Human-Robot Collaborative Teleoperation for Remote Construction Tasks via Natural Wearable Sensing

Stephania Ivanov
School of Computing
Montclair State University
Montclair, USA
ivanovs1@montclair.edu

Jared Wu
West Essex High School
North Caldwell, USA
jaredyuewu@gmail.com

Jesse Parron
School of Computing
Montclair State University
Montclair, USA
parronj@montclair.edu

Weitian Wang
School of Computing
Montclair State University
Montclair, USA
wangw@montclair.edu

***Abstract*—As humans further explore the boundaries of space, certain tasks have become increasingly difficult for direct human involvement due to factors such as travel distance, task complexity, and risk, all of which should be evaluated before a mission. Human-robot collaboration has emerged as a critical component in overcoming these challenges in space exploration. This study develops a natural wearable sensing-based human-robot collaborative teleoperation framework for space applications, enabling users to remotely control a robot arm using a state-of-the-art Vicon Motion Capture System (MCS). The MSC captures the user's motion and velocity in real time to guide the robot along an optimized trajectory and complete a complex task that may also be encountered in space environments. Experimental results have demonstrated high accuracy and precision of the developed approach, highlighting the potential of advanced human-robot systems in enabling effective and safe space task collaborations.**

***Keywords*—Robotics, human-robot collaboration, wearable sensing, space, real-time control**

I. Introduction

The rapid advancement of robotics and sensing technologies has paved the way for intelligent systems to be utilized in more complex environments [1-3]. As such, these systems are increasingly used across diverse fields, including manufacturing, healthcare, and agriculture [4-7], where they improve operational efficiency, precision, and most importantly, human safety [8]. In manufacturing, for instance, robots have long been used to perform tasks such as heavy lifting and repetitive motion, helping to reduce human exposure to physical risks and improve overall workplace safety [9]. Berthet-Rayne et al. similarly investigated the role of robotics in the field of healthcare, focusing on the application of the i2Snake, a teleoperated snake-like robot, that employs inverse kinematics control methods for minimally invasive surgery [10]. Their study emphasized the need for responsive and intuitive control strategies to ensure accuracy, precision, and safety in high-stakes medical environments.

As the capabilities of robotics continue to evolve, particularly alongside the rapid growth of artificial intelligence, it is becoming increasingly evident that there is a growing trend toward the integration of human cognition and robotic autonomy [11]. Human-robot collaboration (HRC) is especially important in environments that are dangerous and difficult for humans to access. With teleoperation, robotic systems can be controlled by human operators from a safe distance, enabling the performance of complex tasks such as reconnaissance and mission execution without the physical risk. Tang et al. developed such a human-robot collaborative teleoperation system for a semi-autonomous mobile robot architecture (SAMRA) that allows the robot to autonomously execute planned missions with additional human oversight and control [12]. Similarly, Lee et al. analyzed the opportunities of teleoperation in construction, emphasizing the critical role of human-focused designs to facilitate smooth human-robot interaction in dangerous workplaces [13]. Their findings highlight the effectiveness of combining HRC with teleoperation, demonstrating enhanced flexibility in remote robotic operations. Extending beyond terrestrial applications, fluid teleoperation is especially critical in space exploration. Researchers such as Schmaus et al. have demonstrated the importance of remotely operating humanoid robots to maintain and construct infrastructure in inaccessible environments, including space [14]. These hazardous conditions make teleoperation combined with HRC essential for ensuring human safety in extraterrestrial missions as well as operational success.

In this study, we developed a natural wearable sensing-based and intuitive teleoperation system by leveraging a collaborative robot and the Vicon Motion Capture System (MCS) [15]. The integration of these technologies enables real-time control that eliminates the need for specialized robotics expertise, significantly lowering the barrier to robot programming and control. In this system, reflective markers tracked by the MCS capture hand gestures of users and are then computed into real-time commands for the robot via the developed algorithms. The MoveIt Servo controller interface [16] allows for continuous control of the robot's joints, which enables smooth, real-time teleoperation based on the data provided by the MCS server. This allows for hand-to-robot translation, which, in comparison to traditional frameworks, such as the Open Motion Planning Library (OMPL) [17, 18], offers significant performance. OMPL requires a predefined start and end pose to compute complete motion plans, whereas MoveIt Servo supports continuous and real-time control, making it better suited for dynamic teleoperation tasks.

