC-Hubo and Spot

Transporting and handling construction items is one of the most common forms of injuries [4]. These forms of injuries consist of sprains, back pains, cuts, and even amputations. Therefore, usage of humanoid robots operated through human-in-the-loop control could prevent injuries while still being advantageous because of their human form giving them the ability to maneuver in human-centered environment and handle human-operated tools. Furthermore, current methods of transporting materials to various places on the construction site require workers to navigate obstacles, as well as rough terrain. Instead, having an automated platform to transport materials from site-to-site can reduce workplace injuries. Traditional mobile robots used for transportation, such as automated guided vehicles, are severely limited by terrain and obstacles. Additionally, safety is paramount in a construction site. A robot that is not able to avoid obstacles would pose a serious threat to itself and human workers.

Companies such as Tesla see value in adding robots in the construction environment due to the benefits they could offer. The team has partnered with the company to automate building construction scanning using the Spot

[1]D. Wallace is with the Electrical and Computer Engineering Department at the University of Nevada Las Vegas (UNLV), USA. Direct all correspondence to `wallad3@unlv.nevada.edu`.

[1]Y.H. He, J.C. Vaz, & L. Georgescu are with the Mechanical Engineering Department at the University of Nevada Las Vegas (UNLV), USA.

[2]Paul Oh is with the Faculty of Mechanical Engineering, University of Nevada Las Vegas (UNLV), USA, IEEE Senior Member. `paul.oh@unlv.edu`

quadruped robot from Boston Dynamics, seen in Fig 1. The safety and flexibility offered by a quadruped robot such as Spot is on many levels similar to collaborative robots that are designed to work together with humans. Unfortunately, Spot has limited payload and does not currently have a manipulator attached, so it has limited capabilities in the construction environment. Therefore, it is advantageous to deploy a quadruped robot for item delivery and a humanoid robot for item manipulation in the construction field. This heterogeneous approach will allow for multiple different sets of tasks to be performed through a virtual teleoperation (VT) framework instead of through human workers.

The demand for teleoperation, in which an "avatar" receives commands and provides feedback to the operator, is expected to increase in the next decade. Improvements in communication, such as 5G, are making it possible for companies such as Doosan to teleoperate construction equipment [5]. These technological breakthroughs are providing operators with dynamic capabilities when teleoperating robotic platforms. The authors, seeing a demand for teleoperated robots in construction, are working towards the ANA Avatar XPRIZE Challenge. All Nippon Airways (ANA), one of many companies preparing for a future with teleoperation, has created the $10 million ANA Avatar XPRIZE competition as a way of driving humanoid telepresence technology forward. The scope of this competition encompasses numerous applications, with construction being just one of many. However, when combined with the team's previous efforts in the DRC and current work in construction inspection with Tesla, the XPRIZE challenge serves as an opportunity to test and develop new human-in-the-loop control methods for robots in construction.

II. Literature Review

Manufacturing and construction industries are trying to integrate humanoids into their day-to-day operations. However, hardware liability and limitation are an ongoing issue. For instance, Stasse *et. al.* [6] introduced the humanoid TALOS (prototype Pyrene uses electric motors) as an industrial friendly robot aimed for aircraft manufacturing environments. Additionally, the authors of that work chose the Gazebo/ODE and OpenHRP/AIST dynamics engine for low-level dynamic simulations and kinematic computations respectively. Three main simulations and experiments were conducted: holding weights with outstretched arms, walking, and climbing stairs. Such a robot could be deployed in industrial facilities where it can perform fairly limited tasks alongside human workers. These scenarios often present a high risk for humans to work with robots. Therefore, this paper seeks to improve teleoperation and robot-robot cooperation where the human worker would be in a safe, controlled environment, but still in full control of the robot(s).

More recently, there is a body of research involving the use of virtual reality equipment for teleoperation of humanoid robots in difficult manipulation tasks. Lipton *et. al.* [7] demonstrated that new users operating a virtual reality teleoperation system for the Baxter robot were faster and more accurate than an automation algorithm and direct control. Allspaw *et. al.* [8] noted that introducing 3D point cloud data into virtual reality significantly improved an operator's awareness of environment and overall performance. Penco *et. al.* [9] solved the problem of loco-manipulation tasks by introducing a low-level direct whole-body control and a high-level velocity control for locomotion. Lee *et. al.* [10] demonstrated that the introduction of virtual fixture into virtual reality teleoperation improve the accuracy of operator's control. In each of these works, virtual reality assisted operators were introduced to complete tasks that typically require human-level dexterity using humanoid robots. Thus, it is determined that virtual reality is desired for the teleoperation of a humanoid robot in dynamic environments. However, these state-of-the-art methods do not include operation of heterogeneous robots using their virtual teleoperation framework.

Improvements in virtual reality technology has allowed for research on teleoperation of robotic equipment on the construction site to become a subject of interest. Tanimoto, Shinohara, and Yoshinada created an architecture for teleoperating construction machinery that would increase efficiency [11]. Their research demonstrated that trajectory control was more accurate than manual control, especially since the operator did not have 3D visual feedback of the environment.

