

# A MIXED REALITY HUMAN ROBOT INTERACTION INTERFACE USING HAND TRACKING AND STEREO VISION FOR CONSTRUCTION APPLICATIONS

Damith Tennakoon<sup>1</sup>, Mojgan Jadidi<sup>1</sup>, and SeyedReza RazaviAlavi<sup>2</sup>

<sup>1</sup>York University, Toronto, Canada

<sup>2</sup>Northumbria University, New Castle, UK

## Abstract

The modern day construction site is able to complete complex tasks using highly skilled on-site workers and robust machinery. On-site workers face severe physical injuries from unforeseen accidents due to construction site dynamics. This project proposes and implements a mixed reality human robot interaction system, utilizing stereo vision systems and hand tracking algorithms, to teleoperate a robot arm within a simulation. Results demonstrate enhanced depth and situational awareness in the robot workspace versus traditional teleoperation. This research allows operators to safely perform complex construction tasks remotely with a robot arm using a head-mounted display.

## Introduction

The modern day construction site completes large-scale tasks by training on-site workers and utilizing heavy-duty machinery. However, success is not achieved with efficiency or safety as a major priority since construction work is prone to surpassing timelines and facing potential delays due to low productivity rates Davila Delgado et al. (2019). On-site workers are often subject to serious health consequences, that are frequently overlooked. Some of these hazards include exposure to high concentrations of particulate matter, workers colliding with heavy machinery, and falls off unsecured high altitude sites Yang et al. (2023); Castro-Lacouture (2009).

Various industries continue to adopt automation techniques, leveraging robotics solutions, to improve task efficiency, productivity, and product quality. For instance, the manufacturing industry utilizes robotics to automate assembly lines, with the advent of performing repetitive, hazardous, and time-constrained tasks with ease, to produce high volumes of products. Not only are robotics solutions very efficient, but are also highly competent with precision-seeking tasks as made evident by their applications within the medical field. The healthcare industry deploys robots to perform surgical procedures with the advent of providing minimally invasive procedures and higher precision than human surgeons. The advantages of uninterrupted work is highlighted by the application of robotics in agriculture where tasks including planting, harvesting, and monitoring crops are automated and carried out by multiple robots Campilho and Silva (2023). Huang and Sakurai (1990) concluded that the implementation of factory automation (FA) for well-established automotive companies, including Mitsubishi, Nissan, and Mazda, dis-

played a noticeable reduction in labor costs, an increase in vehicle quality, and increased flexibility for adapting tasks. An extensive study performed on the implications of introducing industrial robots into manufacturing firms showed increased worker safety; specifically, 1 standard deviation of an increase in robots resulted in a 4% reduction in physical job intensity and a 5% reduction in job worker disabilities Gihleb et al. (2022).

With the clear benefits that automation and robotics provide, many attempts have been made to integrate this technology into the construction industry. A study demonstrated the successful implementation of robots into performing construction tasks require the adaptation of robots to the unpredictable and dynamic nature of construction sites. This has been achieved in modern robotics in two ways: (1) pre-programming robots to handle many different scenarios and rigorously testing them prior to deployment and (2) by training robots, using sophisticated real-world datasets and machine learning techniques, for on-site applications. The fast-paced and dynamic construction environment prevents either of these requirements from being met; pre-programmed robots are not able to handle unforeseen tasks or changing tasks and quality training data for construction applications are limited Zhang et al. (2023). Mukherjee et al. (2022) suggests a more intuitive method to deploy robotics into the construction sector is to create a human-robot collaboration (HRC) system that integrates the on-site worker's experience with the advantages of autonomous robots, to increase task productivity and reduce labor shortage. Advent of such as system alleviates the need for dynamic robot programming and multiple task execution.

In order to perform any intricate task, a complex manipulator is required. Robot arms are used to interact with and manipulate objects within an environment via various teleoperation methods. Teleoperated robots are specially useful where the robot's physical target workspace is challenging or impossible to access. The most commonly used control interfaces for these instances are keyboards, joysticks, and portable touch screen devices. However, these traditional modes of teleoperation are ineffective as the control interfaces are not intuitive. One of the emerging solutions for this problem is the use of mixed reality technologies which facilitate full immersion into the robot's workspace, creating a more natural human-robot interaction (HRI) interface Meng et al. (2023); Wonsick and Padir (2020a). Mixed reality (MR) is a combination of augmented real-

ity (AR) and virtual reality (VR) technologies. MR combines the real and virtual worlds to construct a hybrid environment where interactable 3D virtual models exist Aloqaily et al. (2023). The advent that mixed-reality HRI interfaces provides over traditional teleoperational methods is the ability to view and interact with robots in 3D environments Wonsick and Padir (2020a). The ability to view in 3D is due to the emergent properties of 3D cameras and immersive head-mounted displays (HMDs). Compared to 2D video-based visualization, 3D real-time visualization coupled with HMDs has shown to improve localization, maneuverability, and control in robotics applications Wonsick and Padir (2020b).