II. Approach

A. Overview of the Approach

In this study, we propose and implement a framework composed of a state-of-the-art MCS, a Franka Emika Panda Arm integrated with the Robotics Operating System (ROS) [19], and a real-time Cartesian controller interface—MoveIt Servo. The control flow of the system works in the following manner and is shown in Fig.1.

979-8-3315-5937-3/25/$31.00 ©2025 IEEE

The Vicon MCS streams positional (x, y, z), angular (x, y, z), and calculates velocity data of rigid body objects to a server it hosts, at a rate of 100Hz. A client has the ability to connect to the server and receive packets of the rigid body objects data. If the connection is successful, the data is streamed to the client. The client additionally establishes a ROS master node and facilitates communications with the Panda Arm. Once the data is decoded, the client processes the velocity vector into a TwistStamped message containing the linear and angular x, y, and z components. TwistStamped is then published to the ROS workspace, where the node running MoveIt Servo subscribes and uses the information to evaluate the pseudo-inverse of the Jacobian matrix based on the current joint positions of the robot and the desired velocity vector to compute an updated velocity vector capable of completing the intended motion. The robot will then update its position based on the user's movements. If the user does not move, then the robot will set its velocity vector to zero and halt movement.

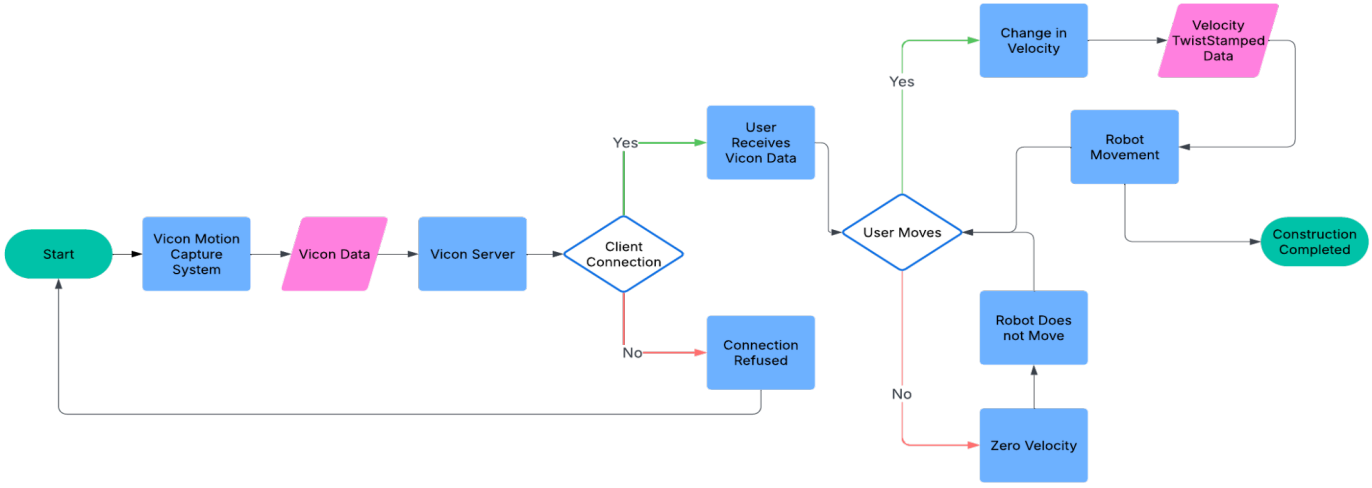

Fig. 1. Control flow diagram of the developed system.

B. *Vicon Motion Capture System*

The Vicon Motion Capture System is a cutting-edge, high-precision tracking platform that provides reliable tracking of movement through optical sensing. The system uses infrared cameras in conjunction with retroreflective markers to capture real-time 3D motion data. In this study, Vicon Vero cameras are utilized to track pearl markers placed on the user's pointer finger, as shown in Fig. 2. The cameras emit infrared light, which reflects off markers placed on an object. This infrared light, while invisible to the human eye, bounces off the markers and then returns directly to the camera's sensors, enabling precise tracking of each marker's position within the tracking space. The Vero cameras feature a resolution of 2.2 megapixels, capturing motion data at 330 frames per second. This capability, combined with a standard tracking error of 0.01 millimeters, enables exceptionally precise tracking of movements and the position of objects in a 3D space [15].

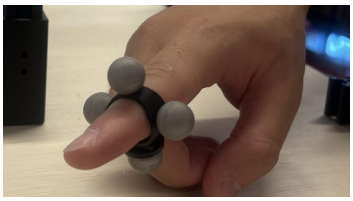

Fig. 2. User with a ring of pearl markers.

