electronics

MDPI

Article

Skeletal Image Features Based Collaborative Teleoperation Control of the Double Robotic Manipulators

Hsiu-Ming Wu [1,*] and Shih-Hsun Wei [2]

[1] Department of Intelligent Automation Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[2] Institute of Mechatronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan; swei80850@gmail.com
* Correspondence: hmwu@mail.ntut.edu.tw

Abstract

In this study, a vision-based remote and synchronized control scheme is proposed for the double six-DOF robotic manipulators. Using an Intel RealSense D435 depth camera and MediaPipe skeletal image feature technique, the operator's 3D hand pose is captured and mapped to the robot's workspace via coordinate transformation. Inverse kinematics is then applied to compute the necessary joint angles for synchronized motion control. Implemented on double robotic manipulators with the MoveIt framework, the system successfully achieves a collaborative teleoperation control task to transfer an object from a robotic manipulator to another one. Further, moving average filtering techniques are used to enhance trajectory smoothness and stability. The framework demonstrates the feasibility and effectiveness of non-contact, vision-guided multi-robot control for applications in teleoperation, smart manufacturing, and education.

Keywords: skeletal image features; MediaPipe; collaborative teleoperation control; robotic manipulators

check for updates

Academic Editor: Giuseppe Prencipe

Received: 23 August 2025
Revised: 28 September 2025
Accepted: 29 September 2025
Published: 30 September 2025

Citation: Wu, H.-M.; Wei, S.-H. Skeletal Image Features Based Collaborative Teleoperation Control of the Double Robotic Manipulators. *Electronics* **2025**, *14*, 3897. https://doi.org/10.3390/electronics14193897

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Due to the rapid advancement of intelligent automation and human–robot collaboration technologies, dual-arm robotic systems have found increasing applications in domains requiring complex operations and high adaptability, such as precision assembly, remote manipulation, healthcare, and disaster rescue. Compared to conventional single-arm robots, dual-arm systems emulate the natural coordination of human bimanual activities, enabling operations that necessitate synchronized motion—such as bilateral grasping, object rotation, or cooperative assembly. This capability substantially expands the functional envelope of robotic automation. Nonetheless, achieving synchronized and coordinated control in dual-arm robotic systems remains a formidable challenge. It necessitates real-time trajectory synchronization, directional control, spatial alignment, and accurate interpretation of operator intent to ensure coherent and stable dual-arm interactions. Existing approaches predominantly rely on wearable technologies (e.g., data gloves [1] or inertial sensors [2]) to capture user motion. Although it is effective in tracking hand gestures, such systems often compromise user comfort and restrict natural movement.

To overcome the limitations associated with wearable-based motion capture systems, this study proposes a non-contact, vision-based human–robot interaction framework for dual-arm control. By leveraging depth cameras in conjunction with the MediaPipe framework [3,4], the system obtains real-time three-dimensional positional and orientation data

of the operator's hands. This data is translated into the robot's operational space through coordinate transformation and kinematic modeling, with inverse kinematics subsequently employed to compute the joint angles required for precise and synchronous dual-arm imitation control. The proposed framework emphasizes high precision and operational flexibility, making it suitable for applications such as teleoperation, cooperative object transportation, and interactive dual-arm manipulation tasks. This vision-based approach enhances system intelligence, naturalness, and usability by eliminating the physical constraints of wearable devices.

As robotic systems evolve from single-task automation to multi-task and collaborative functionalities, the limitations of traditional single-arm configurations become increasingly evident. Although single-arm robots are valued for their mechanical simplicity, low power consumption, and ease of integration, they are inherently constrained in performing tasks requiring spatial versatility and complex coordination. For instance, Peta et al. [5] conducted a comparative analysis of single- and dual-arm robotic systems in electronic assembly workflows, revealing that dual-arm systems achieved a 20% reduction in task completion time by executing operations in parallel—unlike single-arm systems, which must complete actions sequentially. While dual-arm robots entail higher initial costs, their superior adaptability and long-term operational efficiency justify the investment. In particular, tasks involving dual-point interaction—such as lifting large objects, opening containers, or using tools in both hands—pose significant challenges for single-arm systems. Motoda et al. [6] emphasized that such tasks often require additional mechanical complexity or repeated repositioning when performed by single-arm robots, thereby increasing control difficulty and the likelihood of execution failure. In contrast, dual-arm robotic systems are inherently equipped to perform synchronized bimanual tasks, closely replicating human manipulation strategies and achieving higher degrees of precision, fluidity, and coordination. These comparative findings highlight a clear trajectory: while single-arm robots remain effective for simple, repetitive tasks, dual-arm systems are becoming indispensable in scenarios that demand concurrent actions, dynamic force coordination, and interactive collaboration. As such, the shift from single- to dual-arm control represents a pivotal step in advancing human–robot collaboration. Addressing this technological transition, the present study develops a control strategy that eschews wearable devices in favor of vision-based hand tracking. Prior works, such as Wu et al. [7], employed wearable gloves with haptic feedback; however, these solutions are frequently criticized for discomfort, reduced accuracy due to anatomical variability, and operator fatigue during extended use. Even low-latency, open-structured gloves cannot fully eliminate the drawbacks of physical contact, which continues to impair operator freedom and interaction fluidity. By contrast, the proposed non-contact vision-based hand tracking system significantly improves usability and comfort, making it a compelling alternative for high-precision dual-arm robotic control in complex and dynamic environments.

In this study, we employ contactless skeletal image tracking for capturing human posture. Two dominant methodologies for pose estimation exist: top-down and bottom-up. Top-down approaches first localize the human body and then extract keypoints within the region, while bottom-up methods detect all keypoints before grouping them into individuals. According to comparative analyses [8,9], MediaPipe—a top-down model—demonstrates advantages in computational efficiency and real-time tracking, particularly during continuous motion. Sarah Mroz et al. [9] conducted a comprehensive performance evaluation between MediaPipe and the bottom-up OpenPose framework [10–12], concluding that MediaPipe is more suitable for real-time applications due to its lightweight design and reduced computational load. Although the OpenPose exhibits higher static accuracy, its frame-by-frame processing demands limit its applicability in dynamic settings.

Despite these advances, most existing dual-arm teleoperation frameworks rely on contact- or marker-based tracking, which constrains movement naturalness and user intuitiveness. Furthermore, many proposed systems are validated only in simulation, lacking real-world hardware implementation and verification. Given the increasing demand for robotic solutions in hazardous or intricate environments, enabling intuitive, remote control of dual-arm robots represents a significant step forward. This study proposes a teleoperation control strategy that enables synchronous control of two six-degree-of-freedom robotic manipulators via contactless hand tracking. The system incorporates an Intel RealSense D435 depth camera [13] to capture three-dimensional positional and depth information of the operator's hands, with each arm corresponding to the operator's respective hand. To emulate real-world industrial tasks, we further examine the coordination of multiple robotic manipulators operating in parallel. The methodology addresses two primary challenges. First, the system must accurately track the operator's hand pose and convert palm spatial coordinates into the robot's reference frame using vision-based mapping and proportional scaling. This requires robust skeletal tracking and precise spatial transformation. Other vision-based gesture recognition approaches such as the work in [14] employ OpenCV-based handcrafted image processing techniques (e.g., color thresholding and contour extraction) that are often sensitive to illumination variations and only provide coarse two-dimensional information, or additionally the work of Marić et al. [15] utilizes RGB-D palm tracking but remains limited in robustness and generalizability. However, our proposed strategy leverages a common depth camera in conjunction with a skeletal feature extraction technique where this design obviates the need for specialized or contact-based sensors while enabling the acquisition of accurate three-dimensional positional and orientational data of the operator's hands. Accordingly, a notable advantage of this work lies in its enhanced robustness to environmental variations and its flexibility to accommodate different depth cameras as well as diverse robotic manipulator platforms, thereby improving system scalability and applicability in practical scenarios. Second, motion mapping must translate the captured hand data into joint-space parameters via inverse kinematics, followed by forward kinematics to derive the end-effector trajectories. Dealing with these challenges enables real-time, non-contact, and synchronized dual-arm robot control that replicates human bimanual manipulation with high precision, adaptability, and stability.