To achieve teleoperation of a humanoid robot in a dynamic environment, the operator often needs synchronous control of manipulators to accomplish dexterous tasks. Therefore, a swift and robust algorithm for calculating the inverse kinematics of manipulator(s) is required. Traditional Jacobian based numerical methods for solving inverse kinematics have various disadvantages that limit their application in real time robotic systems. In particular, the Jacobian method often suffers from local extrema configurations and the singularity problem [13] [14]. This leads to failure of the algorithm to converge on an optimal configuration. In this regard, Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) offer better solutions because they are largely unaffected by the kinematic configuration of the robot [15] [16] [17].

The core concept behind GA is to replicate biological evolutionary processes by iteratively refining a population of individuals (potential solutions) through a fitness or objective function, and then performing genetic crossover on the surviving individuals to populate a more qualified population [18]. Similarly, PSO simulates the behavior of a flock of birds landing on a small field by adjusting each solution with gradients of global best and personal best solutions using a fitness function [19]. The two iterative algorithms can be combined to quickly converge on the solution of an inverse kinematics problem. In this paper, the GA and PSO are preferred over traditional Jacobian method due to their versatility in incorporating additional parameters such as self-collision avoidance and velocity & acceleration limits to achieve real-time control that is capable of operating in a complex environment.

III. METHODOLOGY

This section discusses the methodology used in inverse kinematics for the synchronous operation of a humanoid robot using the virtual teleoperation (VT) framework. Additionally, an overview of the VT framework and its respective components is given.

A. Upper Body Kinematics

The manipulation of the upper body for the humanoid robot is controlled through the VR operator by utilizing an inverse kinematics solver. The inverse kinematics method is implemented as a means to decrease the arm end effector trajectory error. This is possible by using the Jacobian transformation matrix, which can be obtain by the pseudo-inverse method and damped least squares method (DLSM) [14]. The later offers the faster convergence time, and thus it is chosen for this work in order to minimize system latency. Considering the given task Jacobian $\dot{x} = J\,\dot{\theta}$, the state motion rate can then be calculated by:

$$\dot{\theta} = J^{+}\,\dot{x} - \alpha\,N\,W^{-1}\left(\frac{\partial O}{\partial q}\right)^{T} \tag{1}$$

Where $\dot{\theta}$ is the angular velocity vector for all joints, α is the damping factor, N is the projection matrix within the null space, W is the weighted matrix, and O is an optimization criteria. Moreover, the weighted generalized pseudo-inverse of the calculated Jacobian is given by J^{+}:

$$J^{+} = W^{-1}\,J^{T}\,\left(J\,W^{-1}\,J^{T}\right)^{-1} \tag{2}$$

$$N = I - J^{T}\,J \tag{3}$$

where I is the identity matrix. Next, since the Jacobian matrix depends on angular and linear velocities, consider the velocity vector:

$$\dot{x}_{\text{d}} = \left(\dot{x}_{\text{la}}\,\omega_{\text{la}}\,\dot{x}_{\text{ra}}\,\omega_{\text{ra}}\right)^{LI} = J_{\text{arm}}\,\dot{\theta} \tag{4}$$

where $\dot{x}_{\text{d}}$ is the desired velocity, and the subscripts la and ra stand for left and right arm respectively. Additionally, LI and RT represents linear and rotational velocities respectively. The upper body rotation Jacobian can be represented by:

$$\begin{pmatrix} {}_{G}\dot{x}_{\text{ub}} \\ {}_{G}\omega_{\text{ub}} \end{pmatrix} = \begin{pmatrix} {}_{G}J_{\text{LI,ub}} \\ {}_{G}J_{\text{RT,ub}} \end{pmatrix}\,\dot{\theta} \tag{5}$$

where G is the vector from the global coordinate system based on the inertial frame. For simplicity, the robot is assumed to not be moving the lower body during manipulation, and thus the inertial frame is the body frame located at the CoM of the humanoid. By computing the rotation and linear matrices for both arms, the final Jacobian for the upper body can then be determined. However, this solution creates a singularity when $J^{T}\,J$ gets close to zero. To solve this problem, Wampler [12] proposed a correction factor cf commonly known as the damped least squares method (DLSM) to minimize $||\dot{x} - J\,\dot{\theta}||^{2} + \alpha\,||\dot{\theta}||^{2}$ (residual error). This solution also has the added benefit of eliminating the need to apply the correction factor α, since no singularity occurs.

$$cf(\theta) = \sqrt{det(J^{T}\,J)} \tag{6}$$

Hence, when cf approaches to zero, α is then adjusted via

$$\alpha = \begin{cases} \alpha_{0}\,(1 - t/t^{s}) & if \quad t < t^{s} \\ \quad else & \quad 0 \end{cases} \tag{7}$$

Thus, t^{s} being the threshold value, α_{0} denotes the damping factor at the singular points. Finally, α filters singular and non-singular instances.