The Vicon Tracker software processes the captured motion data from each Vero camera in real-time, precisely reconstructing the 3D positions of the markers. This software additionally manages the calibration of the entire system to maintain spatial accuracy. Wand calibration involves moving a wand equipped with LEDs that emit a red light at known marker positions in circular motions throughout the tracking space in circular motions. As seen in Fig. 3, the Vicon Tracker visualizes each time the calibration wand has been successfully detected and tracked during the calibration process. The higher the "wand count" value, or the times the camera detects the wand, the more accurate the calibration will be. This process ensures that the cameras are synchronized and that the MCS has a precise spatial understanding of the tracking space. Typically, once calibration is complete, the system origin, the point that is the reference for all spatial measurements, should be defined. This creates a fixed coordinate system, ensuring that there is consistency across sessions.

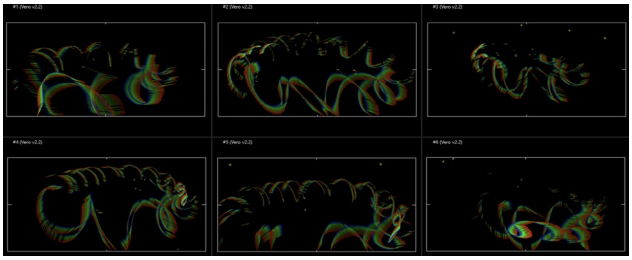

Fig. 3. Vicon Vero camera calibration within Vicon Tracker software.

As seen in Fig. 4, the tracking environment in our lab is equipped with six Vicon Vero cameras, each strategically placed around the perimeter of the space to maximize coverage and minimize marker occlusion. Occlusion, which occurs when a marker is blocked from the cameras' view, making it temporarily untrackable. In the Vicon Tracker, objects are created by assigning three or more markers in a fixed configuration. Once defined, these groupings are recognized by the system, maintain their shape during motion, and are recognized as a single entity.

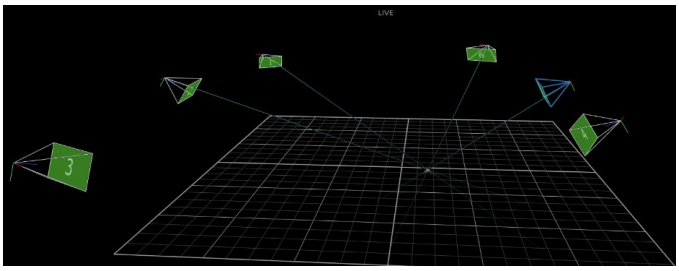

Fig. 4. Camera orientation within tracking space.

C. Data Collection

Data collection throughout the experiment was facilitated through the Vicon Tracker software and API, allowing for real-time access to information captured by the Vero cameras. The tracker captures the 3D positional and angular data (x, y, z) of a rigid body object, which consists of attributes including segments and markers. Each segment corresponds to a named rigid body object and serves as a parent for each marker that defines it. In this experiment, the rigid body object is a ring consisting of four markers on the user's pointer finger, as seen in Fig. 2. The velocity of the rigid body markers can be calculated as:

$$v = \Delta d_{m_i}/1000 * \Delta t, \quad (1)$$

where Δd represents the change in position of each marker m_i for each rigid body object. This translation is divided by the time difference between the two frames to calculate velocity in millimeters per second. The resulting value is further converted to meters per second to match the unit scale used by ROS. Furthermore, the client records the positional data of the rigid body object, velocity vector, orientation, end-effector joint state, end-effector position, and time to a CSV file for further analysis after a task has been completed.

D. Human-Robot System Action Characterization

Using the different types of data collected by the MCS while tracking rigid bodies, users can express different gestures that can be characterized as different actions of the collaborative robot. A user wears a single ring on the pointer finger, composed of four retroreflective pearl markers, capable of being tracked by the Vicon Vero cameras. As a result of the high framerate, precision, and accuracy of the Vicon cameras, the rigid-body object the user is wearing can be tracked anywhere in the Vicon's coordinate plane. A user can convey different robot motions, as presented in Table I, based on the direction and velocity they exhibit while moving around.

TABLE I: HUMAN-ROBOT ACTIONS AND CHARACTERIZATION.