2. MediaPipe Hand Gesture Detection

Teleoperation employs the operator's hands as tracking targets. To accurately capture hand gestures, skeletal recognition technology is essential. Given the variety of skeletal recognition methods, each with distinct approaches extracts skeletal feature points. This work evaluates and selects the MediaPipe skeletal feature recognition framework for its validation. The MediaPipe hand gesture detection framework, as depicted in Figure 1, consists of two model architectures: BlazePalm and BlazeHand [16]. The BlazePalm model, indicated by the dashed line in the figure, is responsible for detecting and localizing the hand. It performs initial detection on the first frame at the Real-Time Flow Limiter stage, with subsequent hand detection inferred from the previous frame's data, thereby eliminating the need for detection in every frame and significantly enhancing computational efficiency. The Detection to Rectangle module generates a rectangular bounding box around the detected hand based on its position. The BlazeHand model extracts hand feature points, with the Image Cropping module isolating the hand image using the rectangular bounding box. The Hand Landmark module then predicts 21 key points of the hand from the cropped image and overlays these feature points onto the hand within the image. To ensure robustness, the model evaluates the alignment confidence between the hand and its feature points. Should the confidence fall below a threshold, the system reverts to

the BlazePalm model for re-detection. By integrating these two models, the framework achieves comprehensive and real-time output of 21 hand feature points.

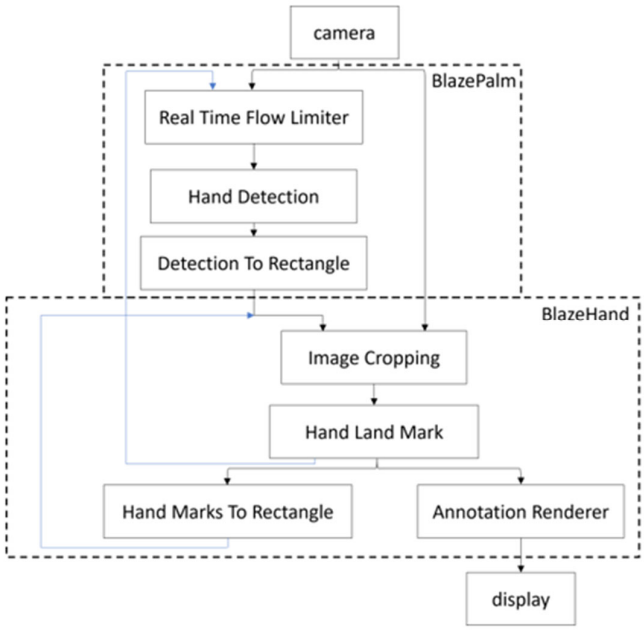


Figure 1. MediaPipe hand detection framework.

The BlazePalm detector is developed by modifying the BlazeFace [17,18] detector, incorporating three additional convolutional layers corresponding to feature maps of sizes 16 × 16, 32 × 32, and 64 × 64. These layers utilize residual connections to mitigate data loss and gradient vanishing issues. The overall architecture is illustrated in Figure 2. As this detector is specifically designed for palm detection, it can be more accurately defined the bounding box for the palm compared to detecting the entire hand.

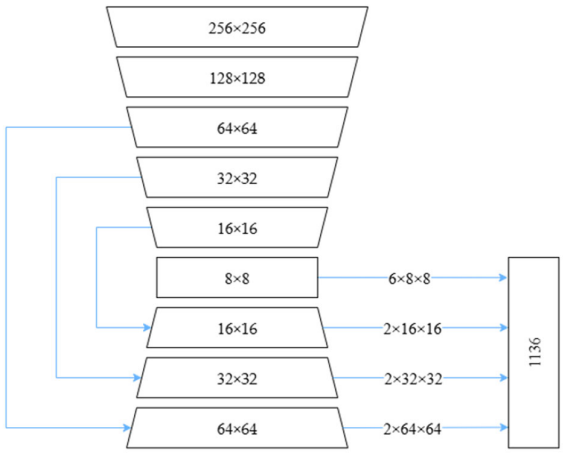


Figure 2. BlazePalm hand detector architecture.

The BlazeHand model, following palm detection, precisely identifies 21 hand landmark points within the palm region of the image, as illustrated in Figure 3. This model maintains robust stability even in cases of partially visible or occluded palms. It produces three outputs: the coordinate positions of the 21 hand landmark points, an indication of whether a palm is detected as shown in Figure 4, and information distinguishing between the left and right hands. Control of the left and right hand grippers is based on the center point between palm points 1 and 5, which corresponds to the position of the robotic arm's

end-effector, while the distance between points 4 and 8 determines if the gripper opens or not.

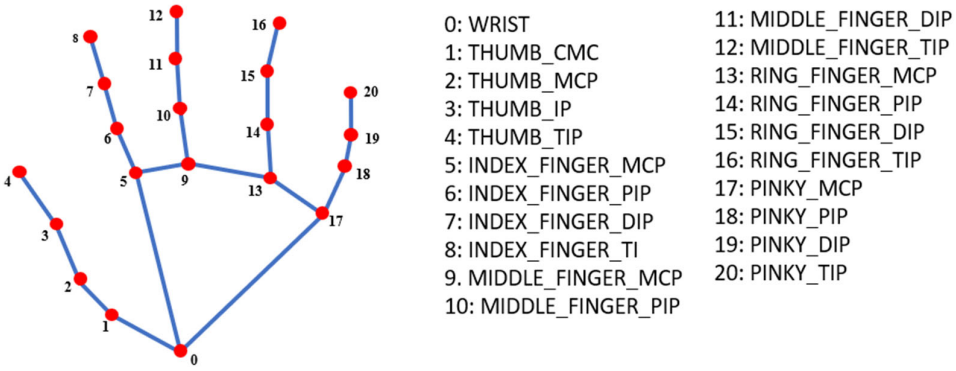


Figure 3. Schematic of BlazeHand 21 hand landmark points.

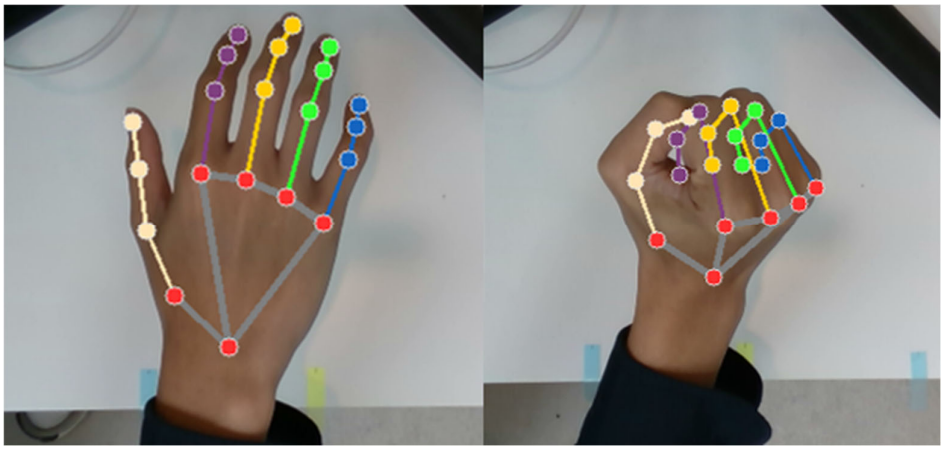


Figure 4. Scenarios for palm detection.

3. Kinematics of Six-Degree-of-Freedom Robotic Manipulators and Problem Statement

The kinematic equations of a robot constitute a mathematical representation that describes the relationship between the robot's structural parameters, input joint angles, and resulting motion. Through kinematic analysis, it becomes possible to accurately predict changes in the robot's posture under given input conditions. In addition, kinematics reveals the robot's workspace—the region that the end-effector can reach and operate within—and facilitates a comprehensive understanding of the robot's motion characteristics. Such mathematical modeling is essential for gaining in-depth insight into the system and for enhancing the performance of robotic motion control. In addition, it also can be used to check and validate whether the solutions solved from inverse kinematics are correct. The workspace can be formally defined as the set of all positions and orientations that the robotic manipulator's end-effector is capable of attaining.

The control objective of the study is to utilize both hands for collaborative teleoperation control of double robotic manipulators based on skeletal images. A depth camera is used to capture skeletal image features of both hands and filter them. Coordinate points in the 3D space are obtained from image features via the coordinate transformation. Then, inverse kinematics is applied and control of desired angles of each joint are achieved in the closed-loop by means of the MoveIt package in the ROS. The overall control block diagram is shown in the following Figure 5.

