Teleoperation of a Franka Emika Panda Robot Using VR-Based Hand Tracking

Danial Khatoonabadi

Based on prior group project from ELEC 442

University of British Columbia

Term 1, 2025

Abstract

This project extends prior work on human-to-robot motion imitation with the Franka Emika Panda by transitioning from depth camera-based pose estimation to virtual reality (VR) teleoperation. Instead of RGB-D sensing and neural network-based keypoint extraction, the system leverages a Meta Quest Pro headset to directly track the user's wrist pose and pinch gestures within a Unity-based VR environment. These signals are mapped to robot end-effector commands, enabling intuitive task-space control for manipulation tasks such as pick-and-place.

The original objective of achieving real-time execution on the physical Franka robot was constrained by system-level limitations, including the need to reimage the Linux machines interfacing with the robot, which prevented full integration and live deployment. To address this, the project pivoted toward simulation using NVIDIA Isaac Sim, where the complete teleoperation pipeline was implemented and validated. Within this simulated environment, the VR-based control framework demonstrated stable and repeatable pick-and-place motions, preserving the core human-in-the-loop interaction paradigm.

This work highlights both the advantages and challenges of VR-based robot teleoperation, including improved intuitiveness and reduced reliance on external perception, as well as practical issues related to system integration and real-time deployment on physical hardware. By bridging immersive VR interfaces with physics-accurate robotic simulation, the project establishes a scalable foundation for future work involving humanoid robotics, advanced manipulation, and applications in teleoperation and human–robot interaction.

Contents

1 Introduction

The rapid advancement of robotics and artificial intelligence has expanded the range of real-world robotic applications and increased the importance of effective Human–Robot Interaction (HRI). As robots become more physically embodied and interactive, intuitive and low-latency control interfaces are essential for enabling direct human–robot collaboration.

A key challenge in HRI is human-guided robot control, where robots are commanded through demonstrations or real-time interaction rather than explicit programming. Teleoperation and imitation-based control are particularly well suited for manipulation tasks, as they allow human intent to be expressed directly. Traditional approaches often rely on external perception pipelines, such as RGB-D cameras and pose estimation networks, to infer human motion. While functional, these methods introduce latency, sensing noise, and system complexity.

Recent work has shifted toward Virtual Reality (VR)-based teleoperation, which captures human intent directly through tracked wrist motion and hand gestures using commodity VR hardware. VR interfaces eliminate the need for external perception, provide immersive operator feedback, and enable precise, continuous control of robotic manipulators. As a result, VR-based teleoperation has become an effective tool for intuitive manipulation, demonstration collection, and learning-from-demonstration pipelines.

- BEAVR is an open-source, bimanual, multi-embodiment VR teleoperation framework that unifies real-time robot control, data collection, and policy learning across heterogeneous robotic platforms [2]. It supports systems ranging from 7-DoF manipulators to full humanoids and achieves end-to-end latencies below 35 ms through a low-latency, zero-copy streaming architecture. Its modular design, asynchronous control structure, and compatibility with modern visuomotor learning policies highlight the convergence of VR teleoperation, scalable system design, and robot learning.

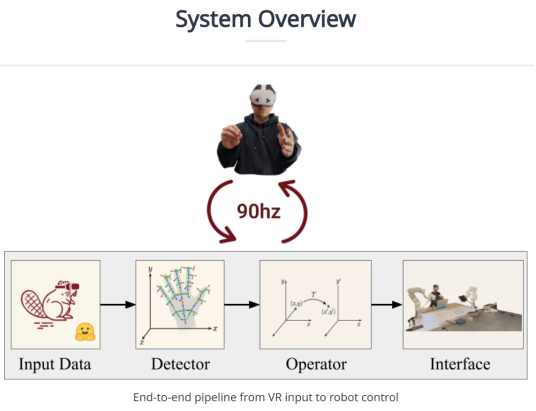

Figure 1: System Overview of BEAVR research

- Another research from the Collaborative Robotics Lab at Virginia Tech focuses on learning and control algorithms that enable robots to collaborate adaptively with humans [3]. This work emphasizes personalized robot behavior and interfaces that allow

robots to both interpret and respond to human intent. VR-based teleoperation aligns naturally with these objectives by providing a shared control space that tightly couples human motion, decision-making, and robotic execution.

Figure 2: Virtual Reality, Interfaces and Active Haptic Feedback Research at Virginia Tech Collaborative Robotics Lab

Many other universities and research labs are actively exploring VR-based teleoperation for robotics, including MIT, Stanford, CMU, and others [4]. These efforts span a wide range of applications, from industrial manipulation to assistive robotics and telepresence. The common thread is the recognition that immersive VR interfaces can significantly enhance the intuitiveness, responsiveness, and effectiveness of human–robot collaboration. Some notable research are included in the references section.

2 Project Objectives

This project investigates immersive Virtual Reality (VR)-based teleoperation as an intuitive interface for human–robot interaction using the Franka Emika Panda robotic arm. Instead of inferring human motion through external depth-camera-based pose estimation, the system directly captures the operator's wrist pose and hand gestures using a Meta Quest Pro headset within a Unity environment, enabling direct mapping between human motion and robotic action. VR-tracked wrist motion and pinch gestures are interpreted as end-effector pose commands for manipulation tasks such as pick-and-place.