User Actions	Robot Motion	Robot Grippers	End Effector Orientation
Translation ∓X	Forward/Backward	None	0°->90°
Translation ∓Y	Left/Right	None	0°->90°
Translation ∓Z	Up/Down	None	0°->90°
Translation ∓X/∓Y	Forward/Backward & Right/Left	None	0°->90°
Translation ∓X/∓Z	Forward/Backward & Up/Down	None	0°->90°
Translation ∓Y/∓Z	Right/Left & Up/Down	None	0°->90°
Translation ∓X/∓Y/∓Z	Diagonals	None	0°->90°
No Translation	No movement	Open/Close	0°->90°

As indicated in Fig. 5, the user starts with their finger high in the coordinate plane, +Z, and moves in a -Z and -Y direction towards the piece they wish to pick up. Once successfully navigating to the piece, the user then halts all movements, which triggers the grippers to close. The system is incorporated with a toggle to switch between gripper and orientation actions. Depending on the operation and mode, the user may halt all movements to invoke the grippers to open and close or start rotating about the Z-axis. In this case, the user triggered the end-effector grippers to close on the piece.

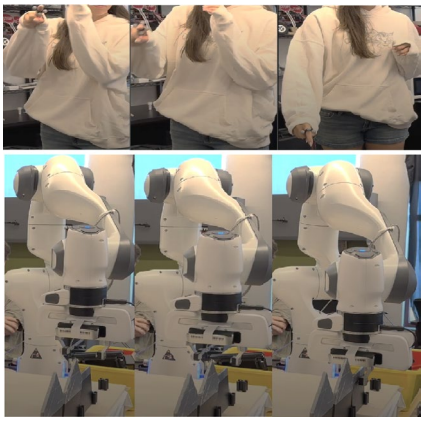

Fig. 5. Human-robot collaborative motion.

E. Robot Motion Planners

Traditionally, robots that are imposed with a motion-based task should employ a trajectory planning algorithm to generate a feasible motion path. The success of the robot's motion execution depends on the effectiveness of the planner, particularly in terms of collision avoidance, path efficiency, and computational cost [18]. Among the many trajectory optimization algorithms developed for robot motion planning, Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [20], Stochastic Trajectory Optimization for Motion Planning (STOMP) [21], and Open Motion Planning Library (OMPL) [17] are extensively used. In this study, we make use of an alternative to the traditional means of motion planning, leveraging MoveIt Servo, a velocity-based Cartesian controller. Furthermore, we conduct a comparative analysis of MoveIt Servo against the traditional motion planning model OMPL.

OMPL is an open-source library that is a composite of different sampling-based motion planning algorithms [17]. Due to its robustness in encapsulating a variety of planners, it ensures that these algorithms can be seamlessly integrated into a variety of applications. Since these algorithms are sampling-based, they require a search space to sample points. To uphold its reliability, OMPL represents search spaces in a generic way, facilitating a framework where these sampling algorithms are effective regardless of the defined search space. A commonly used sampling algorithm, known as Bidirectional Rapidly Exploring Random Trees (RRT_BIDIRECTIONAL), grows two trees within the search space, searching for states that are common between both trees [22]. Once the trees connect, a path has been found for motion planning. This methodology satisfies conditions in a series of environments ranging from simple to complex. However, this algorithm, like many sampling-based algorithms, faces the limitation of requiring both a start state and a goal state. Such a constraint arises when the goal state is

either unspecified or ambiguous. Specifically, this could be a challenge in procedures such as teleoperation.

In contrast to the aforementioned motion planning algorithms, MoveIt Servo is a velocity-based Cartesian controller interface that takes intended velocities, calculates the expected joint motion in relation to the velocities using a pseudo-inverse Jacobian matrix, and publishes them to the robot's manipulator in real-time [16]. Completely eliminating the need for a start or end goal, MoveIt Servo only needs the current state of each joint of the robot. Ultimately, this methodology eliminates the constraints OMPL faces when tasked with a complex goal, such as teleoperation. Though issues may arise due to real-time motion, MoveIt Servo implements a few safeguards, the first is singularity checking, so the robot does not converge on itself. The second is collision checking, to ensure the robot does not collide with objects in the same scene.

F. Robot Motion Planning Configuration

In this work, MoveIt Servo is chosen as the motion controller interface to conduct human-robot teleoperation collaborations due to its real-time capabilities, smooth end-effector updates, and singularity checking. In the following, we discuss the configuration of MoveIt Servo in relation to the systems framework and the mathematical steps taken to achieve the desired joint updates.