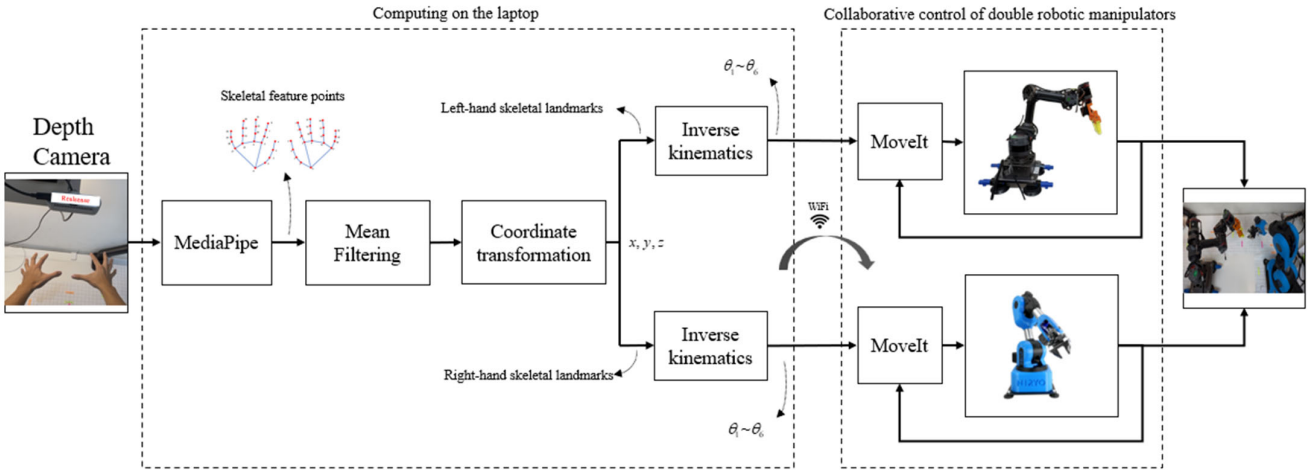


Figure 5. The overall control block diagram.

3.1. Forward Kinematics of Six-Degree-of-Freedom Robotic Manipulators

The homogeneous transformation matrix is a fundamental representation in forward kinematic analysis of robotic manipulators. It enables compact and efficient expression and computation of rigid body motion in three-dimensional space using matrix operations. In robotics, the spatial relationship between successive links is typically described by a series of homogeneous transformation matrices between adjacent coordinate frames. This representation facilitates the simulation and calculation of a robotic manipulator's motion trajectory, as well as the position and orientation of its end-effector, in a more intuitive and computationally efficient manner. Prior to performing kinematic analysis, a Denavit–Hartenberg (D-H) parameter table is usually established to define the geometric relationships between joints and links. The homogeneous transformation matrix incorporates both rotational and translational motion into a unified 4×4 matrix structure, providing a mathematically convenient tool for handling 3D coordinate transformations. For a robotic manipulator composed of multiple revolute joints, the rotation of each joint is described by a rotation matrix, while the displacement between links is represented by a translation matrix. By sequentially multiplying these transformation matrices, one can derive the overall homogeneous transformation matrix from the base frame to the end-effector, thus fully characterizing the manipulator's spatial configuration. In the work, double robotic manipulators are used for collaboration. One is the Niryo Ned [19] robotic manipulator with six-DOFs whose dimensional information is presented in Figure 6, and the other one is Gluon [20] six-axis robotic manipulator whose dimensional information is represented in Figure 7.

After making sure the precise physical dimensions of the robotic manipulators, the next step is to define the coordinate frames for each axis and establish the corresponding kinematic model to enable accurate control of the robot's motion and positioning. This model can be constructed using the D-H parameter convention, a standardized modeling method used to describe the geometric relationships between links and joints. By defining a unified set of parameters, the complex mechanical structure of the robotic manipulator can be transformed into a tractable set of mathematical expressions, thereby simplifying and systematizing subsequent kinematic and dynamic analyses. Accordingly, four primary D-H parameters are used: link length a_i (unit: mm), link offset d_i (unit: mm), link twist angle α_i (unit: degrees), and joint angle θ_i (unit: degrees). These parameters describe the relative position and orientation between each joint and its adjacent link. Figure 8 illustrates the definition of coordinate frames for each joint of the Niryo Ned robotic manipulator, from CS0 (Coordinate System 0) to CS8. Table 1 presents the corresponding D-H parameter table

for the Niryo Ned, which clearly defines and characterizes the geometric configuration and kinematic properties of the manipulator.

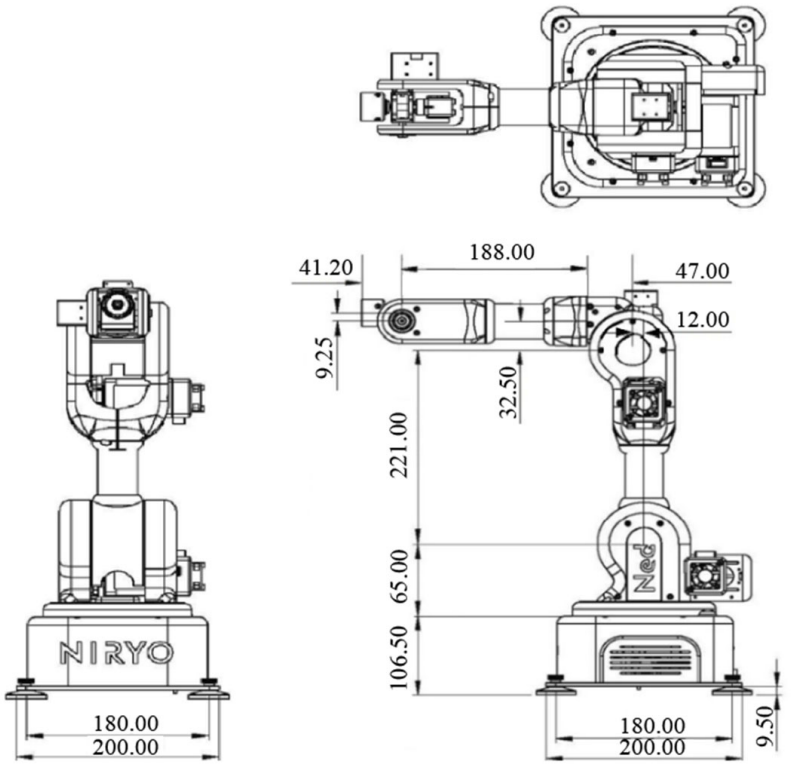


Figure 6. Dimensional diagram of the Niryo Ned robotic manipulator. (Unit: mm).

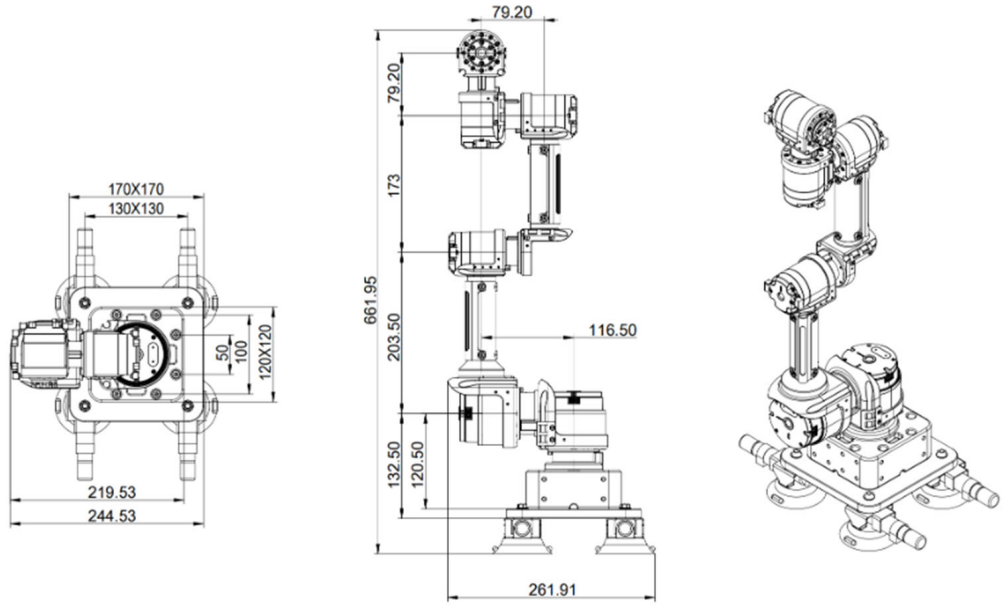


Figure 7. Dimensional diagram of the Gluon robotic manipulator. (Unit: mm).