The Franka Emika Panda is a 7-DoF torque-controlled manipulator with human-like kinematics, making it well suited for task-space control and dexterous manipulation. Since end-effector tracking is a task-space objective and the robot is kinematically redundant, the control framework exploits null-space methods to resolve redundancy. The primary controller drives the end effector to follow wrist-derived target poses, while secondary objectives, such as posture regulation and joint-limit avoidance, are projected into the null space.

(a) Initial Pose (b) Grasp Phase (c) Placement Phase

Figure 3: Pick-and-place sequence executed by the Franka Emika Panda.

Due to system-level constraints that prevented real-time execution on physical hardware, the teleoperation pipeline was implemented and validated in NVIDIA Isaac Sim, a physics-accurate robotic simulation environment. The system demonstrated stable and repeatable pick-and-place behaviors under VR control, allowing evaluation of task-space tracking, null-space posture regulation, and overall system responsiveness without hardware risk.

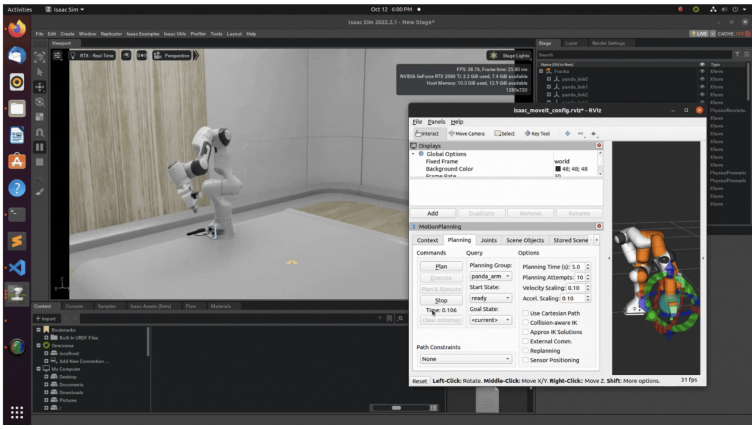

Figure 4: NVIDIA Isaac Sim Franka Emika Panda UI

This setup provides a practical platform to explore key challenges in VR-based teleoperation, including coordinate frame alignment between VR space and robot space, latency and jitter management, task-space trajectory smoothing, and redundancy resolution via null-space control. By implementing an end-to-end pipeline, from immersive human input to simulated robotic execution, this project contributes a hands-on demonstration of modern HRI techniques emphasizing intuitive, gesture-driven control while preserving robot safety and motion quality.

There are two significant things about this project. First, it provides practical experience in VR teleoperation, simulation-driven development, and redundancy-aware control, an increasingly important direction in contemporary robotics research. Second, it offers hands-on exposure to manipulation using the Franka Emika platform, including control design and simulation-to-real considerations. While the demonstrated tasks are intentionally simple, the resulting pipeline establishes a strong foundation for future extensions involving real-time hardware deployment, constraint-aware motion generation, and learning-from-demonstration workflows.

3 Methodology

3.1 System Overview

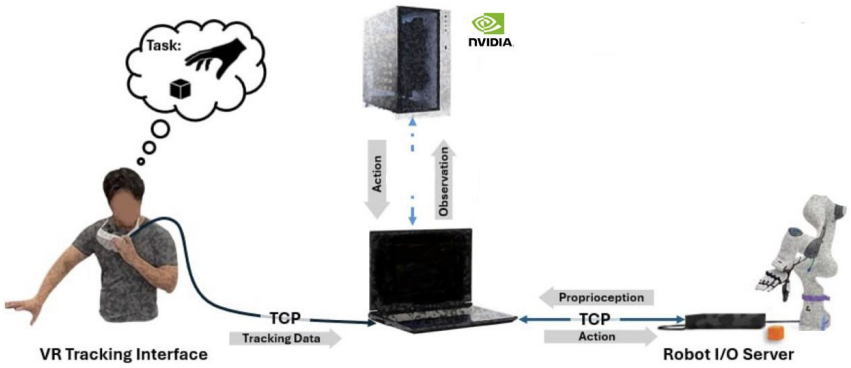

Figure 5: Simple System Diagram of Hardware and Communication

As shown in the diagram above, the proposed system enables immersive VR-based teleoperation of a robotic manipulator by mapping human wrist motion to robot end-effector control. A human operator wearing a Meta Quest Pro headset interacts with a Unity-based virtual environment, where wrist position and hand gestures are tracked in real time. These signals provide an intuitive representation of human intent and are used to guide manipulation tasks such as pick-and-place, while avoiding the need for external perception hardware.

Wrist pose data captured in the VR coordinate frame is transformed into the robot base frame through TCP and interpreted as relative end-effector motion commands. This relative control strategy improves stability and allows fine manipulation without requiring precise alignment between human and robot workspaces. Pinch gestures are mapped to gripper open and close commands, enabling intuitive execution of grasping actions.

Robot motion is controlled in task space, with the primary objective of tracking the desired end-effector pose derived from VR input. Since the Franka Emika Panda is a 7-DoF manipulator, the controller exploits kinematic redundancy through null-space methods. While the task-space controller ensures accurate end-effector tracking, secondary objectives such as posture regulation and joint-limit avoidance are enforced in the null space, improving motion smoothness and predictability during teleoperation.