The system framework, as seen in Fig. 1, consists of the MCS streaming real-time velocity and orientation data over a server to a client. Once the client receives this information, it transforms the data into a ROS TwistStamped message. The message can be expressed as a vector $\mathcal{V}$, denoting intended linear and angular velocity:

$$\mathcal{V} = [v_x, v_y, v_z, a_x, a_y, a_z]. \quad (2)$$

After $\mathcal{V}$ is published to the ROS workspace, the node */servo_server* subscribes to the TwistStamped message as well as the */joint_states* topic. The joint states topic provides the current configuration of the robots, which can be expressed as:

$$q = [JS_1, JS_2, JS_3, JS_4, JS_5, JS_6, JS_7]. \quad (3)$$

The */servo_server* then computes the pseudo-inverse of the Jacobian matrix, which maps joint velocities to end-effector velocities, to determine $\boldsymbol{\delta q}$ using the following formula:

$$\boldsymbol{\delta q} = J^{+}(q) \cdot \mathcal{V}, \quad (4)$$

where $\boldsymbol{\delta q}$ represents the joint velocity vector that is able to complete the intended motion in relation to $\mathcal{V}$, based on the current joint configuration q. $\boldsymbol{\delta q}$ is expressed as:

$$\delta q = [\Delta JS_1, \Delta JS_2, \Delta JS_3, \Delta JS_4, \Delta JS_5, \Delta JS_6, \Delta JS_7]. \quad (5)$$

The Jacobian matrix is derived from forward kinematics, which are typically constructed using Denavit-Hartenberg transformation matrices [23]. The modified Denavit-Hartenberg matrix for the Franka Emika robot is presented in Table II.

TABLE II: DENAVIT-HARTENBERG TABLE [23].

Joint	a(m)	d(m)	α(rad)	θ(rad)
1	0	0.333	0	θ_1
2	0	0	-π/2	θ_2
3	0	0.316	π/2	θ_3
4	0.0825	0	π/2	θ_4
5	-0.0825	0.384	-π/2	θ_5
6	0.088	0	π/2	θ6
7	0	0	π/2	θ7

Using the Denavit-Hartenberg matrix and four base matrices ($Rotation_z(\theta_i)$, $Translation_z(d_i)$, $Translation_x(a_i)$, $Rotation_x(\alpha_i)$), we can derive each joint homogeneous transformation matrix. Each base matrix can be defined as [24]:

$$Rotation_z(\theta_i) = \begin{bmatrix} \cos\Theta_i & -\sin\Theta_i & 0 & 0 \\ sin\Theta_i & cos\Theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$Translation_z(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

$$Translation_x(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$Rotation_x(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

To derive the homogeneous transformation matrix of each joint, we plug the values into each base table with respect to the Denavit-Hartenberg matrix, then apply the dot product between each table. This can be described as:

$$HT_i = Rotation_z(\theta_i) * Translation_z(d_i) * Translation_x(a_i) * Rotation_x(\alpha_i) \quad (10)$$

Next, the cumulative transformation matrices will be computed. These matrices define the cumulative transforms from the base frame of the robotic arm to each joint, where i = 7. The method to compute these is:

$$^0HT_i = \prod_{j=1}^{i} {}^{j-1}HT_j \quad (11)$$

Eq. (12) consists of the cumulative transforms from the base link 0 to link 7, each row is written in a symbolic form, as the actual representation is too large and complex to portray. The bottom row indicates the homogeneous row, the right (d_x, d_y, and d_z) represents the positional vector, and the 3x3 grid from r_{11} to r_{33} represents the rotation matrix relative to the base.

$$^0T_7 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Using all 7 cumulative transformation matrices, we can compute each index of the Jacobian matrix. Since the Franka Emika Panda arm consists of revolute joints, meaning they rotate on its Z-axis, we can use the following formula to calculate each index [25].

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{J}_{v_i} \\ \mathbf{J}_{\omega_i} \end{bmatrix} = \begin{bmatrix} {}^0Z_i \times \left({}^0\mathbf{P}_{\text{ee}} - {}^0\mathbf{P}_{\text{joint}_{i-1}} \right) \\ {}^0Z_i \end{bmatrix} \quad (13)$$

^0Z_i refers to the Z-axis of joint J_i, $^0P_{ee}$ is the position vector of the end-effector, and $^0P_{joint\ i-1}$ is the position vector of joint J_{i-1} origin. Referring to Eq. (3), the pseudo-inverse of the Jacobian can be determined, and we can solve for $\boldsymbol{\delta q}$. The node

/servo_server then publishes the δq information to the */velocity_joint_trajectory_controller* at a rate of 100Hz, as seen in Fig. 6. In turn, it establishes real-time movement of the robotic arm in relation to the user's real-life movements.