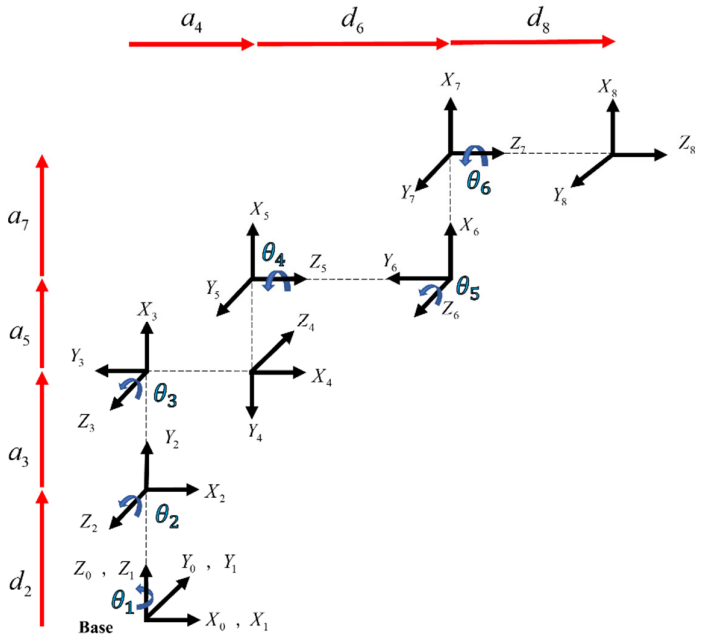


Figure 8. Establishment of the coordinate systems for each joint of the Niryo Ned robotic manipulator.

Table 1. The D-H parameter table of the Niryo Ned robotic manipulator.

Link	α_i	a_i	d_i	θ_i
0	0	0	0	θ_1
1	$\pi/2$	0	171.5	θ_2
2	0	221	0	θ_3
3	0	12	0	$\pi/2$
4	$-\pi/2$	32.5	0	θ_4
5	$-\pi/2$	0	235	θ_5
6	$\pi/2$	9.3	0	θ_6
7	0	0	41.2	0

Based on the coordinate axes defined at each joint of the robotic manipulator, the D-H parameter table, as shown in Table 1, can be established. From Joint 1 to Joint 6, the homogeneous transformation matrix of coordinate frame {1} relative to frame {0} is denoted as T_1^0, and the transformation of frame {2} relative to frame {1} is denoted as T_2^1. Following this pattern, the homogeneous transformation matrices of the Niryo Ned robotic manipulator can be derived and are represented by Equations (1)–(8). By successively multiplying these matrices up to the sixth joint, the complete transformation is expressed as Equation (9).

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$T_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$T_3^2 = \begin{bmatrix} -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ \cos(\theta_3) & -\sin(\theta_3) & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$T_4^3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -a_3 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$T_5^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -\cos(\theta_4) & \sin(\theta_4) & 0 & -a_4 \\ -\sin(\theta_4) & -\cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$T_6^5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_5) & -\cos(\theta_5) & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$T_7^6 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & a_6 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$T_8^7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$T_8^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5 \cdot T_7^6 \cdot T_8^7 \tag{9}$$

The second Gluon robotic manipulator, as illustrated in Figure 9, has its D-H parameters listed in Table 2. Based on the configuration, coordinate frames can be established at each joint accordingly. The corresponding homogeneous transformation matrices for the Gluon robotic manipulator, denoted as Equations (10)–(16), are derived based on these coordinate frames. By sequentially multiplying these matrices up to the sixth joint, the overall transformation can be expressed as Equation (17).

$$T_1^0 = \begin{bmatrix} \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ 0 & 0 & -1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$T_2^1 = \begin{bmatrix} -\cos(\theta_2) & \sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & d_2 \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$T_3^2 = \begin{bmatrix} \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$T_4^3 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & a_4 \\ -\sin(\theta_4) & -\cos(\theta_4) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$T_5^4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$T_6^5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$T_7^6 = \begin{bmatrix} \sin(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & -1 & -d_7 \\ -\cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

$$T_7^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5 \cdot T_7^6 \tag{17}$$

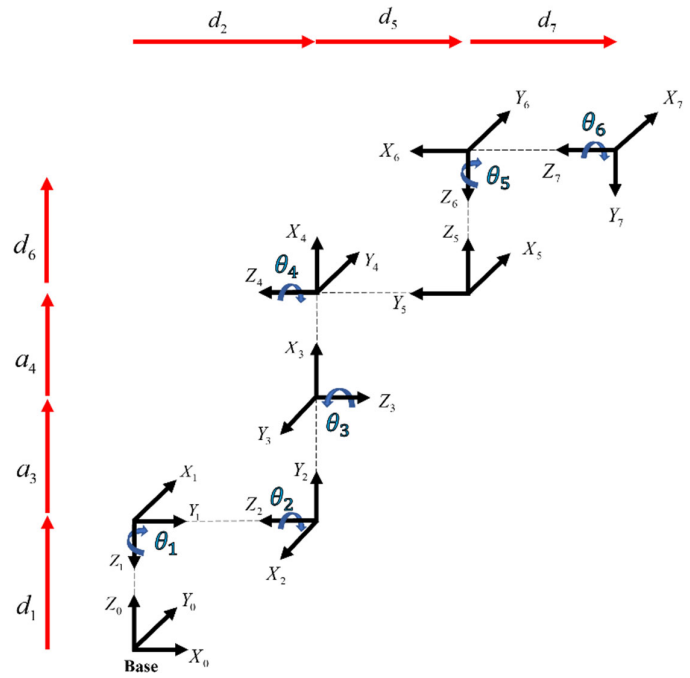


Figure 9. Establishment of the coordinate systems for each joint of the Gluon robotic manipulator.

Table 2. The D-H parameter table of the Gluon robotic manipulator.

Link	α_i	a_i	d_i	θ_i
0	0	0	0	θ_1
1	$\pi/2$	0	70	θ_2
2	0	210	0	θ_3
3	0	173	0	θ_4
4	$\pi/2$	0	40	$\pi/2$
5	0	0	79.2	θ_5
6	$-\pi/2$	0	112	θ_6

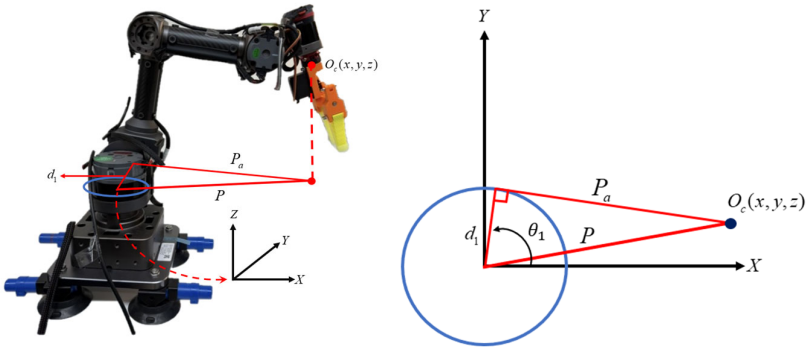
3.2. Inverse Kinematics of Six-Degree-of-Freedom Robotic Manipulators

The homogeneous transformation matrix is not only a fundamental tool in forward kinematics—used to compute the position and orientation of the end-effector given specific joint angles—but also plays a critical role in solving inverse kinematics problems. The goal of inverse kinematics is to determine the set of joint angles required for the end-effector to reach a desired position and orientation. By performing algebraic operations on the homogeneous transformation matrices, the corresponding joint variables can be derived. In this study, a geometric approach is adopted to calculate the joint angles of the robotic manipulator. The geometric method is characterized by its intuitive and easily understandable nature, as it determines the reachable workspace by analyzing the structural geometry of the robot. Specifically, the first three joint angles of the manipulator are computed in order to guide the end-effector to the target position. As shown in Figure 10, the first joint angle corresponds to the rotation about the X-axis toward the link d_1 of the Gluon robotic manipulator. O_C denotes the position of the end-effector, P represents the distance from the origin of the robotic manipulator to the end-effector, and P_a is the combined length of Link 2 and Link 3 of the Gluon manipulator. The first joint angle can be calculated using Equations (18)–(20).

$$P = \sqrt{x^2 + y^2} \tag{18}$$

$$P_a = \sqrt{P^2 - d_1^2} \tag{19}$$

$$\theta_1 = \cos^{-1}\left(\frac{d_1}{P}\right) + \tan^{-1}\left(\frac{y}{x}\right) \tag{20}$$

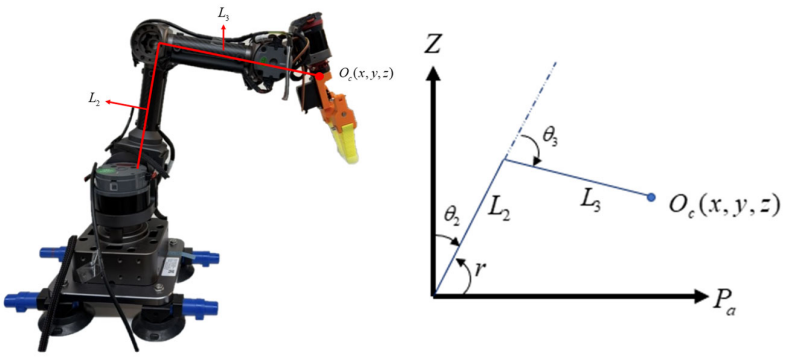


(**a**) (**b**)

Figure 10. The first joint angle of the (**a**) Gluon robotic manipulator and (**b**) schematic diagram.