3.2 VR Setup and Unity Interface

The VR teleoperation framework used in this project adopts a modular design that separates general VR input handling from robot-specific control logic [14]. This structure allows the VR interface to remain independent of the underlying robotic platform, while kinematic mapping, control constraints, and robot-specific adaptations are handled downstream in the control stack. Such separation simplifies system integration and supports future extensions to additional robotic embodiments with minimal modification.

Human input is captured using the Meta Quest Pro's built-in hand tracking through the OpenXR Hand API. Rather than leveraging the full hand skeleton for dexterous manipulation, the system focuses on tracking the operator's wrist pose, which provides a stable and intuitive proxy for robot end-effector motion. Unity processes the tracked wrist position and orientation at approximately 30 Hz and outputs a wrist pose represented by a 3D position and quaternion orientation, which serves as the primary control signal for pick-and-place manipulation tasks.

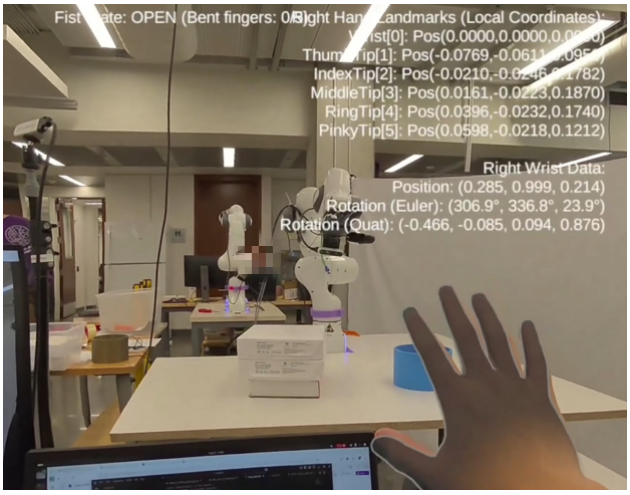

Figure 6: VR Hand-tracking Interface: shadow overlay showing the tracked operator's hand in real time.

Within the Unity environment, the tracked wrist pose is visualized through a real-time overlay that mirrors the operator's motion. This visual feedback enables verification of tracking quality, alignment, and motion scaling between the human and virtual workspaces, helping maintain operator confidence and consistent teleoperation performance.

To ensure reliable and low-latency communication, wrist pose data is streamed to the robot control pipeline using a wired TCP connection. This approach avoids the jitter and latency fluctuations associated with wireless communication and maintains temporal consistency between VR input, control updates, and simulation rendering. At the start of teleoperation, the initial wrist pose and robot end-effector pose are recorded, and subsequent motion is interpreted using a relative control strategy. Incremental wrist displacements are mapped into the robot base frame via a calibrated transformation between the VR and robot coordinate frames, improving robustness and preventing large discontinuities that can arise from absolute pose mapping.

Figure 7: VR hand tracking signal processing the arm branch and hand branch

The resulting target end-effector pose is passed to the robot control layer to generate task-space commands. In this project, manipulation focuses on end-effector control rather than full hand or finger retargeting. Grasping actions are triggered through simple gesture detection, with pinch gestures mapped to discrete gripper open and close commands. This design choice reduces system complexity while remaining sufficient for pick-and-place manipulation.

Overall, the VR and Unity setup provides an intuitive, low-latency interface for human-in-the-loop robot control. By combining direct wrist tracking with relative motion mapping, the system enables precise end-effector control while preserving simplicity and extensibility. This VR interface forms the front end of the teleoperation pipeline and underpins the task-space and null-space control strategies implemented downstream in simulation.

Figure 8: Meta Quest Pro setup with Frank Emika Panda robot

3.3 Arm Branch – Differential Intent Inverse Kinematics

At the start of teleoperation, the initial operator wrist pose $T_{\text{wrist},0}$ and the initial robot end-effector pose $T_{\text{ee},0}$ are recorded. At each timestep t, a differential motion intent is computed in the operator frame based on the relative motion of the operator's wrist.

Let $\mathbf{p}(T) \in \mathbb{R}^3$ denote the Cartesian position of a pose T, and let $\mathbf{q}(T) \in \mathbb{S}^3$ denote its unit-quaternion orientation. Quaternion multiplication is denoted by $\otimes$. The differential position and orientation of the operator's wrist are computed as*

$$\Delta\mathbf{p}_{\text{op}} = \mathbf{p}(T_{\text{wrist},t}) - \mathbf{p}(T_{\text{wrist},0}), \tag{1}$$

$$\Delta\mathbf{q}_{\text{op}} = \mathbf{q}_{\text{wrist},t} \otimes \mathbf{q}_{\text{wrist},0}^{-1}. \tag{2}$$

3.3.1 Mapping to the Robot Base Frame

Let $T_{\text{op}}^{\text{base}} \in SE(3)$ (Special Euclidean Group in 3 dimensions) be a calibrated rigid-body transform from the operator frame to the robot base frame. The target end-effector pose is then computed by applying the differential operator motion to the robot's initial end-effector pose:

$$T_{\text{ee}}^{\text{target}} = \left(T_{\text{op}}^{\text{base}}\ [\Delta\mathbf{p}_{\text{op}}, \Delta\mathbf{q}_{\text{op}}]\ \left(T_{\text{op}}^{\text{base}}\right)^{-1}\right) T_{\text{ee},0}. \tag{3}$$

Here, $[\Delta\mathbf{p}, \Delta\mathbf{q}]$ denotes an affine transformation composed of a rotation $R(\Delta\mathbf{q})$ and a translation $\Delta\mathbf{p}$.