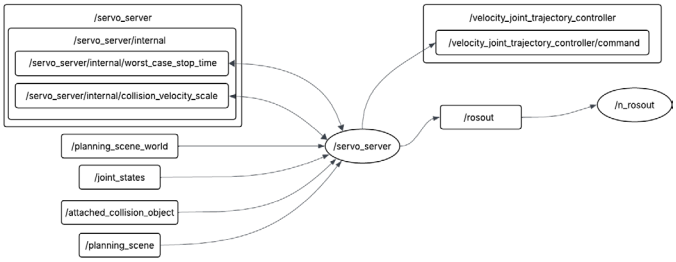

Fig. 6. Rqt_Graph of MoveIt Servo for robot motion planning configuration.

III. Experimental Setup

A. Experimental Platform

The experiment platform consists of a 7-DoF robot arm [26, 27], six Vicon Vero motion capture cameras, and 3D-printed PLA components of a house, including four side walls, a roof, and a base frame with rails for construction replication.

Fig. 7. Experimental platform.

B. User Setup

During the experiment, the user wears one marker ring on the index finger of their dominant hand. As pictured in Fig. 2, the ring holds four retroreflective markers placed equidistantly to minimize occlusions and ensure concise tracking of the user's finger movements. For the purposes of the experiment, the user is positioned next to the robotic arm, within the calibrated tracking space of the Vicon. Although this setup simulates teleoperation, the user remains in close proximity to the robot to ensure accurate tracking of finger movements.

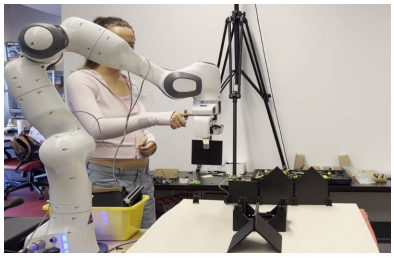

Fig. 8. User setup.

C. Task Description

In this study, the robot is tasked with mimicking the user's intended movements to facilitate the construction of a 3D-printed house. The user portrays a series of movements and actions, guiding the robot to each piece, invoking the grippers to pick them up, and ensuring proper orientation of the end effector. Following, the user then guides the robot to place each piece within the base frame rails. Each base frame rail has a small margin of error ~1.5mm so the user and robot motions should be precise; otherwise, a collision will trigger the robot's force sensor. Once all four side walls have been placed, the user picks up the roof and places it on top. The task is successfully completed if the user and robot experience no collisions or involuntary dropping of house pieces, placing all pieces correctly in the base frame rails.

IV. Results and Analysis

A. Human-Robot Collaboration in Real-World Tasks

To evaluate human-robot collaborative teleoperations, we conducted a series of experiments. In the first experiment, one user is instructed to complete the full task of guiding the robot to successfully build the house. The user is successful in this process, as shown in Fig. 9, and an evaluation of the attempt shows a small displacement between the user's hand and the robot end effector (as presented in Fig. 10). The displacement in the Y-axis is caused by the user walking to a different part of the environment, leading to such a large change. Recording of the experiment is available at: *Experimental Results*.

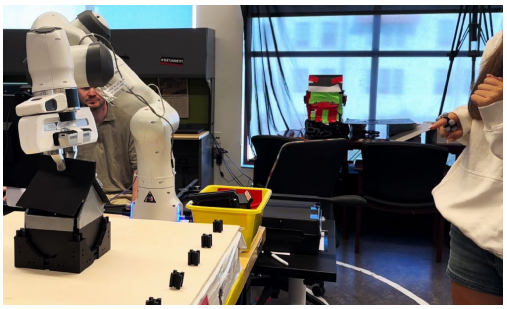

Fig. 9. The user successfully completes the task.