B. Genetic Algorithm and Particle Swarm Optimization

When genetic algorithms are applied to solving the inverse kinematics problem, the genes of individual solutions within the population consists of a set of joint angles for the kinematic configuration of the manipulator. Contrary to the Jacobian based method, joint limits can be easily implemented by limiting the range of random joint angle generation for a manipulator with n degrees of freedom.

$$\begin{gathered} x = [\theta_{1}, \theta_{2}, ..., \theta_{n-1}, \theta_{n}] \\ \theta_{min} \leq \theta_{i} \leq \theta_{max} \quad for \quad i = 1...n \end{gathered} \tag{8}$$

Once the population is randomly generated, a fitness/objective function is used to eliminated unfavorable solutions while maintaining the fittest individuals among the population. For an inverse kinematics problem, the fitness/objective function is based on the error between the desired and individual end effector position and orientation obtained through forward kinematics:

$$E = \omega_{p}||p_{t} - p_{i}|| + \omega_{o}(q_{t} \cdot q_{i}) \tag{9}$$

Where p_{t} and p_{i} represent the desired and individual end effector position and q_{t} and q_{i} represent the desired and individual end effector orientation expressed in quaternion format. ω_{p} is a normalization constant based on the total link length of the manipulator and ω_{o} is a normalization constant for end effector orientation.

The survivors of the selection process become parents of the next generation of individuals through genetic crossover where solution sets of two parents are randomly combined to create new offspring. An elitism phase can be implemented where the best solutions are directly imported into the next generation to improve convergence speed.

After genetic crossover, particle swarm optimization (PSO) can be implemented during creation of a new population to further increase the convergence rate.

$$\begin{aligned} \dot{\theta}_{i}(k+1) = w(t)\dot{\theta}_{i}(k) + w_{1} \cdot rand()[\theta_{p}(k) - \theta_{i}(k)] \\ + w_{2} \cdot rand()[\theta_{g}(k) - \theta_{i}(k)] \end{aligned} \tag{10}$$

$$\theta_{i}(k+1) = \theta_{i}(k) + \dot{\theta}_{i}(k+1) \tag{11}$$

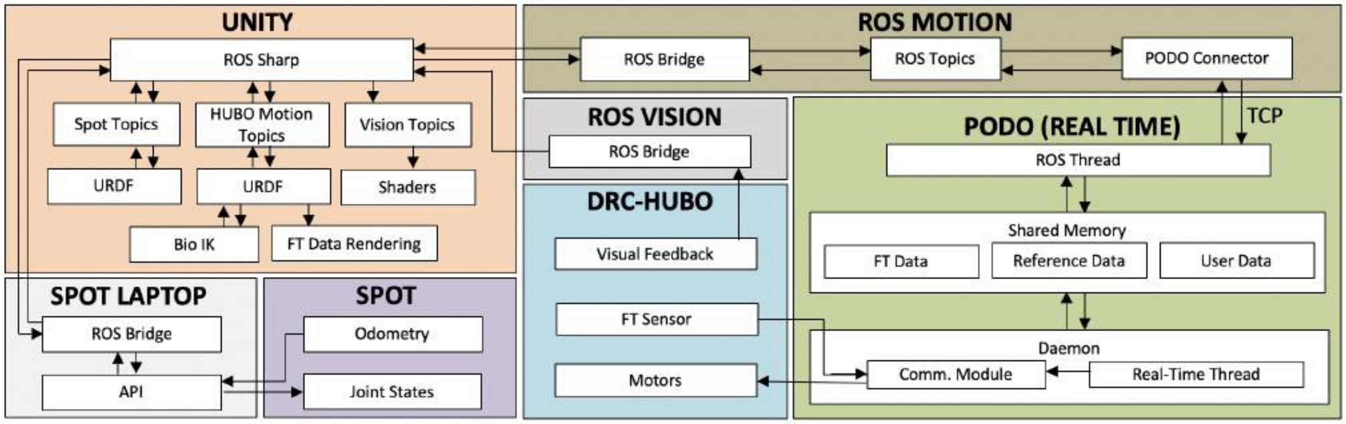

Fig. 2: System integration diagram illustrating the six main sectors; PODO, DRC-Hubo, Spot, Unity and ROS Motion/Vision

Where $\dot{\theta}_i(k)$ is the gradient of the i_{th} joint angle in the previous generation (parents), $\theta_i(k)$ is the previous i_{th} joint angle, $\theta_p(k)$ is the personal best joint angle, $\theta_g(k)$ is the global best joint angle, $w(k)$ is the inertia weight, w_1 and w_2 are weighting factors for personal and global gradient, and $rand()$ is a random number function between range [0, 1]. Through these gradient adjustment on the new population, the algorithm exploits PSO characteristics to improve the convergence rate. The evolved offspring combined with elites from previous generations forms a new population. The process terminates when the fitness function satisfies the criteria.