3.3.2 Inverse Kinematics

The desired end-effector pose is converted into joint-space targets using an inverse kinematics (IK) solver, which may be analytical or numerical depending on the implementation:

$$\mathbf{q}_t^{\text{arm}} = \text{IK}\left(T_{\text{ee}}^{\text{target}}, \text{URDF}_{\text{arm}}\right), \tag{4}$$

subject to joint limits and kinematic constraints defined by the robot Unified Robot Description Format (URDF).

To improve stability during teleoperation, end-effector orientation tracking may be disabled or down-weighted. In such cases, the orientation increment is approximated as

$$\Delta\mathbf{q}_{\text{op}} \approx \mathbf{1}, \tag{5}$$

or incorporated as a weighted objective within the IK formulation. (Appendix A provides more details on the IK formulation and redundancy resolution.)

*Portions of the mathematical formulation and notation in this section are adapted from the LeVR and BEAVR virtual reality teleoperation frameworks.

3.3.3 Gripper Control

In this project, full dexterous hand retargeting is not performed. Instead, grasping actions are triggered through a simple gesture-based mapping, where a pinch gesture commands the gripper to close and its absence commands the gripper to open:

$$u_{\text{gripper}} = \begin{cases} \text{close}, & \text{pinch detected}, \\ \text{open}, & \text{otherwise}. \end{cases} \tag{6}$$

More advanced dexterous hand retargeting is left as future work.

3.3.4 Redundancy Resolution via Null-Space Projection

Since the Franka Emika Panda is a redundant manipulator, joint-level commands are generated using null-space projection to allow secondary objectives without affecting end-effector tracking. The commanded joint velocity is given by

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}}_{\text{ee}} + \left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\dot{\mathbf{q}}_{\text{null}}, \tag{7}$$

where $\mathbf{J}$ is the manipulator Jacobian, $\mathbf{J}^{\dagger}$ its Moore–Penrose pseudoinverse, and $\dot{\mathbf{q}}_{\text{null}}$ encodes secondary objectives such as posture regulation and joint-limit avoidance. This formulation allows the robot to exploit kinematic redundancy while preserving accurate end-effector tracking. (Appendix B provides more details)

3.4 NVIDIA Isaac Sim Integration

To validate the teleoperation and pick-and-place framework without requiring continuous access to the physical Franka Emika Panda, a full digital twin was developed using NVIDIA Isaac Sim within the NVIDIA Omniverse ecosystem [12]. Isaac Sim provides a high-fidelity physics-based robotics simulation environment built on USD (Universal Scene Description), enabling accurate rigid-body dynamics, collision modeling, and sensor simulation.

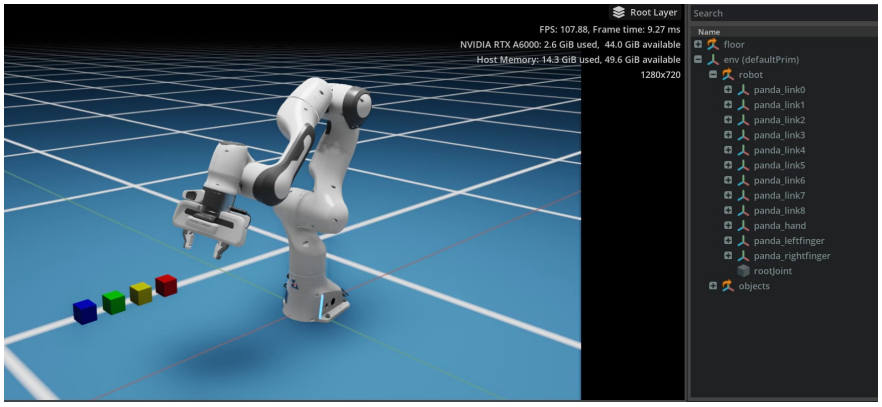

Figure 9: Isaac Sim Franka Emika Panda Simulation

The teleoperation framework is implemented and validated in NVIDIA Isaac Sim, which provides a physics-accurate simulation of the Franka Emika Panda. The simulation environment enables safe evaluation of VR-based pick-and-place tasks, including task-space tracking performance, null-space behavior, and overall system responsiveness.

3.5 Simulation Setup

The simulation environment was configured to replicate the physical laboratory setup using the Franka Panda model available in NVIDIA Isaac Sim. The articulated 7-DOF manipulator, including joint limits, collision meshes, and rigid-body dynamics, was instantiated within the USD stage. NVIDIA PhysX was configured with a fixed timestep, gravity, and tuned solver parameters to ensure stable contact behavior during grasping. The virtual workspace consisted of a planar table, a cubic object, and a defined placement region to reproduce the physical pick-and-place task.