In this paper, we introduce a novel HRI interface that leverages MR technology, state-of-art hand tracking methods, coupled with stereo vision systems to teleoperate a 4-DoF robot arm in an immersive first-person view (FPV) using a Meta Quest 2 HMD. By providing a mode to operate a robot arm using hand gestures from a safe distance to the manipulator, on-site construction workers can perform complex tasks, utilizing their experience, while taking advantage of the strength, precision, and deployability of robots.

## Related Works

Much research and development has accumulated over the past few years specifically on leveraging AR, VR, and MR technologies to effectively improve human-robot interaction and collaboration. Puljiz et al. (2019) proposes and implements an HRI interface to teleoperate an industrial grade robot arm utilizing the Microsoft HoloLens AR HMD. The approach presented utilizes the built-in spatial mapping of the HoloLens to identify the target robot arm, and its end effector. This is achieved by storing a previously known model of the robot arm and matching the known model with the detected mesh of the spatial mapping output. The known mesh and the HMD is localized with the detected robot arm, of the physical space, and hand detection techniques are used to translate the end effector. This interface enables the teleportation of a robot arm without the need for physical interaction with the manipulator.

Yim et al. (2022) emphasized the importance of full immersion into the robot workspace through means of stereo vision. They implement a method to teleoperate a 6-DoF robot arm by fix mounting an Intel RealSense stereo camera to the platform the robot arm is fixated on, with a view facing down and towards the end effector. The stereo images are streamed to a Meta Quest 2 VR HMD to replicate human vision. Hand held controllers are localized with the robot arm to manipulate its end effector. This HRI interface enables a robot to be controlled from a safe distance with human-like perception.

Chang et al. (2023) tackles on the ongoing challenge of effective robot workspace visualization and perception by their design of a 2D/3D multi-view teleoperated robot arm. Specifically, they have devised an approach to view a

robot's workspace via an FPV real-time 3D point cloud (PCD) model, from a stereo vision camera mounted to the end effector of the manipulator, and a 2D view from a global RGB camera. This effectively enabled for the visualization of the direct workspace, via the PCD, as well as the global workspace, via the RGB camera. The developed MR interface enabled the user to switch between the two visual systems in their HMD, using a handheld controller, providing a comfortable teleoperation process.

Many other similar approaches have been proposed and implemented showcasing positive results towards achieving a natural HRI interface. Though, these HRI systems do not come without their own setbacks. With the implementation of the HoloLens AR approach by Puljiz et al. (2019), though the interface provides robot arm teleoperation without required physical interaction with the manipulator, it is constrained by the requirement of a physical operator needing to be present near the robot's workspace. This limits the control bandwidth as the operator needs to have a direct line-of-sight of the robot arm for teleoperation. In the case of the immersive stereo vision teleoperated robot arm developed by Yim et al. (2022), many safety issues arise when the user's view is obstructed by the robot arm itself or other objects, within the robot's workspace, due to the vision system being fixed mounted. Chang et al. (2023) is on the right track with their multi-view MR HRI interface, however, the cognition load required for the operator increases when frequently needing to switch between a 3D view and a fixed 2D view; the operator will need significant training to be comfortable localizing their coordinate system with the robot arm's coordinate frames, in the two views. Many lessons were taken away from all aforementioned approaches and has aided the development of the HRI interface discussed in this paper.

## Methodology

### Approach

The design of the HRI system that is being developed revolves around creating an immersive, hands-free, and safe interface to teleoperate a robot arm towards performing construction tasks. There are 4 main categories that are required to achieve this interface; (1) stereo vision system to replicate human-like perception; (2) immersive wearable technology to construct an immersive MR interface; (3) real-time hand-tracking algorithms for intuitive teleoperation; (4) tracking systems to provide real-time operator situational awareness.