C. System Overview

The Virtual Teleoperation (VT) framework is spread across multiple different robots and computers with different operating systems, and thus needs to be designed agnostically to accommodate all current and future hardware. Furthermore, the VT framework is inherently designed to be used over a network, where all computers within the system framework can communicate effectively and in real-time. A diagram depicting the system architecture of the VT framework can be seen in Figure 2. The VT framework is split into 3 main systems:

1) Unity, the 3D interface for the virtual reality interface
2) ROS, the Robot Operating System
3) PODO, the real-time system for operating DRC-Hubo

1) ***Unity****:* Unity is a cross-platform engine used in a wide variety of industries, especially for virtual reality/augmented reality (VR/AR) development. Running on a Windows PC, the Unity program developed consists of a simulated URDF of the robot(s), a numerical IK solver on the upper body of the humanoid robot for manipulation, discussed in Section III-B, rendering of live pointcloud data from multiple RGBD cameras, waypoint navigation, and haptic feedback from forces during manipulation tasks. The main goal of the Unity interface is to act as a central "control room" for any number of robots being operated. The system allows for switching between modes of control (synchronous IK vs. asynchronous waypoint navigation), robots, and displays. This combination of control gives the operator a wide variety of control options that allow for multitasking, seamless teleoperation, and 3D view of live visual data.

2) ***ROS****:* The Robot Operating System (ROS) is an open-source framework for popular tools, software, and devices used in the robotics field. ROS was chosen as the main system to communicate between Unity and other robots due to its widespread use and adaptability for future solutions and experimentation. ROS runs separately on the motion computer (real-time) and the vision computer (NOT real-time) for the humanoid robot (DRC-Hubo), and also on the control laptop for the quadruped robot (Spot). The motion computer consists of a node that connects ROS to PODO through a TCP server and a node that connects ROS and Unity through rosbridge. The vision computer, which process depth images for manipulation, has one node that communicates between ROS and Unity. Additionally, the control laptop for Spot contains an instance of rosbridge that communicates with Spot's API for feedback and control.

3) ***PODO****:* PODO is the real-time system developed by KAIST HUBO Lab & Rainbow Robotics for operation of DRC-Hubo. PODO runs on the motion computer (real-time) and consists of a daemon that manages real-time operation of joints through CAN bus, as well as on-board sensors such as inertial measurement unit (IMU) + fibre-optic gyroscope (FoG) and force-torque (FT) sensors. The daemon updates sensor data and writes to the CAN bus in real-time (200 Hz) through the shared memory structure. This shared memory can then be used by individual programs (ALs) within PODO to control actions such as manipulation, walking, etc. To receive data from ROS, a special thread is created through TCP which then writes data to the shared memory in real-time. A specific AL was developed for VT framework to handle manipulation and finger control. In this AL, trajectories are generated based upon desired joint angles from shared memory. Since the IK Solver is based in Unity, processing for manipulation is offloaded from the on-board PODO computer.

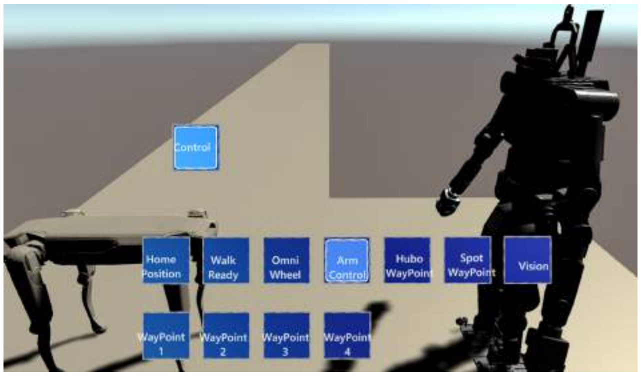

Fig. 3: Operator GUI with Spot & DRC-Hubo commands

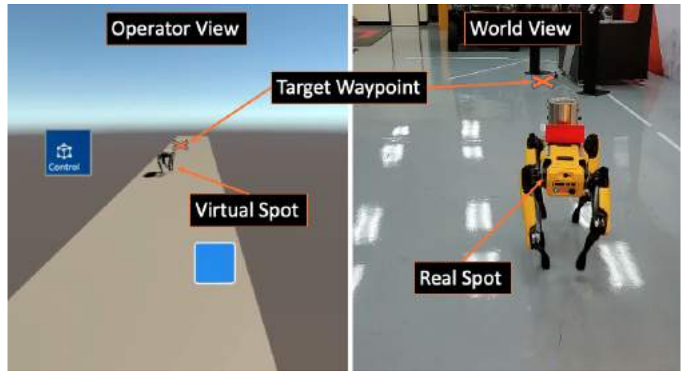

Fig. 4: Spot going to waypoint placed by operator

*4) **Spot API**:* The Spot API released by Boston Dynamics is written in the Python programming language and uses Google's remote procedure call API (gRPC) to communicate between the Spot robot and the developer's workstation. A custom GUI application was programmed to communicate between Unity and the developer's workstation through rosbridge, and between the workstation and the Spot robot through the Spot API. The odometry and joint states are transmitted to Unity to animate a virtual representation of Spot. Waypoint data is sent from Unity to the workstation to command Spot's locomotion through trajectory commands.