Robot control was implemented using Isaac Sim's articulation controller, supporting joint position, velocity, and torque-level commands. This enabled compatibility with the torque-based control framework used on hardware, ensuring that both kinematic and dynamic behaviors in simulation closely matched the real Franka system.

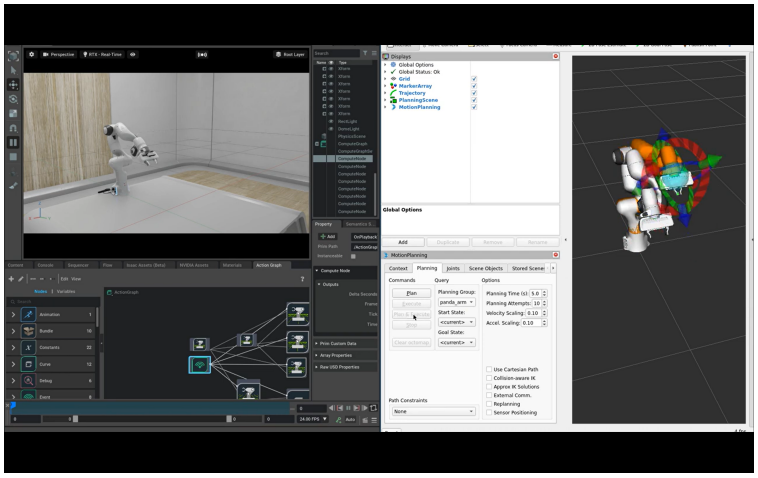

Figure 10: Franka Emika Panda Simulation in Isaac Sim

3.6 ROS 2 and MoveIt 2 Integration

The system was deployed on Ubuntu 22.04 using ROS 2 (Humble) and MoveIt 2 as the motion planning framework [13]. Isaac Sim's ROS 2 bridge was integrated into the same ROS computation graph used for hardware experiments, allowing the simulator to publish standard topics such as `/joint_states`, TF transforms, and robot state feedback. The MoveIt 2 configuration package for the Franka robot was generated from the URDF and SRDF, enabling kinematic consistency, collision checking, and planning scene management within the ROS 2 environment.

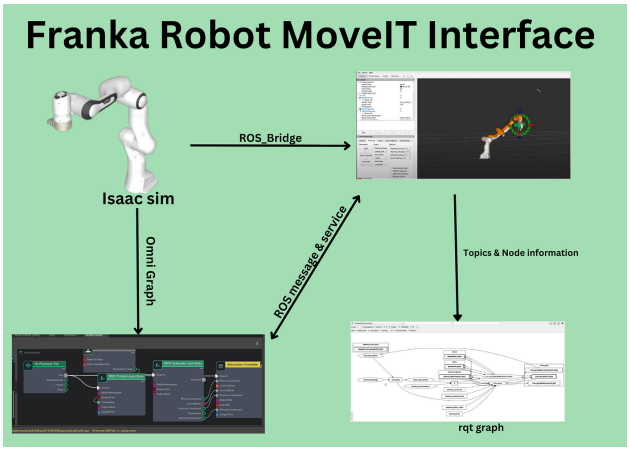

Figure 11: Franka Robot MoveIt Interface

The previously developed torque-level controller (`jointMatchingController`) was reused without modification. The controller interfaces with the Franka state and model interfaces, incorporates Ruckig for trajectory generation, and enforces joint limits through virtual joint walls. In simulation, Isaac Sim provided equivalent joint state and dynamic feedback, allowing seamless execution of the same control pipeline. Teleoperation commands were published to the `/joint_states` topic as in hardware operation. This hardware-agnostic architecture ensured that only the execution backend differed (physical robot vs. simulated articulation), while all planning, kinematics, and control layers remained identical across environments.

3.7 Pick-and-Place Motion Validation

The complete pick-and-place sequence was validated in simulation. During the approach phase, Ruckig generated smooth joint trajectories with bounded velocity, acceleration, and jerk. The grasp phase verified stable contact and collision handling under PhysX dynamics. During object transport and placement, joint-limit enforcement, saturation behavior, and torque-level stability were evaluated.

The high-fidelity physics engine enabled assessment of motion smoothness, contact stability, and controller robustness under realistic interaction forces. Workspace boundary behavior and virtual joint wall effectiveness were also validated prior to hardware execution.

Figure 12: Pick-and-place sequence executed in Isaac Sim

3.8 Digital Twin Implementation

Within this framework, Isaac Sim effectively served as a digital twin of the Franka Emika Panda. The complete control stack—including joint-level torque control, saturation logic, Ruckig trajectory generation, and virtual joint wall enforcement—remained identical between simulation and physical deployment.

This one-to-one correspondence ensured a seamless transition from simulation to hardware. Commissioning time was reduced, and the risk associated with first-time real-world execution was significantly lowered. The digital twin approach provided high confidence in controller performance prior to physical experimentation.

4 Results

The proposed VR-based task-space teleoperation framework was successfully validated in simulation using NVIDIA Isaac Sim integrated with Ubuntu 22.04, ROS 2 (Humble), and MoveIt 2. By mapping wrist pose increments directly to end-effector motion, the system avoids the instability and ambiguity associated with full-arm vision-based pose imitation. The teleoperation loop operated in real time in simulation, producing smooth and intuitive robot motion.