The stereo vision system used for this project is the ZED Mini (ZEDM) stereo camera, developed by StereoLabs. Stereo vision is a visualization system that enables 3D point extraction of an environment, through images obtained via two cameras separated by a fixed distance. The general concept is founded on the basis of finding the disparity between the two images to extract the missing depth information, through means of geometry DANDIL and ČEVÍK (2019). The ZEDM camera replicates the human-

eyes, enabling real-time depth perception, with extreme accuracy. The approach is to mount the ZEDM camera on top of the end effector of the 4-DoF (degrees of freedom) robot arm. The left and right videos will be streamed, in real-time, to the left and right displays of the Meta Quest 2 VR HMD. This enables the operator, wearing the HMD, to perceive the robot's point-of-view (POV) with human-like vision. Open source libraries utilize the 4 on-board cameras of the HMD, used for spatial mapping, to track the user's hands in real-time. The tracked hand models are superimposed to the FPV stereo view on the HMD. To provide full situational awareness, the ZEDM camera is mounted on a servo actuated two-axis gimbal that tracks the orientation of the HMD, in real-time. As the operator wearing the HMD looks around, the servo motors are actuated to map the HMDs look angles with the ZEDM camera. This will provide a natural one-to-one orientation system enabling full situational awareness of the robot's workspace. The ZEDM vision system has a noteworthy advantage of facilitating pass-through reality, allowing seamless integration of 3D virtual models into the real-time stereo images. This is a fundamental requirement for this HRI system since the robot arm will be controlled via an MR control sphere; in the FPV view of the HMD, a translucent spherical virtual object will be fixated onto the visualized end effector, using pass through reality. The operator will be able to place their hands into this tracking sphere towards the desired direction for the robot to move towards. Hand gestures, such as pinching and releasing two fingers together, will be detected to provide gripper controls.

This paper covers the development of the proposed architecture, in a simulated environment, and preliminary results from experiments conducted, comparing traditional teleoperation methods with the developed MR HRI interface, when performing simple tasks.

### **Simulation Environment Setup**

The simulation environment chosen for this project was Unity Game Engine. Unity is a cross-platform game development engine that enables developers to build video games, simulations, and mixed reality applications, by utilizing 3D objects and scripting via the C# object-oriented programming language Haas (2014). Unity was the ideal candidate for this simulation as it provides open-source libraries for AR, VR, and MR development. It is also built on the capable NVIDIA PhysX 3.4 physics engine, providing the ability to run real-time physics simulations.

The first step in designing a real-time simulated robot arm was selecting a 3D model that represented a real-world industrial grade robot arm, namely the manipulators developed by KUKA. The model used for this simulation was retrieved from the open-source 3D modelling website Sketchfab, by a developer named Yuki. The 3D model features a 4-DoF robot arm, in FBX format. In the Unity environment, the design underwent many modifications to

organize the multi-component model into a structured hierarchical set of objects, where each rotating joint, connecting each of the links together, were aligned relative to a global coordinate system. To actuate the robot arm during run-time, a joint-space controller was scripted, enabling each joint to rotate to an input angle with a constant angular velocity. Although joint-space control for robot arm's are used in real-world scenarios, it is not representative of how robot arm's are often operated majority of the time. Rather, a target position vector, a 3D coordinate in space, is provided to the manipulator. The system then computes a set of ideal joint angles to achieve the desired end effector position. In robotics, this mathematical process is referred to as inverse kinematics, a key-part of robot trajectory planning. One of the challenges with inverse kinematics is for a given target end effector position, there may be infinite many solutions, leading to computation efficiency problems. Many open-source inverse kinematic solver libraries exist such as Ikpy library. Ikpy is a Python library that performs inverse kinematic solutions efficiently, given a kinematic chain, as a Unified Robotics Description Format (URDF) file, and a target position and orientation. A URDF file is a low level description file, written in the Extensible Markup Language (XML), which describes the geometry, mobility, and their corresponding relationships, of a robot. Utilizing the Unity 3D model, a URDF file was developed for the robot arm. To connect the Python inverse kinematic scripts to the Unity simulation environment, a Transmission Control Protocol/Internet Protocol (TCP/IP) server-client system was designed. TCP/IP is a data link protocol that enables the transmission & reception of messages, between devices, over an internet connection. A three-way handshake connection, between two devices, enables the host server to send packets of information to the client, where the client transmits a message back to the server, confirming data reception. TCP's error handling capability ensures reliable point-to-point communication, without data loss. Python TCP server scripts, integrating the inverse kinematic solvers, and the corresponding C# client scripts, were developed to establish a connection between the computing server and the simulation environment client. This simulation architecture is summarized in Figure 1 in which the 3D coordinates of a target object, in the Unity environment, is transmitted to the server which returns a set of joint angles to achieve the desired target position. The result is a 4-DoF robot arm that tracks and follows a target sphere object, in real-time. To further actualize the simulation, a prismatic gripper tool was designed, using the CAD modelling software Autodesk Fusion 360, and subsequently affixed to the end effector joint of the manipulator. To devise experiments relating to construction work, as discussed in later sections, 3D models of pallets and bricks were placed into the virtual environment. This simulated environment, ascribed in Figure 2, is an ideal representation of the physical robot arm, and its implementation, that is underway at the time of writing.