*5) **Image Processing**:* Post-processing filters are applied to the depth images obtained via the Intel Realsense D435 Cameras to improve the versatility and smoothness of the reconstructed data. One dimensional edge-preserving spatial filtering is applied to the depth image using 2 iterations with 20 discrete step-size boundary and 0.5 factor in an exponential moving average. A simple 4.0 meter clipping distance was employed to eliminate unwanted data. Finally, the depth image was aligned with the color image using extrinsic and intrinsic camera parameters. To optimize bandwidth usage, the depth image update rate was constrained to 6 FPS. Instead of sending pointcloud data over the network, a custom shader in Unity is used to reconstruct and render 3D pointcloud data from the color and depth images separately. This greatly reduces the bandwidth required to stream vision data from the robot to the VR interface.

IV. Hardware

Due to the multimodal nature of the VT framework, a variety of different hardware platforms can be used for input/output, processing, and actuation. The robot platforms used in the current implementation and experiments are the DRC-Hubo humanoid robot developed by Rainbow Robotics & KAIST HUBO Lab, and the Spot quadruped robot from Boston Dynamics. The VR platform used is the HTC VIVE. These systems are not essential to the working order of the VT framework, and thus they could be replaced with other VR or robot platforms in the future for further development and experimentation. Additionally, a custom vision system was made to augment the previous capabilities of DRC-Hubo, described in Section IV-D.

A. HTC VIVE

The main platform used for rendering the virtual haptic and visual data is the HTC VIVE virtual reality system. This system contains a 6 DoF headset as well as 2 hand controllers that also have 6 DoF tracking as well as rumble motors for haptic rendering. The display on the VIVE sports a resolution of 1080 x 1200 pixels per eye (2160 x 1200 total resolution) with a refresh rate of 90 Hz and a FoV of 110°. The VR headset is rendered using a custom PC with 4-core i7 CPU, GTX 980Ti GPU, and 16 GB RAM.

B. Spot

Spot is a quadruped robot developed by Boston Dynamics. The system allows for dynamic four-legged walking which lends itself well to a construction environment with unpredictable surfaces. The five on-board RGBD cameras provide Spot with 360°field of view and depth awareness that allow Spot to avoid obstacles during locomotion. The robot has the ability to carry up to 14 kg of payload. Spot's usage is mainly designed as a "blackbox" delivery system that handles obstacle avoidance and local navigation on-board.

C. DRC-Hubo

DRC-Hubo is a humanoid robot designed for the DARPA Robotics Challenge (DRC) in 2015, where it came in 1st place with Team KAIST and 8th place with team DRC-Hubo@UNLV [3]. The robot has 27 total degrees-of-freedom (DoF), including 7 DoF per arm, 6 per leg, and 1 for the waist. The hands on the robot consist of 3 fingers each and are controlled at once at a constant speed. DRC-Hubo also contains many state-of-the-art sensors such as force-toque (FT), IMU, fiber-optic gyroscope (FoG), and encoders on every joint.

D. Vision Head

For the operator to manipulate objects dynamically through DRC-Hubo, dense visual feedback is a necessity. The vision head is designed to be mounted on Hubo's head and consists of three Intel Realsense D435 Cameras that provide both RGB and depth images. Due to the limited FoV of the RGBD cameras, multiple cameras are used at varying angles. Two cameras on each side expand the FoV laterally. The third camera is mounted orthogonal to the robot's head.

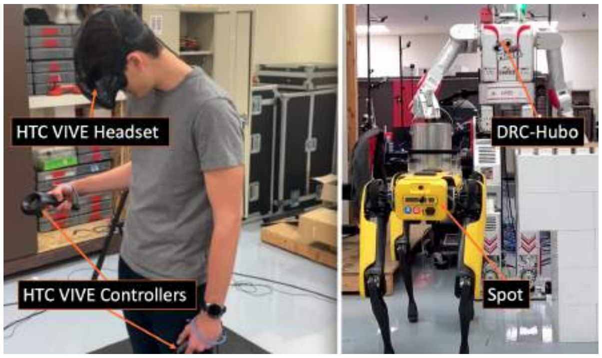

Fig. 5: Experimental setup: operator (left) & robots (right)

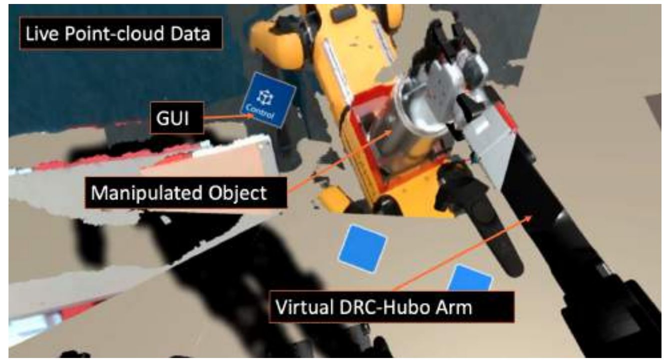

Fig. 6: Operator view during collaborative robot task

A sample of the 3D visual data observed by the operator through the vision head can be seen in Figure 5.