Ruckig-based trajectory generation ensured bounded velocity, acceleration, and jerk profiles, eliminating discontinuities during rapid operator movements. Null-space redundancy resolution maintained safe joint configurations and prevented drift toward joint limits during extended operation. The separation of task-space tracking from posture regulation resulted in predictable and stable behavior across repeated trials.

The complete pick-and-place sequence was executed consistently, including approach, grasp, transport, and placement phases. Contact interactions remained stable under the PhysX physics engine, and torque-level tracking showed no instability. Because the same ROS 2 computation graph, MoveIt 2 configuration, and joint-level controller were used in both simulation and hardware setups, the system maintains architectural consistency and reduced deployment risk.

Overall, the results demonstrate that VR-based task-space teleoperation with null-space redundancy resolution provides a robust and intuitive solution for redundant manipulator control in structured manipulation tasks.

5 Discussion and Conclusion

Virtual reality (VR)–based human–robot interaction (HRI) is an increasingly important paradigm in modern robotics, particularly for humanoid systems and advanced manufacturing environments. As automation shifts toward human-centered collaboration under Industry 4.0 and Industry 5.0 frameworks, immersive interfaces are essential for bridging human intent and robotic execution. VR-driven teleoperation enables natural motion capture, remote manipulation, and safe supervision in complex or hazardous environments, making it highly relevant for both adaptable humanoid platforms and flexible manufacturing systems.

In industrial robotics, pick-and-place manipulation remains a canonical benchmark due to its central role in assembly, logistics, and material handling. Humanoid robots similarly use pick-and-place tasks to evaluate dexterity and shared autonomy. VR-based teleoperation offers advantages in remote supervision, rapid task prototyping, and human-in-the-loop training. By enabling intuitive joint-level control with dynamic safety constraints, the framework presented in this work contributes toward scalable manipulation strategies applicable to industrial and humanoid contexts.

This work developed a real-time VR-based teleoperation framework for a 7-DOF manipulator using torque-level joint control. The system integrates jerk-limited trajectory generation via Ruckig, proportional–derivative control with Coriolis compensation, drift correction, and safety-aware velocity, acceleration, and jerk limits. Virtual joint walls derived from URDF soft limits further enhance safety. Together, these components enable smooth, dynamically feasible tracking of human-inspired commands while mitigating oscillations and abrupt motion, demonstrating that immersive human input can be effectively coupled with advanced control strategies for safe teleoperation.

Beyond industrial applications, VR-based teleoperation also holds promise in medical and rehabilitation robotics [15]. Stroke and neurodegenerative diseases often impair upper-limb motor control, while traditional physiotherapy can be repetitive and resource-intensive. Robotic rehabilitation systems—such as end-effector upper-limb devices providing assist-as-needed support through Cartesian impedance control—show that adaptive interaction can enhance engagement and neuroplastic recovery. Virtual Reality Mirror Therapy (VRMT), which visually mirrors movements of a healthy limb to stimulate cortical reorganization in the impaired side [17], further demonstrates the potential of immersive rehabilitation. Integrating VR teleoperation, 3D pose estimation, and adaptive robotic assistance could enable intelligent rehabilitation assistants that combine mirror therapy principles with safe physical guidance. A 7-DOF manipulator such as the Franka Emika Panda, coupled with vision-based tracking and AI-driven adaptation, may provide a scalable and personalized platform for three-dimensional therapeutic training.

However, the current framework remains limited to partial joint-level mapping and lacks full task-space control, which may reduce intuitiveness compared to end-effector teleoperation. The system operates in an open-loop unilateral configuration without haptic feedback, and controller gains remain fixed without adaptive impedance or learning-based intent prediction. Future work may extend this approach toward task-space control, redundancy resolution, bilateral teleoperation, adaptive impedance strategies, and AI-guided rehabilitation applications. Overall, this project establishes a robust foundation for VR-driven HRI and contributes toward scalable control architectures for humanoid robotics, intelligent manufacturing, and assistive medical technologies.

6 Acknowledgement and References

I would like to sincerely thank the *Robotics and Controls Laboratory (RCL)* at the *University of British Columbia (UBC)* for providing the opportunity to gain hands-on research experience in advanced robotic control and human–robot interaction.

I acknowledge the use of generative AI tools for limited editorial assistance in improv-

ing clarity and formatting. All research design, implementation, analysis, and technical validation were conducted independently, and all technical content is grounded in prior research and relevant coursework with appropriate citations provided. The author retains full responsibility for the integrity of this work.

References

[1] D. Khatoonabadi, "Franka VR Lab: VR-based teleoperation framework for the Franka Emika Panda," GitHub repository. [Online]. Available: https://github.com/Dani1382/franka_vr_lab. Accessed: Feb. 2026.

[2] MIT Agile Robotics and Control Laboratory (ARC Lab), "BEAVR: Bimanual Enhanced Autonomous Virtual Reality Teleoperation Framework." [Online]. Available: https://arclab-mit.github.io/beavr-landing/. Accessed: Sep. 2025.

[3] Collaborative Robotics Laboratory (CoRL), Virginia Tech, "Collaborative Robotics Laboratory." [Online]. Available: https://collab.me.vt.edu/. Accessed: Sep. 2025.