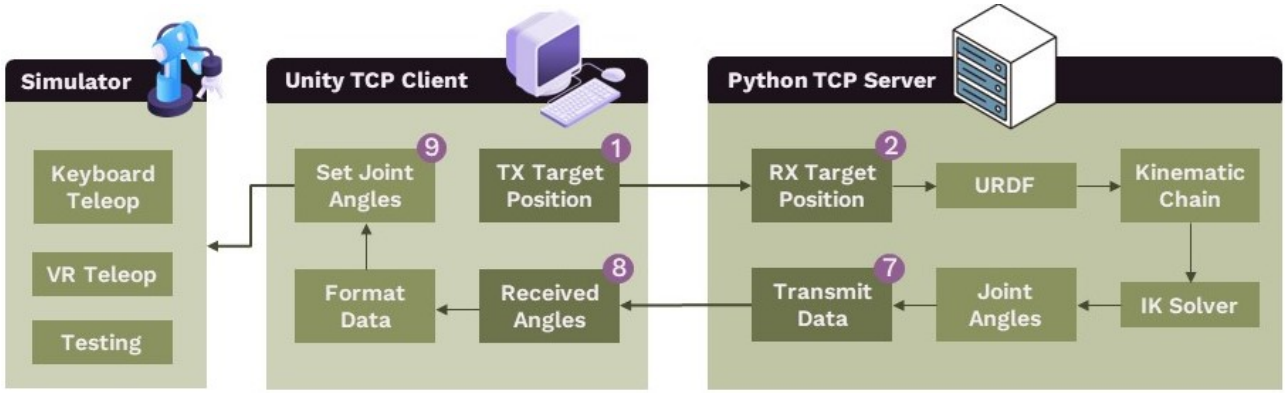


Figure 1: Real-time 4-DoF robot arm inverse kinematic simulation workflow diagram.

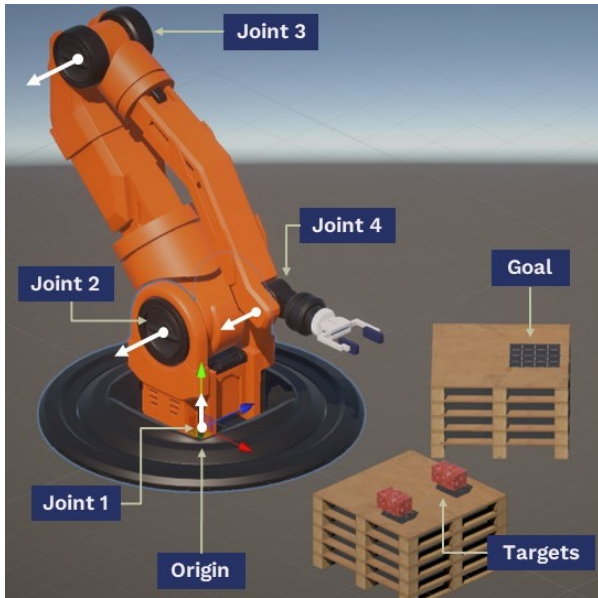


Figure 2: Simulated virtual environment with key labels.

### Traditional Teleportation Interface

Prior to developing the proposed MR HRI interface, traditional modes of robot teleoperation were simulated. One of the common modes a robot arm is controlled via keyboard inputs. The developed control algorithm enables the robot's end effector to translate along the horizontal and vertical, while the gripper can be manipulated via two separate keys, as documented on Table 1. The underlying control algorithm takes the input from the keys and translates the manipulator's end effector target vector, in 3D space, by at a constant rate. This effectively provides real-time end effector control via keyboard inputs. Teleportation is commonly coupled with visual input systems to provide a real-time view of the robot's workspace, when it is operating at an inaccessible setting. In the real-world, this is often done by fix mounting an RGB camera that has a direct line-of-sight to the robot's workspace. A monocular (single-lens) camera was integrated into the virtual environment, presented in Figure 3, to provide the operator with a view of the robot arm's workspace for teleoperation via keyboard inputs. This achieves a simulation of

a traditional teleoperation method which will be used for comparison with the MR HRI interface.

Table 1: End effector keyboard control inputs.