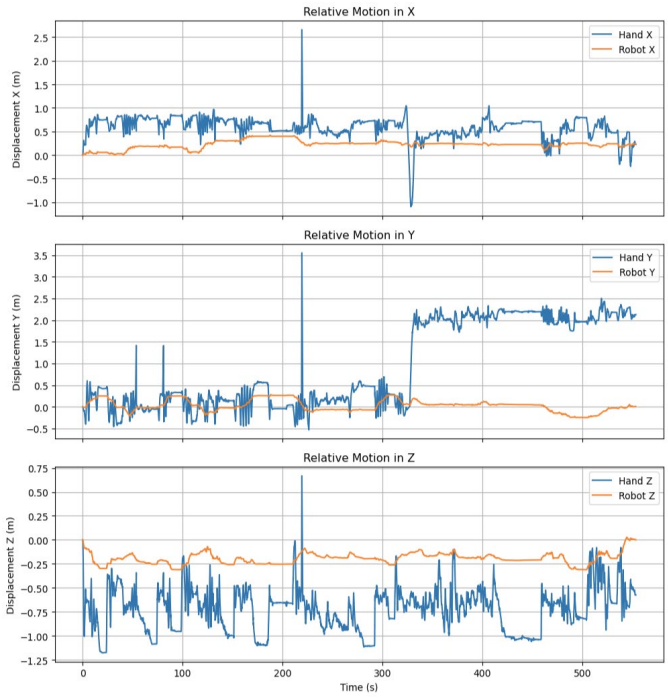

Fig. 10. Relative displacement between the user's hand and robot end effector.

B. Evaluation of MoveIt Servo and OMPL

In the second experiment, as shown in Fig. 11, two users are tasked to repeatedly guide the robot to pick up and place the same piece 5 times using the MoveIt Servo approach. Afterwards, they are tasked to do the same experiment but using OMPL as the motion planner. Results show that both users are able to repetitively and successfully complete the task using MoveIt Servo, but when using the OMPL motion planner, they are unable to complete one iteration. These findings highlight the high precision and accuracy demonstrated by the implemented MoveIt Servo approach over traditional motion planners that use sampling techniques to solve a trajectory in a search space. A demo can be found at: *OMPL Demo*.

MoveIt Servo - User 1				OMPL - User 1			
Attempt	Success	Grabbed Pieces	Unexpected Movement	Attempt	Success	Grabbed Pieces	Unexpected Movement
1	Yes	Yes	No	1	No	No	No
2	Yes	Yes	No	2	No	No	No
3	Yes	Yes	No	3	No	Yes	No
4	Yes	Yes	No	4	No	No	Yes
5	Yes	Yes	No	5	No	Yes	No

MoveIt Servo - User 2				OMPL - User 2			
Attempt	Success	Grabbed Pieces	Unexpected Movement	Attempt	Success	Grabbed Pieces	Unexpected Movement
1	Yes	Yes	No	1	No	No	Yes
2	Yes	Yes	No	2	No	No	No
3	Yes	Yes	No	3	No	No	No
4	Yes	Yes	No	4	No	No	No
5	Yes	Yes	No	5	No	Yes	No

Fig. 11. Evaluation of MoveIt Servo and OMPL.

V. Conclusions and Future Work

This work demonstrates the feasibility of the integration of wearable motion capture using the MCS and real-time robot control to enable intuitive human-robot collaboration. By precisely collecting human motion data in real-time and translating it into robot commands, the developed framework supports teleoperation suitable for remote construction tasks.

In future experiments, we aim to focus on expanding system autonomy by developing a machine learning model capable of learning from human demonstrations and replicating tasks. Enhancing robustness to occlusions and tracking errors will enable the inclusion of more objects and more complex interactions with the MCS, and implementing a full 1:1 motion mapping between user and robot movements will further enhance precision and overall performance during teleoperation. Together, these potential improvements will enable high-precision robot motion planning for complex tasks in dynamic environments. Such advancements are critical for extending human-robot collaboration to remote settings, including space exploration and extraterrestrial construction.

Acknowledgment

This work is supported in part by the National Science Foundation under Grants CMMI-2338767 and CNS-2117308.

References

[1] E. Appleton and D. J. Williams, *Industrial robot applications*. Springer Science & Business Media, 2012.

[2] W. Wang, Y. Chen, R. Li, Z. Zhang, V. Krovi, and Y. Jia, "Human-Robot Collaboration for Advanced Manufacturing by Learning from Multi-Modal Human Demonstrations," *Recent Advances in Industrial Robotics, WSPC,* pp. 1-29, 2019.

[3] R. Li, W. Wang, Y. Chen, S. Srinivasan, and V. N. Krovi, "An End-to-End Fully Automatic Bay Parking Approach for Autonomous Vehicles," in *Dynamic Systems and Control Conference*, 2018, pp. 1-6.