[4] Z. K. Weng, M. L. Elwin, and H. Liu, "LeVR: A Modular VR Teleoperation Framework for Imitation Learning in Dexterous Manipulation," *arXiv preprint* arXiv:2509.14349v1, Sep. 2025. [Online]. Available: https://arxiv.org/html/2509.14349v1.

[5] A. George, A. Bartsch, and A. B. Farimani, "OpenVR: Teleoperation for Manipulation," *arXiv preprint* arXiv:2305.09765, May 2023. [Online]. Available: https://arxiv.org/abs/2305.09765. Accessed: Feb. 2026.

[6] K. Wan, C. Li, F.-S. Lo, and P. Zheng, "A virtual reality-based immersive teleoperation system for remote human-robot collaborative manufacturing," *Manufacturing Letters*, vol. 41, pp. 43–50, Oct. 2024, doi: 10.1016/j.mfglet.2024.09.008.

[7] Y. Li, H. Zhang, J. Sun, and X. Chen, "Virtual Reality–Enabled Human–Robot Interaction for Intelligent Manufacturing Systems," *Cyber-Physical Systems*, vol. 3, no. 2, Art. no. 0098, 2024, doi: 10.34133/cbsystems.0098.

[8] K. Joseph, "Gaze-Enabled Grasping Assistance for Teleoperation of Robotic Manipulators," M.A.Sc. thesis, Dept. Mech. Mechatronics Eng., Univ. Waterloo, Waterloo, ON, Canada, 2025. [Online]. Available: https://uwspace.uwaterloo.ca/bitstreams/39a89260-4444-41cb-aa1c-3ffb69399819/download. Accessed: Jan. 2026.

[9] NVIDIA Corporation, "ROS 2 MoveIt Tutorial — Isaac Sim 4.2.0 Documentation." [Online]. Available: https://docs.isaacsim.omniverse.nvidia.com/4.2.0/ros2_tutorials/tutorial_ros2_moveit.html. Accessed: Oct. 2025.

[10] PickNik Robotics, "Pick and Place with MoveIt Task Constructor — MoveIt Documentation (ROS 2 Humble)." [Online]. Available: https://moveit.picknik.ai/humble/doc/tutorials/pick_and_place_with_moveit_task_constructor/pick_and_place_with_moveit_task_constructor.html. Accessed: Nov. 2025.

[11] NVIDIA Corporation, "Create realistic robotics simulations with ROS 2, MoveIt, and NVIDIA Isaac Sim," *NVIDIA Developer Blog*, Aug. 2023. [Online]. Available: https://developer.nvidia.com/blog/create-realistic-robotics-simulations-with-ros-2-moveit-and-nvidia-isaac-sim/. Accessed: Nov. 2025.

[12] NVIDIA Corporation, "NVIDIA Omniverse Developer Platform." [Online]. Available: https://developer.nvidia.com/omniverse. Accessed: Dec. 2025.

[13] Open Robotics, "ROS 2 Humble Documentation." [Online]. Available: https://docs.ros.org/en/humble/index.html. Accessed: Oct. 2025.

[14] Meta Platforms, Inc., "Meta Horizon Developer Documentation — Unity Development." [Online]. Available: https://developers.meta.com/horizon/develop/unity/. Accessed: Dec. 2025.

[15] Z. Zhang, Y. Liu, Y. Yang, and X. Wang, "Design and control of an end-effector upper limb rehabilitation robot based on assist-as-needed strategy," *Applied Sciences*, vol. 10, no. 19, Art. no. 6684, 2020, doi: 10.3390/app10196684.

[16] Healthcare in Europe, "Meet Robert: Your robotic physiotherapist." [Online]. Available: https://healthcare-in-europe.com/en/news/meet-robert-your-robotic-physio-therapist.html. Accessed: Feb. 2026.

[17] H. Zhao, L. Chen, Y. Wang, and X. Li, "Virtual reality-based mirror therapy system for upper limb rehabilitation after stroke," *Journal of Healthcare Engineering*, vol. 2024, Art. no. 2001037024000175, 2024, doi: 10.1016/j.jhge.2024.2001037024000175.

[18] S. Thieme, N. Mehrholz, J. Pohl, J. Behrens, and M. Dohle, "Mirror therapy for improving motor function after stroke," *Cochrane Database of Systematic Reviews*, no. 3, Art. no. CD008449, 2018, doi: 10.1002/14651858.CD008449.pub3.

[19] Meta Platforms, Inc., "VRMT — Virtual Reality Mirror Therapy." [Online]. Available: https://www.meta.com/experiences/vrmt/5954556421337276/. Accessed: Oct. 2025.

7 Appendix

A Inverse Kinematics Formulation

A.1 Notation and Initialization

We denote the VR/world frame by $\{\mathcal{V}\}$ and the robot base frame by $\{\mathcal{B}\}$. End-effector poses are represented by homogeneous transformations $T \in SE(3)$ with position $\mathbf{p} \in \mathbb{R}^3$ and orientation represented either by a quaternion $\mathbf{q} \in \mathbb{H}$ or a rotation matrix $\mathbf{R} \in SO(3)$.

At teleoperation start, we record the operator wrist pose $T_{\text{wrist},0}$ expressed in $\{\mathcal{V}\}$ and the robot end-effector pose $T_{\text{ee},0}$ expressed in $\{\mathcal{B}\}$. Because the Franka arm's workspace is similar to a human arm, unit scaling ($1\times$) is applied to Cartesian increments.