V. Experimentation

When the operator enters the VT framework through the VR headset and controllers, they are greeted with a control display that allows the selection of multiple control options, as seen in Figure 3. The operator has the option to toggle visual displays from 3 different RGBD cameras, and can see the current joint states and relative positions of the DRC-Hubo and Spot through their virtual representations. The operator can then choose to control Spot, DRC-Hubo, or both through a selection of buttons in the control display. When the operator enables synchronous control of DRC-Hubo, they can control the end-effector position and orientation of the robot model using the VR controllers. Spot can be controlled through waypoints placed by the operator using the VR controllers or pre-programmed waypoints using the control display, as seen in Figure 4. Spot's on-board obstacle avoidance and navigation handle any obstacles within the environment, and thus these can be ignored by the operator.

To effectively measure the performance of the VT framework and the implementation on both the DRC-Hubo and Spot, multiple experiments were designed to demonstrate the unique features of the system applied to a construction environment. During all experiments, force data and joint angles of both the simulated (Unity) and actual (DRC-Hubo) robots were recorded. Additionally, video recordings were taken of the operator in the VR space, the view of the operator's VR headset, and a view of the robot(s) for further review in the results and video comparison of the work. Figure 5 shows an example of the operator performing an experimental task with visual feedback included.

The first experiment consists of 5 different operators, with varying levels of ability, synchronously controlling the DRC-Hubo through the VT framework to grasp 2 PVC pipes in front of it, put the pipe fittings together, and then place the new part onto the spot for delivery. Each operator performed 5 separate trials for this experiment. This experiment was mainly designed to test the two-handed manipulation capabilities and to demonstrate the dynamic capabilities of the system as the operator adjusts to a task.

The second experiment assesses the validity of multimodal teleoperation of heterogeneous robots. The task presented consists of the operator manipulating a paint bucket, summoning Spot to a placed virtual waypoint, placing the bucket onto Spot as seen in Figure 6, and finally designating the "drop-off" point for the bucket to be retrieved. These multimodal tasks were performed from the GUI. This experiment was designed to test the collaboration between synchronously controlled DRC-Hubo and asynchronously controlled Spot robot.

The third experiment performed exists to quantitatively assess the performance of the kinematics and trajectory generation for the VT framework. This experiment consists of the operator synchronously controlling the DRC-Hubo to draw on a whiteboard. This approach lends itself well to comparing trajectories between simulated (Unity) and actual (DRC-Hubo) robot models.

Finally, an experiment is provided to show the direction of the future work toward material handling and manufacturing. This experiment consists of the operator controlling DRC-Hubo synchronously to grab a drill from a table and drill a large hole in a piece of drywall. This task is also similar to one presented in the DARPA Robotics Challenge, and thus provides a good comparison to a mostly-automated approach to the task.

The authors invite the interested reader to go to the following url to see a demonstration of several trial runs: `https://youtu.be/QRNAWOvpGtw`.

VI. Results and Discussion

Results from the first experiment discussed in Section V demonstrate the ability for the VT framework to be used for two-handed manipulation of construction objects. Figure 7 demonstrates a qualitative comparison between the simulated (Unity) and actual (DRC-Hubo) joint states for the two-handed pipe manipulation task. This simulation is obtained via Matlab© using the measured encoder values of the appropriate joints, and calculating the upper body kinematics through the respective rotation Jacobians, discussed in Section III-A. A video comparison of these joint states, demonstrating the minimal delay in the VT framework, can be seen in the companion video linked in Section V.

The image data and force torque data was sent from DRC-Hubo to Unity with an average measured bandwidth of 10.7 MBps. Robot joint data was sent from Unity to PODO with an average measured bandwidth of 109 KBps. At maximum

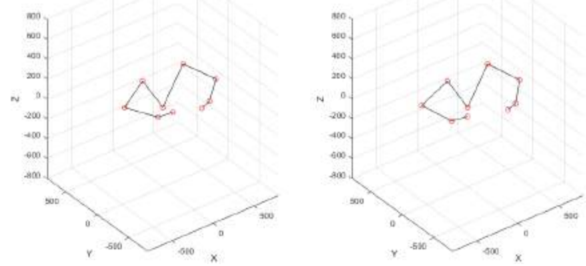

Fig. 7: Robot joint states: DRC-Hubo (left), Unity (right)

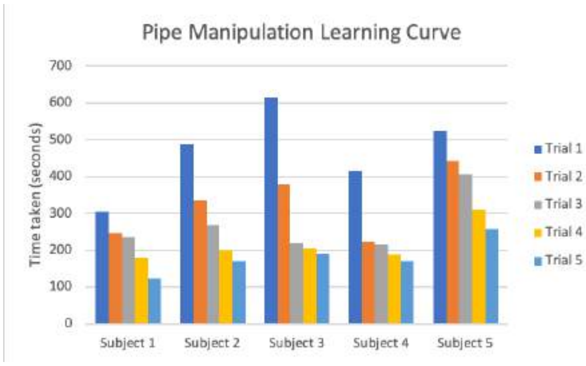

Fig. 8: Pipe manipulation task trial times

operation, the virtual reality environment was able to update each frame at 12.8 ms or 78 FPS on average with maximum latency at 16.9 ms or 59 FPS. The responsive update rate provided the operator with intuitive control and reduced the risk of motion sickness.