Horizontal	Vertical	Gripper
I, J, K, L	O, P	Q, E

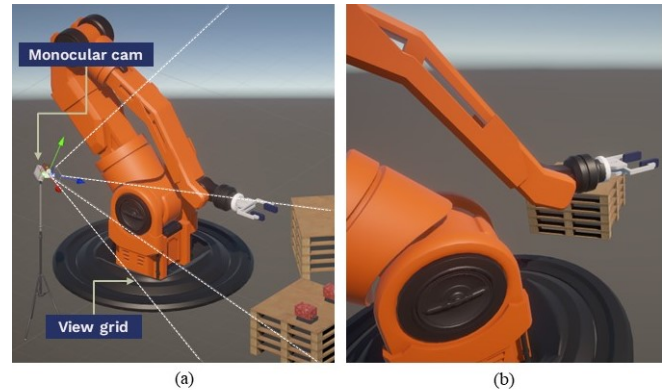


Figure 3: Monocular camera system: (a) Camera rig setup within the virtual environment; (b) Real-time view from the operator's computer screen during run-time.

### Mixed Reality Teleoperation Interface

The first sub-component of developing the proposed MR HRI interface was to simulate a real-time FPV stereo view from the robot arm's end effector, utilizing the Meta Quest 2 VR HMD. The ZEDM stereo camera, by StereoLabs, is able to provide human-like perception. This stereo camera system was integrated into the simulation by hierarchically attaching a 3D model of the camera to a CAD modelled, and maneuverable, servo actuated two-axis gimbal, shown in Figure 4. The objective of this system is to receive real-time orientation data, of the HMD, and map its rotations about the simulated servos y-z axis. This enables real-time head tracking of the operator, presented in Figure 5. Natural head-tracking is of significance as MR wearable technology is prone to creating discomfort for operators. This occurs when the physical movements of the operator do

not align with their sensory input systems (vision, sound, touch). As a result, constructing a one-to-one human-like vision system was a key priority.

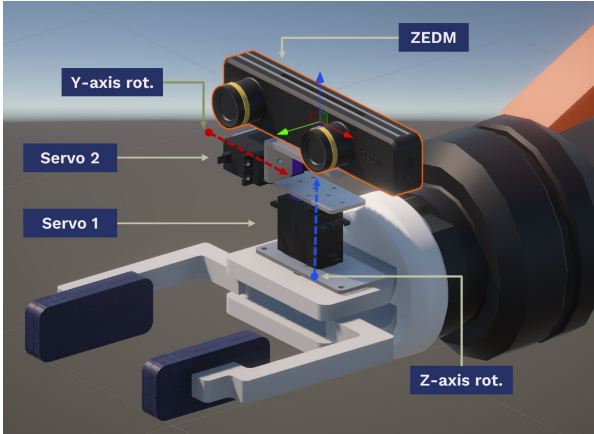


Figure 4: Simulated head tracking 2-axis gimbal system for ZEDM camera.

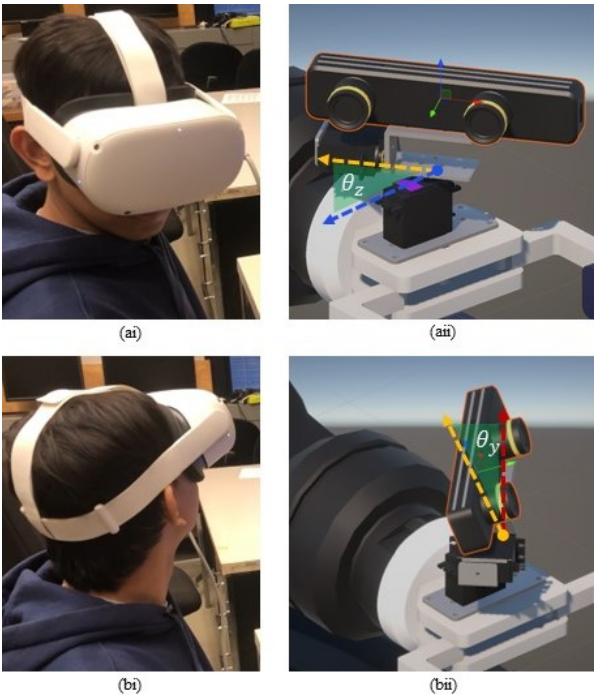


Figure 5: HMD head tracking system during run-time: (ai) operator turning their head towards the right, rotating about the z-axis by angle  $\theta_z$ ; (aii) 2-axis gimbal rotating the z-axis servo by corresponding angle; (bi) operator turning their head upwards, rotating about the y-axis by angle  $\theta_y$ ; (bii) 2-axis gimbal rotating the y-axis servo by corresponding angle.

The second sub-component was the integration of real-time hand tracking and gesture detection. The objective here was to provide a mode to manipulate the robot arm using nothing more than the operator's hands, which was achieved utilizing the open-source Oculus Integration package. This is a well-developed library, by Oculus, that provides many features for the HMD including access to IMU data, hand tracking, and spatial mapping. The hand tracking functions by utilizing the 4 onboard multi-camera

system of the HMD. Utilizing this library, a VR rig was setup in the virtual environment, with the integrated hand models, to render the tracked hand data in real-time. To perform gesture detection, such as pinch recognition, algorithms were written to measure the relative separation of individual joints in 3D space. Figure 6 diagrammatically shows the real-time hand tracking and how simple gestures are detected.