A.2 Target Pose Computation

At time step i, the wrist position increment in the VR frame is

$$\mathbf{p}'_{\mathcal{V}} = \mathbf{p}_{\mathcal{V},i} - \mathbf{p}_{\mathcal{V},0}. \tag{8}$$

The mapping from VR to robot coordinates is given by

$$\mathbf{T}_{\mathcal{V}\to\mathcal{B}} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{9}$$

Thus,

$$\mathbf{p}'_{\mathcal{B}} = \mathbf{T}_{\mathcal{V}\to\mathcal{B}}\mathbf{p}'_{\mathcal{V}}, \qquad \mathbf{p}_{\text{target}} = \mathbf{p}_{\text{ee},0} + \mathbf{p}'_{\mathcal{B}}. \tag{10}$$

For orientation, we compute the relative quaternion in the VR frame:

$$\mathbf{q}_{\delta,\mathcal{V}} = \mathbf{q}_{\mathcal{V},i} \otimes \mathbf{q}_{\mathcal{V},0}^{-1}, \tag{11}$$

where $\otimes$ denotes quaternion multiplication.
This is mapped into the robot base frame via

$$\mathbf{R}_{\delta,\mathcal{B}} = \mathbf{T}_{\mathcal{V}\to\mathcal{B}}\mathbf{R}_{\delta,\mathcal{V}}\mathbf{T}_{\mathcal{V}\to\mathcal{B}}^{\top}, \tag{12}$$

and the target quaternion is formed as

$$\mathbf{q}_{\text{target}} = \mathcal{Q}(\mathbf{R}_{\delta,\mathcal{B}}) \otimes \mathbf{q}_{\text{ee},0}. \tag{13}$$

The resulting target pose is

$$T_{\text{target}} = \begin{bmatrix} \mathbf{R}(\mathbf{q}_{\text{target}}) & \mathbf{p}_{\text{target}} \\ \mathbf{0}^{\top} & 1 \end{bmatrix}. \tag{14}$$

A.3 Analytical Inverse Kinematics (GeoFIK)

The inverse kinematics (IK) problem seeks joint angles

$$\mathbf{q} = [q_1, \ldots, q_7]^{\top} \tag{15}$$

such that

$$T_{\text{target}} = f(\mathbf{q}), \tag{16}$$

where $f(\mathbf{q})$ denotes the forward kinematics mapping.

GeoFIK parameterizes redundancy using joint q_7. For a fixed value q_7^*, the remaining 6-DOF system yields up to eight discrete solutions:

$$\mathcal{S}(q_7^*) = \{\mathbf{q}_1, \ldots, \mathbf{q}_k\}, \qquad k \leq 8, \tag{17}$$

computed through geometric decomposition of the shoulder (joints 1–3), elbow (joint 4), and wrist (joints 5–7).

Solutions that violate joint limits or exhibit poor manipulability are discarded. Continuity is enforced by selecting candidates closest to the previous joint configuration $\mathbf{q}_{t-1}$.

A.4 Redundancy Resolution

The redundancy parameter q_7 is optimized over the interval

$$q_7 \in [q_{7,\min}, q_{7,\max}]$$

using Brent's method (tolerance 10^{-6}, maximum 100 iterations), initialized from $q_{7,t-1}$.

The scalar objective function is defined as

$$J(\mathbf{q}) = w_m M(\mathbf{q}) - w_n \left\|\mathbf{W}_n(\mathbf{q} - \mathbf{q}_{\text{neutral}})\right\| - w_c \left\|\mathbf{W}_c(\mathbf{q} - \mathbf{q}_{t-1})\right\|, \tag{18}$$

where the Yoshikawa manipulability measure is

$$M(\mathbf{q}) = \det(\mathbf{J}\mathbf{J}^\top). \tag{19}$$

The scalar weights w_m, w_n, and w_c trade off manipulability, proximity to a neutral posture $\mathbf{q}_{\text{neutral}}$, and continuity with the previous configuration $\mathbf{q}_{t-1}$, respectively. The diagonal matrices $\mathbf{W}_n$ and $\mathbf{W}_c$ normalize joint ranges and allow joint-specific emphasis.

This scalar search over q_7 is equivalent to optimizing within the manipulator's kinematic null space.

B Null-Space Control Formulation

Since the Franka Emika Panda has 7 DOF, the number of joints exceeds the 6 DOF required to control end-effector pose. This means the robot is kinematically redundant.

Redundancy allows multiple joint configurations to achieve the same end-effector motion.

The task-space velocity command is computed as

$$\dot{\mathbf{x}}_d = \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}), \tag{20}$$

with joint velocities obtained via

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}}_d + \left(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}\right) \dot{\mathbf{q}}_0. \tag{21}$$

The null-space term $\dot{\mathbf{q}}_0$ is chosen to regulate posture:

$$\dot{\mathbf{q}}_0 = -\mathbf{K}_n(\mathbf{q} - \mathbf{q}_{\text{nom}}). \tag{22}$$

This formulation allows the robot to maintain a desired posture while ensuring accurate end-effector tracking, improving stability and predictability during teleoperation.