Even though the operator receives visual and haptic feedback, it still takes a few trials for the operator to consciously learn how to control the robot. As seen in Figure 8, there is a correlation between the number of trials and amount of time it takes to complete the tasks. The time drastically decreased across each subject as more trials were performed. Subjects 1, 2, and 4 were familiar with the robot platform the interface, and completed their first trial faster compared to Subject 3 who was completely unfamiliar with the architecture. However, Subject 3 improved their time significantly in next trials showing that there is a low learning curve. Subject 5 was instructed not to worry about timing, and instead to prioritize the safety of the robot. This resulted in a more gradual learning curve, but still demonstrates how quickly a subject can learn to use the VT framework. On average operators experienced a 60.8% reduction in task completion time, with a minimum decrease of 50.6% and a maximum decrease of 69.2%. As seen with all the subjects, it is thus expected that the operator will take some time to get used to the controls, but over time efficiency will increase significantly.

For manipulation tasks that require high levels of dexterity, the end-effector trajectory of the actual robot should closely follow the trajectory of the avatar controlled by the operator. The task of drawing geometric shapes on a whiteboard successfully reflects the precision between the virtual and physical robot's end-effector position. Figure 9 exhibits the end-effector trajectory of both virtual (Unity, green) and physical (DRC-Hubo, red) robot during the drawing of a rectangular shape on the whiteboard. The figure shows that the two sets of trajectories almost exactly overlay each other, suggesting that the trajectory of the physical robot follows the trajectory of the virtual robot with significant accuracy. The results of this experiment demonstrate that the VT framework is capable of manipulation tasks that require high accuracy and precision.

To demonstrate multimodal teleoperation of heterogeneous robots, the operator used DRC-Hubo to manipulate and Spot to deliver an item to a waypoint. Similar to the pipe manipulation experiments, a "learning curve" effect can be noted from the results. The operator improved the speed of delivery of the bucket by nearly a full minute. The advantage of multimodal teleoperation of heterogeneous robots is the ability of the operator to control distinctive robots to work collaboratively to accomplish tasks. The humanoid robot with its two manipulators is suited for object manipulation, while Spot, a quadruped robot, is better suited to traverse rough terrain. During the multimodal teleoperation, the operator has the ability to synchronously control DRC-Hubo's manipulators. At appropriate times, the operator can asynchronously summon the Spot robot to specific locations using custom waypoint markers or predetermined locations using waypoint commands in the GUI, as seen in Figure 4. The collaboration between the two robots allows an operator to accomplish a task that would be otherwise difficult or impossible for an individual robot to accomplish.

For an operator to perform actions without damaging the robot, haptic feedback must accurately represent the forces the robot is experiencing. Figure 10 presents the forces on Hubo's right arm as it is cutting the drywall with a drill. Since the operator is manually controlling the trajectory, the forces are not constant; instead, the operator must adjust for these based on haptic feedback. Forces should not pass the maximum recommended stress force of 60 N or the robot could be at risk of damage. Force data rendered to the operator via haptic vibrations to the HTC VIVE controllers, mentioned in Section IV-A. As the operator passes 10 N in any direction, the controllers will vibrate, indicating that Hubo is experiencing substantial forces. The system currently lacks the ability of providing the operator with knowledge when they pass the maximum stress force of 60 N. However, in future work, a method of "blacking-out" the screen of the operator may be implemented to prevent damage to the robot with force of over 60 N.

VII. Conclusion And Future Work

This paper demonstrates the successful collaboration between heterogeneous robots using multimodal teleoperation. Results demonstrated the feasibility of the system as applied to a construction environment, and especially demonstrated the ability to control multiple heterogeneous robots through the interface at the same time. An operator in the interface can control DRC-Hubo synchronously through the framework, but is also able to control Spot asynchronously. The

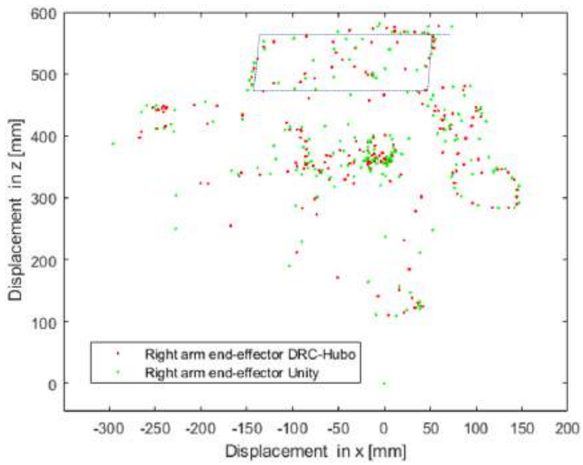

Fig. 9: Right arm trajectory for whiteboard drawing

integration of multiple RGBD cameras to stream live 3D data from different angles proved to be essential to the success of the implementation. A construction environment could benefit from implementing this technology due to increased efficiency and reduced injuries.