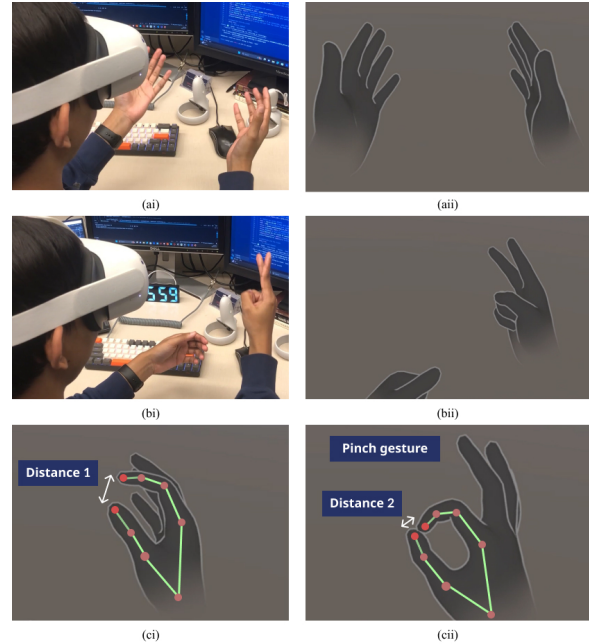


Figure 6: (ai, aii, bi, bii) demonstrates the implemented real-time hand tracking system with the Meta Quest 2 HMD. (ci, cii) demonstrates the pinch gesture being detected by measuring the separation between two joints.

To manipulate the end effector, the tracked hand models were fused with the stereo camera vision system. The result is a full situational aware FPV of the robot arm's end effector with the users tracked hand models. To move the robot arm using the hand tracking system, a Control Sphere was designed. The Control Sphere consists of two 3D virtual models of translucent spheres, one with a smaller radius than the other, anchored at the center of the larger sphere, spatially mapped onto the robot arm's end effector. The outer sphere, with radius  $R_2$ , is the Movement Sphere, and the inner sphere, with radius  $R_1$ , is the Gesture Sphere, described in Figure 7. When a tracked hand model is inside the Movement Sphere but outside the Gesture Sphere, a vector, from the 3D position of the hand model to the Control Sphere center, is computed and normalized. The resulting unit vector is indicative of the direction the operator desires to move end effector. The robot arm's target tracking algorithm is updated by taking the sum of the current target position with the product of the unit vector and a constant multiplier. The result is the end effector moving in the vector direction, relative to the center of the Control Sphere, as input by the operator's tracked hand models, providing omnidirectional control. When the tracked hand models are within the Gesture

Sphere, the tracking target's movement speed is set to zero, effectively pausing motion control. Users are then able to input hand gestures within this sphere, such as pinching the thumb and index fingers together, to close the gripper and releasing these fingers to open the gripper. Equation 1 computes the relative position vector between the tracked hand model to the center of the Control Sphere while equation 2 normalizes the position vector. In equation 3, the variable  $\Delta t$  corresponds to the elapsed time between the current and previous frame in the simulation while  $C$  is a constant multiple to set the movement speed of the manipulator. The updated target tracking position,  $\vec{T}_{t+1}$ , is computed, taking into account its current position  $\vec{T}_t$ . Figure 8 shows the described mixed reality HRI system, at simulation run-time, for moving the manipulator and grabbing objects, using the hand tracking system.

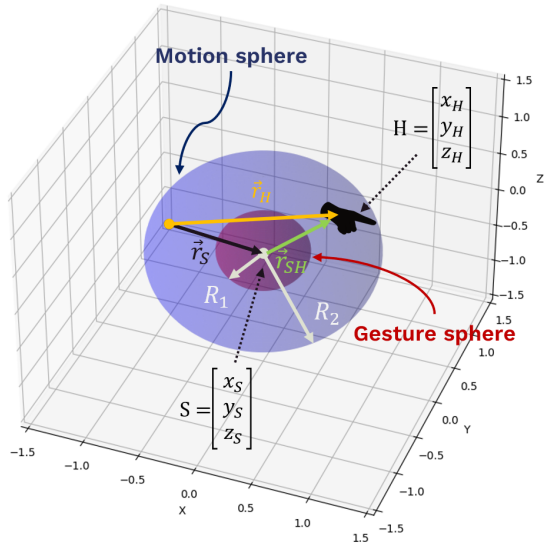


Figure 7: Diagram of the Control Sphere structure used to control the robot arm.