In inverse kinematics, the computation of the second joint angle θ_2 and the third joint angle θ_3 is based on the known lengths of Link 2 and Link 3. As illustrated in Figure 11, the angle θ_2 corresponds to the rotation of the robotic manipulator about the Z-axis toward the second joint, while θ_3 represents the angle between Link 2 and Link 3. The angle θ_3 can be determined using the law of cosines. Once θ_3 is obtained, θ_2 can be calculated through trigonometric relationships. As shown in Figure 10, the angle r denotes the angle between Link 2 and the axis P_a, and it is defined by Equation (21). Using the angle r, in conjunction with θ_3, θ_2 can then be derived via trigonometric relations.

$$\cos(\theta_3) = \frac{L_2^2 + L_3^2 - P_a^2 - Z^2}{2L_2L_3} = D \tag{21}$$



(**a**) (**b**)

Figure 11. The second and third joint angles of the (**a**) Gluon robotic manipulator and (**b**) schematic diagram.

The above expression can be represented as D, from which the following equation can be derived

$$\sin(\theta_3) = \pm\sqrt{1 - D^2} \tag{22}$$

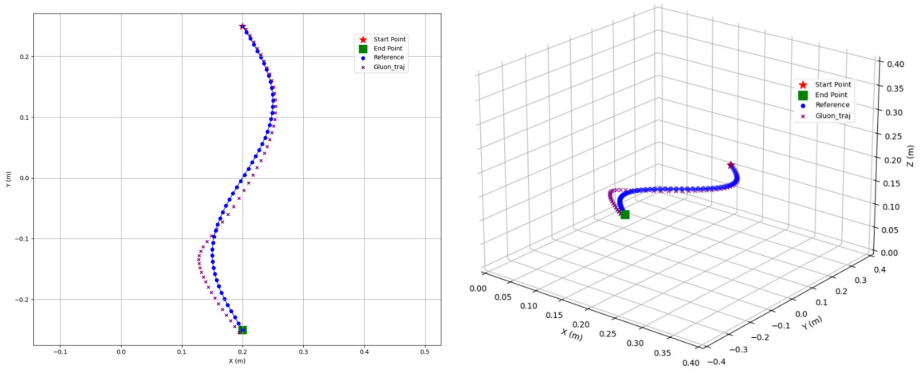
$$\theta_3 = \tan^{-1}\left(\frac{\pm\sqrt{1 - D^2}}{D}\right) \tag{23}$$

$$r = \tan^{-1}\left(\frac{Z}{P_a}\right) - \tan^{-1}\left(\frac{L_2\sin(\theta_3)}{L_1 + L_2\cos(\theta_3)}\right) \tag{24}$$

$$\theta_2 = \frac{\pi}{2} - r \tag{25}$$

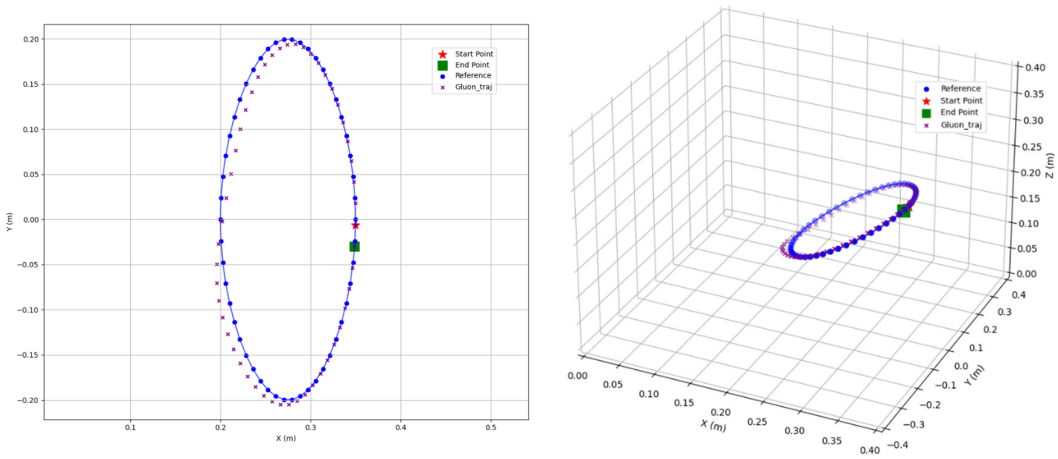
After completing the inverse kinematics calculations, the joint angles of the first three axes of the robotic manipulator can be obtained. These joint angles are then substituted into the forward kinematics model to determine the position of the end-effector. Figures 12 and 13 illustrate the position of the Gluon robotic manipulator's end-effector obtained through this kinematic approach. Based on the computed positions, a motion trajectory is designed to guide the end-effector through space. Kinematic validation is conducted in both the 2D XY plane and the 3D XYZ space. In the 2D case, a sinusoidal trajectory, as described in Equation (26), is employed—where the X-axis and Y-axis values correspond to the coordinates of the motion. This trajectory is discretized into 50 points, beginning at the position (0.2, 0.25) and ending at (0.2, −0.25).

$$x = 0.2 + \sin\left(\frac{2}{\pi}y - \frac{\pi}{2}\right) \tag{26}$$



(**a**) (**b**)

Figure 12. Sinusoidal motion trajectory of the Gluon robotic manipulator shown in the (**a**) 2D space and (**b**) 3D space.



(**a**) (**b**)

Figure 13. Circular motion trajectory of the Gluon robotic manipulator shown in the (**a**) 2D space and (**b**) 3D space.

An elliptical trajectory in 3D space is formulated and analytically described through parametric equations. The ellipse is centered at the point (0.275, 0, 0.15), and its motion is defined by the following Equations (27)–(29).

$$x(t) = 0.275 + 0.075cos(t) \times \text{ux} \tag{27}$$

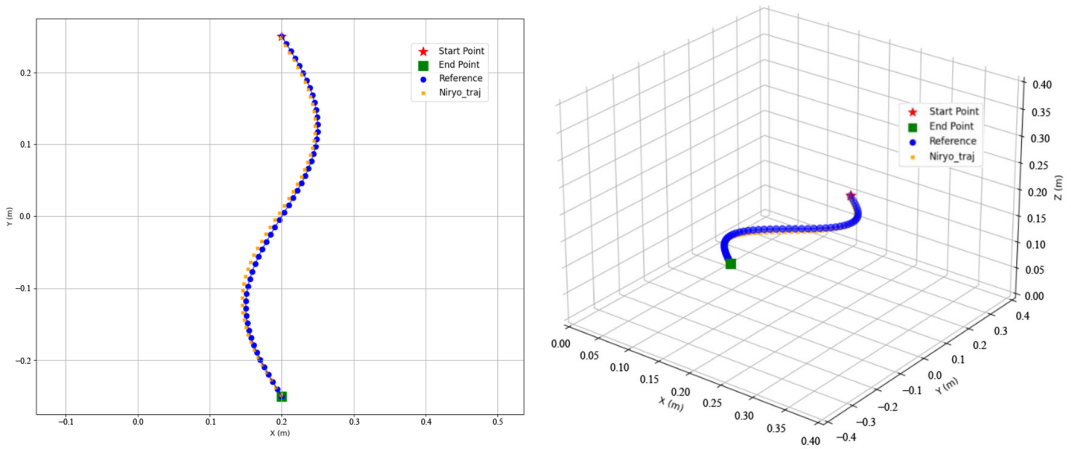
$$y(t) = 0.2sin(t) \tag{28}$$

$$z(t) = 0.15 + 0.075cos(t) \times \text{uz} \tag{29}$$

In this context, t serves as the trajectory parameter, used to describe position variations along the elliptical path. The ellipse has a semi-major axis length of 0.2 and a semi-minor axis length of 0.075. It lies within the XY plane and is inclined at an angle of 33.7° with respect to the positive X-axis. The coefficients ux $=$ 0.832 and uz $=$ 0.555 represent the unit vector components of the semi-minor axis direction along the X and Z axes, respectively. A key characteristic of this trajectory is its non-planar nature. Unlike a conventional 2D ellipse, this trajectory does not lie entirely within any of the standard Cartesian planes (i.e., XY, XZ, or YZ planes). Instead, it is distributed over an inclined plane in 3D space. This spatial configuration enables precise path control in complex environments, providing a solid mathematical foundation for subsequent motion planning and trajectory tracking. The motion begins at the starting point (0.337, 0, 0.192) while $t = 0$ and proceeds to the top vertex (0.275, 0.2, 0.15) while $t = \pi/2$ and continues to the opposite side point (0.213, 0, 0.108) at $t = \pi$, then to the bottom vertex (0.275, -0.2, 0.15) at $t = 3\pi/2$, and finally returns to the starting point, completing the elliptical trajectory. These key position coordinates validate the accuracy of the trajectory design and serve as reference points for path verification in practical applications.