Future work will consist of using an anthropomorphic hand with independent finger control to improve the manipulation of compliant construction materials. Visual feedback can be further improved by adding roll, pitch, and yaw control to the vision head mentioned in Section IV-D. Lastly, an attempt was made using a drill to cut a circle through the drywall. Improving trajectory control of the current framework to account for delay would increase the accuracy of the cut while adding more realistic haptic feedback would reduce potential damage to the robot or the object. These will be the main topic of the future work toward material handling and manufacturing using the VT framework with DRC-Hubo and Spot.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of Matt Jezyk and Chris Reilly from Tesla on the research opportunity with Boston Dynamics' Spot robot, and contributions from Blake Hament, Joowon Lee, and Akshay Dave at DASL-UNLV.

REFERENCES

[1] "The Construction Industry's Productivity Problem." The Economist. The Economist Newspaper, August 17, 2017. https://www.economist.com/leaders/2017/08/17/the-construction-industrys-productivity-problem.

[2] Cilia, Juliette. "The Construction Labor Shortage: Will Developers Deploy Robotics?" Forbes. Forbes Media LLC., July 31, 2019. https://www.forbes.com/sites/columbiabusinessschool/2019/07/31/the-construction-labor-shortage-will-developers-deploy-robotics/.

[3] P. Oh, K. Sohn, G. Jang, Y. Jun and B.K. Cho, "Technical Overview of Team DRC-Hubo@ UNLV's Approach to the 2015 DARPA Robotics Challenge Finals,"*In Journal of Field Robotics,* pp.874-896, March, 2017.

[4] "Top Work-Related Injury Causes." Injury Facts. National Safety Council, 2018. https://injuryfacts.nsc.org/work/work-overview/top-work-related-injury-causes/.

[5] "Doosan First To Use 5G for Worldwide 'TeleOperation'." Doosan. April 5, 2019. https://eu.doosanequipment.com/en/news/2019-28-03-doosan-to-use-5g

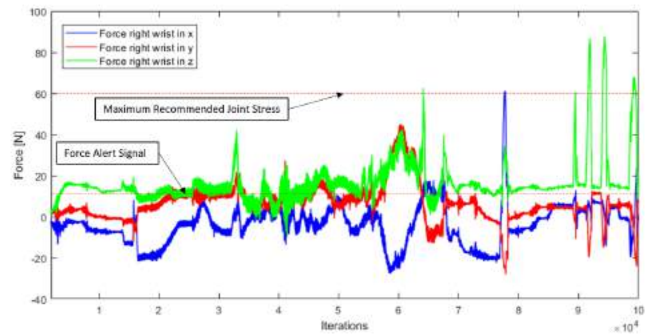

Fig. 10: FT analysis for drilling task

[6] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Mansard, N. Lamiraux and J.P. Laumond, "TALOS: A new humanoid research platform targeted for industrial applications,"*In International Conference on Humanoid Robotics (Humanoids),* pp. 689-695, Birmingham, UK, November, 2017.

[7] Lipton, Jeffrey I., Aidan J. Fay, and Daniela Rus. "Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing." IEEE Robotics and Automation Letters 3, no. 1 (2017): 179-186.

[8] Allspaw, Jordan, Jonathan Roche, Adam Norton, and Holly Yanco. "Teleoperating a humanoid robot with virtual reality." In Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for Human-Robot Interaction. 2018.

[9] Penco, Luigi, Nicola Scianca, Valerio Modugno, Leonardo Lanari, Giuseppe Oriolo, and Serena Ivaldi. "A Multimode Teleoperation Framework for Humanoid Loco-Manipulation: An Application for the iCub Robot." IEEE Robotics Automation Magazine 26, no. 4 (2019): 73-82.

[10] Lee, Donghyeon, and Young Soo Park. "Implementation of Augmented Teleoperation System Based on Robot Operating System (ROS)." In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5497-5502. IEEE, 2018.

[11] Tanimoto, Shinohara, and Yoshinada. "Research on effective teleoperation of construction machinery fusing manual and automatic operation." Robomech J4, 14 (2017).

[12] C.W. Wample, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,"*In IEEE Transactions on Systems, Man, and Cybernetics,* pp.93-101, January, 1986.

[13] Duleba, Ignacy, and Michał Opałka. "A comparison of Jacobian-based methods of inverse kinematics for serial robot manipulators." International Journal of Applied Mathematics and Computer Science 23, no. 2 (2013): 373-382.

[14] Chiaverini, Stefano, Bruno Siciliano, and Olav Egeland. "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator." IEEE Transactions on control systems technology 2, no. 2 (1994): 123-134.

[15] S. Starke, N. Hendrich, D. Krupke and J. Zhang, "Evolutionary multi-objective inverse kinematics on highly articulated and humanoid robots," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 6959-6966.

[16] Aguilar, Omar Alejandro, and Joel Carlos Huegel. "Inverse kinematics solution for robotic manipulators using a cuda-based parallel genetic algorithm." In Mexican International Conference on Artificial Intelligence, pp. 490-503. Springer, Berlin, Heidelberg, 2011.

[17] Falconi, Riccardo, Raffaele Grandi, and Claudio Melchiorri. "Inverse kinematics of serial manipulators in cluttered environments using a new paradigm of particle swarm optimization." IFAC Proceedings Volumes 47, no. 3 (2014): 8475-8480.

[18] Holland, John Henry. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4.