$$\vec{r}_{SH} = \vec{r}_H - \vec{r}_S = \begin{bmatrix} x_H - x_S \\ y_H - y_S \\ z_H - z_S \end{bmatrix} \quad (1)$$

$$r_{SH}^{\hat{}} = \frac{\vec{r}_{SH}}{|\vec{r}_{SH}|} = \frac{\langle x_H - x_S, y_H - y_S, z_H - z_S \rangle}{\sqrt{(x_H - x_S)^2 + (y_H - y_S)^2 + (z_H - z_S)^2}} \quad (2)$$

$$\vec{T}_{t+1} = \vec{T}_t + C\Delta t \times r_{SH}^{\hat{}} = \begin{bmatrix} x_t + C\Delta t \times r_{SH_x}^{\hat{}} \\ y_t + C\Delta t \times r_{SH_y}^{\hat{}} \\ z_t + C\Delta t \times r_{SH_z}^{\hat{}} \end{bmatrix} \quad (3)$$

## Experiment Setup

An experiment was devised to compare the implemented MR HRI system to the traditional monocular fixed-view keyboard teleoperation method. The goal of the experiment was to observe trends relating to depth awareness, overall situational awareness, and task completion efficiency. The experiment consisted of driving the robot arm,

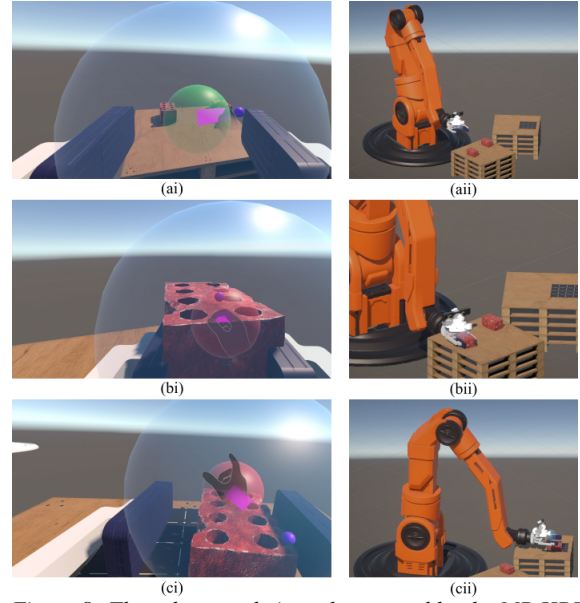


Figure 8: The robot arm being teleoperated by the MR HRI system using hand tracking for motion control and gesture recognition to control the gripper. Left: hand tracking input, right: robot's response.

from its starting position, to a brick, that is placed on a pallet, with the goal of moving to the position where the brick would be in between the gripper, as shown in Figure 9. To collect data during run-time, C# scripts were written to store the coordinates of the end effector, along with the elapsed time, between iterations. Each teleoperation method was executed by the same operator 5 times. The raw flight path data will be indicative of depth perception and situational awareness of the operator while the time to minimize the distance to the target will be indicative of efficiency.

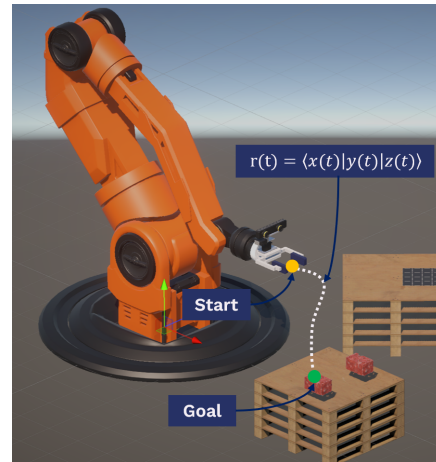


Figure 9: Experiment Setup

## Results

The trajectories of the robot arm's end effector for each of the 5 runs of the monocular fixed-view keyboard teleoperation is shown in Figure 11a). The trajectories of the robot arm's end effector for each of the 5 runs of the MR

HRI teleoperation is shown in Figure 11b). A side-by-side comparison of both teleoperation modes, for each run, is shown in Figure 11c). Inspection of the trajectories of the monocular fixed-view keyboard teleoperation method demonstrates irregular flight paths on each of the runs. The flight path trends show two distinct patterns: simultaneously moving forward and laterally towards the target followed by an immediate singular lateral movement. The inability to maintain a smooth trajectory indicates a lack of depth perception of the robot’s workspace. Inspection of the trajectories of the MR HRI teleoperation method demonstrates consistent and smooth flight paths on each of the runs. Each of the trajectories has the properties of a direct vector, from the starting pose to the goal pose. This suggests that there is a high level depth-awareness and situational awareness of the robot’s workspace. The side-by-side view of both teleoperation methods demonstrates this difference in operator awareness.