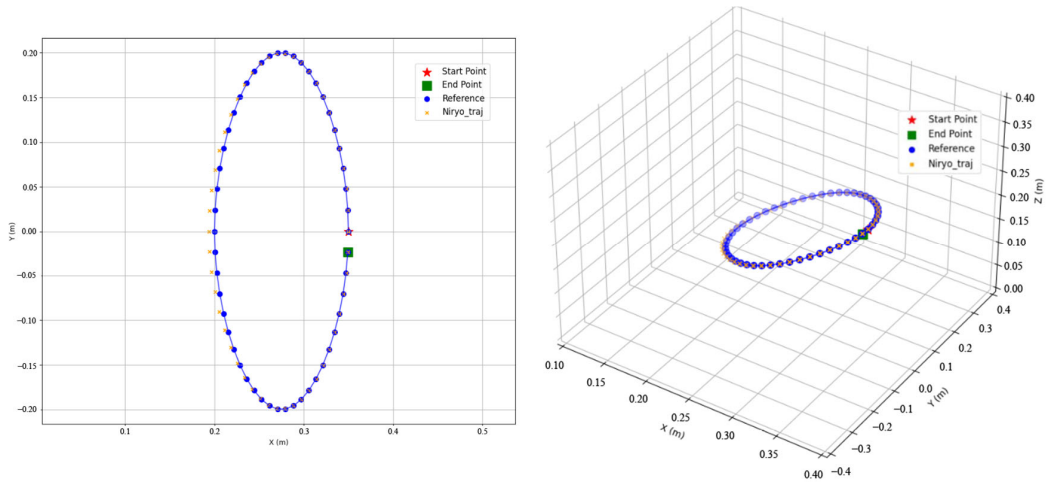
In this study, the Niryo Ned robotic manipulator and the Gluon robotic manipulator employ distinct inverse kinematics solvers. The inverse kinematics for the Niryo Ned robotic manipulator is computed using MoveIt within the Robot Operating System (ROS) [21]. Figures 14 and 15 validate the accuracy of its inverse kinematics. After processing the transformed coordinates through inverse kinematics, the position of the end-effector is analyzed using the previously discussed forward kinematics. The validated trajectory motion aligns with the trajectory of the Gluon robotic manipulator. As observed in the figures, the kinematics of the Niryo Ned robotic manipulator is calculated with high precision,

with minimal deviation between the motion trajectory and the target trajectory. Therefore, this study employs two distinct kinematic algorithms for the experimental validation.



(**a**) (**b**)

Figure 14. Sinusoidal motion trajectory of the Niryo Ned robotic manipulator in the (**a**) 2D space and (**b**) 3D space.



(**a**) (**b**)

Figure 15. Circular motion trajectory of the Niryo Ned robotic manipulator in the (**a**) 2D space and (**b**) 3D space.

4. Mapping from Image Coordinates to the Coordinate System of the Robotic Manipulators

We employ a linear mapping method here, which differs from the conventional camera coordinate transformation. Although the pixel value range remains consistent across different depths, the real-world scene captured by the camera varies in terms of its upper and lower bounds at different depth levels. Therefore, the proposed mapping method applies a depth-specific compensation to ensure that the transformation from the palm coordinate space to the robotic manipulator's movement space is free from distortion. Moreover, this mapping method does not require consideration of the extrinsic parameters between the depth camera and the robotic manipulators, only the image distance of the depth camera needs to be taken into account. The study aims to enable gesture-based control of a robotic manipulator through vision input by accurately converting the pixel coordinates of the hand in the image into three-dimensional spatial coordinates recognizable and controllable by the robot. The mapping relationship is defined in Equations (42)–(45).

The hand's position in the image is represented by pixel coordinates (x_{Tp}, y_{Tp}) and a depth value *depth*, which is provided by the depth camera. Since the robotic manipulator operates within a three-dimensional coordinate system rather than in pixel units, it is necessary to establish a transformation model that maps the "image coordinate space" to the "robot control space." It can be obtained using a linear transformation. To begin with, the valid movement range of the left palm in the image is defined by determining the minimum and maximum values in the pixel coordinate system, as described below:

$$W^p_{Lm} < x_{Tp} < W^p_{LM} \tag{30}$$

$$L^p_{Lm} < y_{Tp} < L^p_{LM} \tag{31}$$

$$H^p_{Lm} < depth < H^p_{LM} \tag{32}$$

The above inequalities define the boundary of pixel coordinate positions within the left-hand region. The corresponding boundary range for the pixel coordinates within the right-hand region is defined as

$$W^p_{Rm} < x_{Tp} < W^p_{RM} \tag{33}$$

$$L^p_{Rm} < y_{Tp} < L^p_{RM} \tag{34}$$

$$H^p_{Rm} < depth < H^p_{RM} \tag{35}$$

where x_{Tp} and y_{Tp} represent the coordinate positions in the image space, while the symbol *depth* denotes the depth value acquired from the depth camera. The parameters W^p_{Lm}, W^p_{LM} and W^p_{Rm}, W^p_{RM} signify the horizontal boundaries in the image space for the left and right hands, respectively. Similarly, L^p_{Lm}, L^p_{LM} and L^p_{Rm}, L^p_{RM} represent the vertical boundaries for the left and right hands, while H^p_{Lm}, H^p_{LM} and H^p_{Rm}, H^p_{RM} correspond to the upper and lower depth boundaries for the left and right hand regions in the image space. These boundaries in the robot's control space can be represented as

$$W^R_{Lm} < x < W^R_{LM} \tag{36}$$

$$L^R_{Lm} < y < L^R_{LM} \tag{37}$$

$$H^R_{Lm} < z < H^R_{LM} \tag{38}$$

The above inequalities define the boundary of coordinate positions in the robotic manipulator's control space corresponding to movements within the left-hand region. Likewise, the boundaries for coordinate positions within the right-hand region of the robotic manipulator's control space can be represented as

$$W^R_{Rm} < x < W^R_{RM} \tag{39}$$

$$L^R_{Rm} < y < L^R_{RM} \tag{40}$$

$$H^R_{Rm} < z < H^R_{RM} \tag{41}$$

Let W^R_{Lm}, W^R_{LM} and W^R_{Rm}, W^R_{RM} denote the movable range of the robotic manipulator along the x-axis within the control space; L^R_{Lm}, L^R_{LM} and L^R_{Rm}, L^R_{RM} denote the movable range along the y-axis; H^R_{Lm}, H^R_{LM} and H^R_{Rm}, H^R_{RM} signify the movable range along the z-axis within the control space. By mapping the positions of the user's hands in the image space to the control space of the robotic manipulators, a proportional relationship between image coordinates and the robotic control space is established. This mapping enables the end-effector of the robotic manipulator to move in accordance with the opera-

tor's hand movements. Based on this proportional relationship, the following coordinate transformation relationship can be derived as

$$x_L = (x_{Tp} - W^p_{Lm}) \times \frac{(W^R_{LM} - W^R_{Lm})}{(W^p_{LM} - W^p_{Lm})} + Comp_{Lx} \tag{42}$$

$$y_L = (y_{Tp} - L^p_{Lm}) \times \frac{(L^R_{LM} - L^R_{Lm})}{(L^p_{LM} - L^p_{Lm})} + Comp_{Ly} \tag{43}$$

where $(x_L,\ y_L)$ represents the planar coordinate point of the left-hand region. Similarly, the coordinate transformation for the right-hand region can be represented as

$$x_R = (x_{Tp} - W^p_{Rm}) \times \frac{(W^R_{RM} - W^R_{Rm})}{(W^p_{RM} - W^p_{Rm})} + Comp_{Rx} \tag{44}$$

$$y_R = (y_{Tp} - L^p_{Rm}) \times \frac{(L^R_{Rm} - L^R_{RM})}{(L^p_{RM} - L^p_{Rm})} + Comp_{Ry} \tag{45}$$

where $(x_R,\ y_R)$ signifies the coordinate point of the right-hand region. To enhance overall transformation accuracy and operational stability, compensation mechanisms are designed to address potential linear mapping errors along each axis. The compensation values for the left and right hands are listed in the following Table 3, with the adjustments separately applied to each hand. The coordinates for the left hand are represented as $Comp_{Lx}$ and $Comp_{Ly}$, while the compensation for the right hand's axes is denoted as $Comp_{Rx}$ and $Comp_{Ry}$. As shown in the equations above and based on the preceding derivation, the palm's position information in the image space can be successfully mapped to the robotic manipulator's control coordinate system using the corresponding transformation relationships.