[4] W. Wang, Z. Przedworska, J. Parron, M. Lyons, M. Zhu, and A. Tuininga, "MCROS: A Multimodal Collaborative Robot System for Human-Centered Tasks," in *2024 International Conference on Networking, Sensing and Control (ICNSC)*, 2024: IEEE, pp. 1-6.

[5] O. Obidat, G. Modery, W. Wang, X. Guo, and M. Zhou, "A Multifaceted User Study for the Teaching-Learning-Prediction-Collaboration Framework in Human-Robot Collaborative Tasks," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 2895-2900.

[6] M. Kyrarini *et al.*, "A survey of robots in healthcare," *Technologies,* vol. 9, no. 1, p. 8, 2021.

[7] T. T. Nguyen, J. Parron, O. Obidat, A. R. Tuininga, and W. Wang, "Ready or Not? A Robot-Assisted Crop Harvest Solution in Smart Agriculture Contexts," in *2023 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2023: IEEE, pp. 373-378.

[8] C. Hannum, R. Li, and W. Wang, "A Trust-Assist Framework for Human-Robot Co-Carry Tasks," *Robotics,* vol. 12, no. 2, pp. 1-19, 2023.

[9] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," *IEEE Robotics & Automation Magazine,* vol. 14, no. 1, pp. 90-103, 2007.

[10] P. Berthet-Rayne, *et al.*, "Inverse kinematics control methods for redundant snakelike robot teleoperation during minimally invasive surgery," *IEEE Robotics and Automation Letters,* vol. 3, no. 3, pp. 2501-2508, 2018.

[11] J. Parron, T. T. Nguyen, and W. Wang, "Development of A Multimodal Trust Database in Human-Robot Collaborative Contexts," in *IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2023: IEEE, pp. 0601-0605.

[12] H. Tang, *et al.*, "Human-robot collaborative teleoperation system for semi-autonomous reconnaissance robot," in *International Conference on Mechatronics and Automation*, 2009: IEEE, pp. 1934-1939.

[13] J. S. Lee, Y. Ham, H. Park, and J. Kim, "Challenges, tasks, and opportunities in teleoperation of excavator toward human-in-the-loop construction automation," *Automation in Construction,* p. 104119, 2022.

[14] P. Schmaus *et al.*, "Knowledge driven orbit-to-ground teleoperation of a robot coworker," *IEEE Robotics and Automation Letters,* pp. 143-150, 2019.

[15] Vicon Vero Technical Information, https://www.vicon.com/hardware/cameras/vero.

[16] Moveit_servo, https://index.ros.org/p/moveit_servo.

[17] I. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine,* vol. 19, no. 4, pp. 72-82, 2012.

[18] W. Wang, Y. Chen, and Y. Jia, "Evaluation and Optimization of Dual-Arm Robot Path Planning for Human–Robot Collaborative Tasks in Smart Manufacturing Contexts," *ASME Letters in Dynamic Systems and Control,* vol. 1, no. 1, 2020, doi: 10.1115/1.4046577.

[19] S. Bier, R. Li, and W. Wang, "A Full-Dimensional Robot Teleoperation Platform," in *2020 IEEE International Conference on Mechanical and Aerospace Engineering*, 2020: IEEE, pp. 186-191.

[20] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*, 2009, pp. 489-494.

[21] M. Kalakrishnan, *et al.*, "STOMP: Stochastic trajectory optimization for motion planning," in *IEEE international conference on robotics and automation*, 2011, pp. 4569-4574.

[22] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research,* pp. 378-400, 2001.

[23] C. Gaz, et al., "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters,* pp. 4147-4154, 2019.

[24] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.

[25] F. Merat, "Introduction to robotics: Mechanics and control," *IEEE Journal on Robotics and Automation,* vol. 3, no. 2, pp. 166-166, 1987.

[26] H. Diamantopoulos and W. Wang, "Accommodating and Assisting Human Partners in Human-Robot Collaborative Tasks through Emotion Understanding," in *2021 International Conference on Mechanical and Aerospace Engineering (ICMAE)*, 2021: IEEE, pp. 523-528.

[27] I. Jacoby, J. Parron, and W. Wang, "Understanding Dynamic Human Intentions to Enhance Collaboration Performance for Human-Robot Partnerships," in *2023 IEEE MIT Undergraduate Research Technology Conference (URTC)*, 2023: IEEE, pp. 1-6.