The time to complete the task for each teleoperation method and runs are shown in Figure 10a). The number of iterations in the simulation each teleoperation method took to complete the task for each of the runs is shown in Figure 10b). Upon initial inspection, both teleoperation methods are observed to complete the task, per run, approximately around the same duration of time. The results show the traditional monocular fixed-view keyboard teleoperation method completes the task quicker, in some runs, than the MR HRI method. This is contradictory to the trajectory results as the MR HRI method moves in a direct path to the goal pose while the traditional method has an indirect path. Furthermore, for both teleoperation methods and all simulation runs, the robot arm was set to move at the same constant velocity. Inspection of the number of iterations for each of the runs shows that the MR HRI system completes the task in approximately 2.5x fewer iterations than the traditional teleoperation method. Therefore, the longer task completion duration for the MR HRI method is hypothesized to be due to its higher computational demand compared to the simplistic keyboard teleoperation method. The MR HRI simulation takes longer to execute a single iteration, or a lower frame rate, resulting in a longer simulation run-time. However, it is evident that the MR HRI system is able to minimize the same distance to the target as the keyboard teleoperation method with fewer iterations. To truly understand the relationship between task completion time and the two teleoperation methods, a real-world implementation is required.

## Conclusion and Future Work

We proposed and implemented a mixed reality human robot interaction system for a 4-DoF robot arm. The results of the our architecture demonstrate a higher operator depth perception and situational awareness of the robot’s workspace compared with a traditional monocular fixed-view keyboard teleoperation method. Supported by stereo vision systems, a head mounted display, and hand tracking algorithms, the operator is able to manipulate the robot

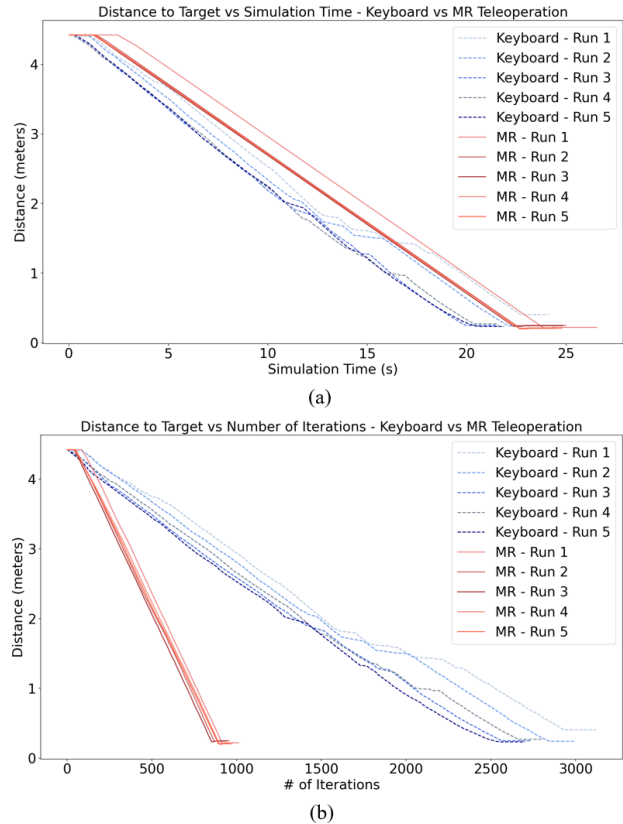


Figure 10: (a) Distance to target vs simulation run-time data and (b) Distance to target vs number of simulation iterations to complete task data.

arm’s end effector safely and efficiently to perform simple construction tasks, such as picking up bricks in a simulation environment. The modularity and mechanics of the MR HRI system will enable the control of various real-world robot arms, rather than a select few. The system alleviates the need for on-site construction workers, greatly reducing accidents and serious injuries in construction sites. Rather, the system enables operators to utilize their knowledge and expertise from a safe distance, performing intricate tasks while simply wearing an HMD. Future work will require the extension of the MR HRI system to a physical robot arm. At the time of writing, a successful implementation of real-time hand tracking with the real-time stereo vision of the physical ZEDM camera has been completed. Next steps will include the development of a two-axis gimbal, to mount the vision system, to demonstrate HMD tracking. Additionally, control algorithms to interface between a physical 4-DoF robot arm and the MR HRI system will be developed. A test with a physical robot arm will provide important data on task completion efficiency, operator training, and robot control accuracy.

## Acknowledgments

This research is funded by the Natural Sciences and Engineering Research Council of Canada (NSERC DG 485794).