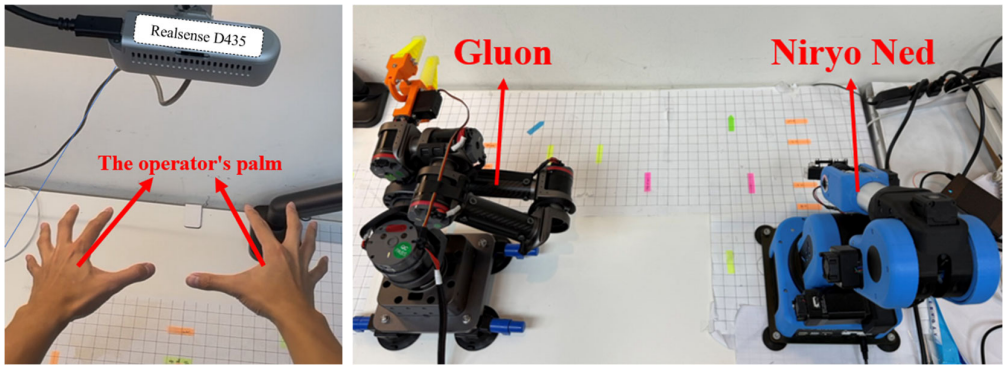
Table 3. Compensation values for the left and right hands. (Unit: mm).

	Left Hand		**Right Hand**	
Depth value	X	Y	X	Y
35~40	0.154	−0.425	0.154	0.425
40~45	0.146	−0.505	0.146	0.505
45~50	0.139	−0.575	0.139	0.575
50~55	0.133	−0.635	0.133	0.635
55~60	0.128	−0.675	0.128	0.675
60~65	0.124	−0.705	0.124	0.705
65~70	0.126	−0.725	0.126	0.725

5. Experimental Results and Discussions

Robotic manipulators manufactured by Niryo and Mintasca are used to experimentally validate the effectiveness of the proposed teleoperation control method. The Niryo Ned robotic manipulators employs a Raspberry Pi 4 as its central control board, running an operating system based on Ubuntu 18.04 and ROS Melodic to manage motion control, sensor data processing, and communication. Its motor system comprises three NiryoSteppers stepper motors and three Dynamixel XL servo motors, supporting the EasyConnect modular end-effector system and vision kit. In contrast, the Mintasca Gluon utilizes QDD Lite-NE30 Smart Compliant Actuators (SCA) for distributed control, with each actuator integrating a brushless DC motor, high-precision encoder, reducer, and driver, eliminating the need for a single central processor. In the experimental setup, a RealSense D435 depth camera shown in Figure 16a is employed to capture depth information. This depth data,

obtained through coordinate transformations, enables operators to control double robotic manipulators shown in Figure 16b enhancing precision and versatility in spatial manipulation tasks. The Gluon robotic manipulator represents the operator's left hand, while the Niryo Ned robotic manipulator represents the operator's right hand. To better mimic the movement of the operator's palms, the distance between the origins of the double robotic manipulators is set to match the natural distance between the operator's palms when the arms are extended forward.



(**a**) (**b**)

Figure 16. Experimental setup of (**a**) the depth camera and (**b**) double robotic manipulators.

Essentially, this study assesses the capability of robotic manipulators to replicate the motion trajectories of the operator's palms and examines the applicability of the proposed method across robotic manipulators with varying configurations. In the experiment, the operator uses the depth camera from a distance to capture hand gestures. The acquired data is transmitted to a computer for coordinate transformation. The transformed information is then sent to the robotic manipulator's end-effector to adjust its position. Simultaneously, the orientation of the robotic manipulator's end-effector can be adjusted based on the operator's hand orientation, enabling more precise replication of the operator's movements. The time taken is recorded in the average duration of each step of the experiment. For each step of completing the task, the average time from capturing the image to detecting the hand pose and generating coordinate points is approximately 120.04 ms. Subsequently, mapping the coordinate points to the robotic manipulator takes about 132.06 ms. The computational time for the robotic manipulator's kinematics is approximately 42.85 ms. Finally, sending the position command to the robotic manipulator and completing the motion takes about 1121.15 ms. The overall operation time is approximately 1416.1 ms.

The depth camera is positioned at a height of 70 cm above the desktop and is utilized to capture the operator's hand gesture information. This information is transmitted to a laptop for computational processing, including mean filtering and coordinate transformation. The transformed positional data is then used to calculate the joint angles of the robotic manipulator through inverse kinematics. These joint angles are input into MoveIt to control the robotic manipulator, achieving remote control of double robotic manipulators. In the experimental setup, the operator is positioned 70 cm from the desktop, where a depth camera is used to capture the operator's hand gestures. The camera's field of view defines the operational space for the operator's movements, which corresponds to the workspace of double robotic manipulators. The bases of the double robotic manipulators are positioned with an inter-base distance of 0.7 m at the origin. The Gluon robotic manipulator represents the operator's left hand gestures, while the Niryo Ned robotic manipulator represents the right hand gestures. To enhance the robotic manipulators ability to mimic the operator's movements accurately, the Gluon arm is configured in the negative y-axis plane, and the

Niryo Ned arm is configured in the positive y-axis plane. This arrangement ensures both sensitivity in the workspace and precise positional accuracy.

Figure 17 shows the time-sequence snapshots of the task is presented. Initially, the operator's left hand moves to the starting point, mimicking the action of picking up an object. The operator's right hand remains stationary, awaiting the left hand's movement. Once the left hand reaches the starting point, it proceeds to the midpoint between the double robotic manipulators to perform the object handover. After the operator's right hand receives the object, it moves to the endpoint to place the object. Figures 18 and 19 show the motion trajectories of the double robotic manipulators for this task. The left hand controls the Gluon robotic manipulator to execute the experimental trajectory, initially moving to the starting point to represent the grasping of an object. It then proceeds to the endpoint of the Gluon manipulator's task, which serves as the handover point between the left and right hands, denoted by a black square. Subsequently, the right hand controls the Niryo robotic manipulator, using the handover point as its starting point and moving to the endpoint, which represents the placement of the object. As both robotic manipulators continuously record the operator's hand gestures, even during non-active phases of the task, the starting point in Figures 18 and 19 does not appear at the beginning of the trajectory, nor does the endpoint appear at the end. The figures reveal that, prior to filtering, the motion trajectories of both robotic manipulators exhibit noises, resulting in positional inaccuracies during the handover. However, the filtered trajectories demonstrate that both robotic manipulators can smoothly reach their designated positions throughout the process, with the acceptable errors caused by noises during the handover. Furthermore, the starting and ending positions are precisely achieved, thereby validating the effectiveness and accuracy of the proposed control scheme. Additionally, this study addresses the issue of hand tremors or minor errors in sensor data acquisition during the operator's movements. The task trajectory is executed ten times, with an average duration of 36.5 s. The position of the palm after coordinate transformation is compared with the position of the robotic manipulator's end-effector for the error evaluation. The coordinates of all points in each task are averaged, and the Euclidean distance is calculated simultaneously. Additionally, the Root Mean Square Errors (RMSE), mean errors, and maximum errors are investigated, as presented in Tables 4 and 5. The larger average error observed in the Gluon robotic manipulator is attributed to measurement errors of each link. In contrast, the Niryo Ned robotic manipulator using the inverse kinematics of MoveIt demonstrates the higher position accuracy. However, these position errors are acceptable and the task is still successfully completed, enabling effective object handoff.

To improve the robotic manipulator's following ability to track the operator's palm movement speed while effectively reducing noises, a fast and efficient filter (i.e., moving average filter) is used. It can be seen in Figure 20 that the use of a smaller window size still results in a noticeable error. Conversely, when too many values are included in the average process, the palm will move an additional position to complete the averaging, leading to a delay in the robotic manipulator's motion. In addition, the noise distortion happened with 9 points in the filter of Figure 20. According to the principle, the selected every five values here are averaged. The positional data (x, y, z) of the Gluon and Niryo Ned robotic manipulators' end-effectors are, respectively, recorded in Figures 21 and 22. We can see from these figures that mean filtering indeed can effectively remove noises to make the motion trajectory smoother. Although the filtered parameters exhibit a slight delay compared to the unfiltered data, the positional information corresponding to the operator's movements is successfully achieved.

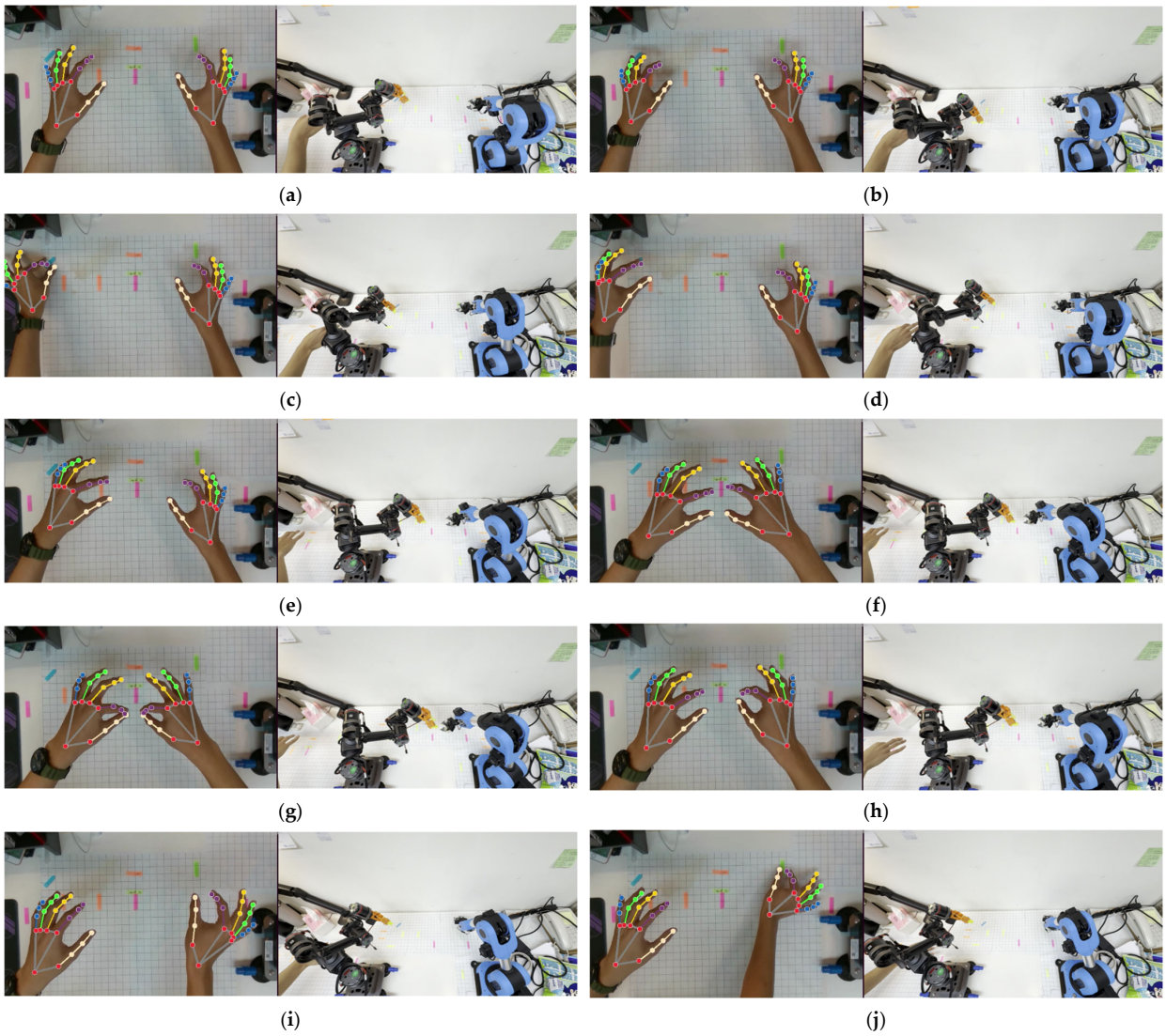


Figure 17. Snapshots of collaborative teleoperation control in the time-sequence from (**a**–**j**).

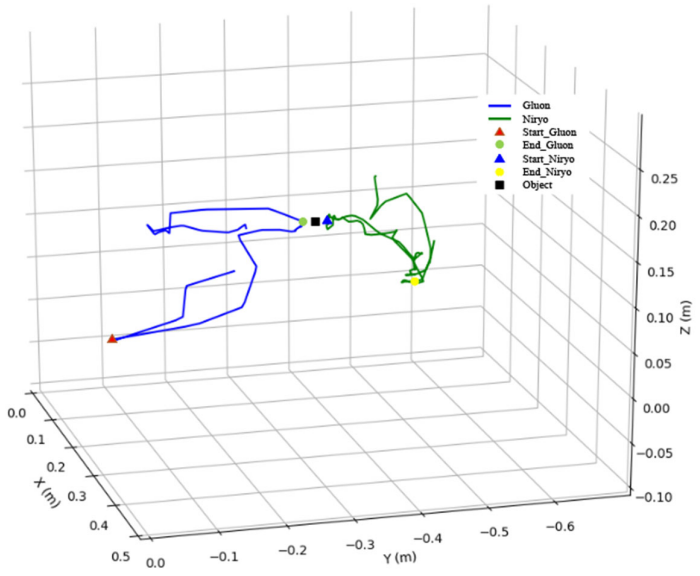


Figure 18. Three-dimensional motion trajectories of the end-effectors of the double robotic manipulators before filtering.

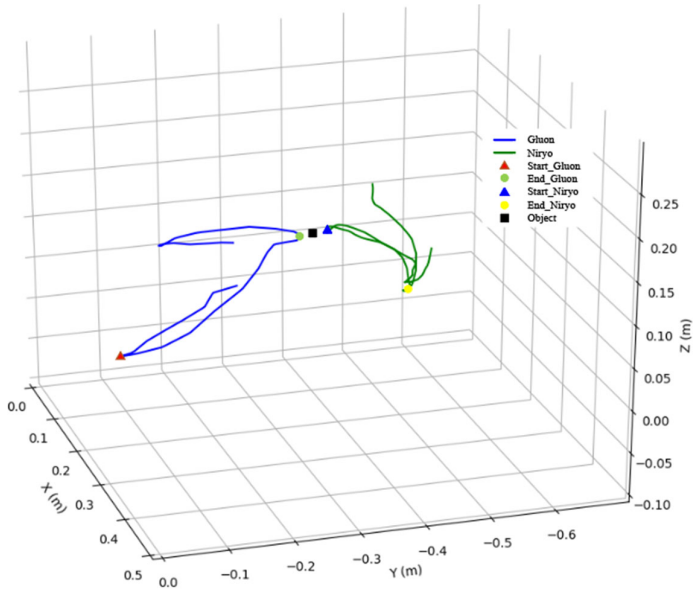


Figure 19. Three-dimensional motion trajectories of the end-effectors of the double robotic manipulators after filtering.

Table 4. Position error of the Gluon robotic manipulator's end-effector (Unit: mm).

	X	Y	Z	Euclidean Distance
1	−15.2612	30.0352	−9.6990	42.3112
2	−7.5631	1.0310	−3.0463	22.5031
3	−8.7644	13.1673	−4.0349	27.6794
4	−13.9797	21.8293	−11.4860	36.8854
5	−12.9011	30.1603	−2.9056	38.2387
6	−11.4238	32.3847	−3.2470	39.8298
7	−9.9547	15.2047	−7.6328	30.4293
8	−13.2032	20.1582	−9.4055	33.0436
9	−14.5489	27.6841	−10.5359	38.5163
10	−14.1280	26.7250	−9.9789	37.7001
RMSE	12.1	29.1	10.9	
Maxima	16.1343	33.1486	12.4763	
Mean ± SD	8.8 ± 8.4	13.8 ± 25.6	4.7 ± 9.8	33.8 ± 13.9

Table 5. Position error of the Niryo Ned robotic manipulator's end-effector. (Unit: mm).

	X	Y	Z	Euclidean Distance
1	0.6802	0.1507	−0.0029	0.5785
2	0.0755	0.1102	−0.0580	0.5185
3	0.0319	0.1108	−0.0634	0.4925
4	0.0078	0.1541	−0.0266	0.5571
5	0.0605	0.0908	−0.0697	0.5434
6	0.0305	0.0637	−0.0497	0.5180
7	0.0598	0.1276	−0.0316	0.6517
8	0.0482	0.0942	0.0134	0.5305
9	0.0117	0.0706	0.0194	0.4949
10	0.0345	0.1347	−0.0354	0.5290
RMSE	0.036	0.042	0.031	
Maxima	1.22	1.32	0.55	
Mean ± SD	0.06 ± 0.35	0.04 ± 0.41	−0.02 ± 0.31	0.507 ± 0.0299

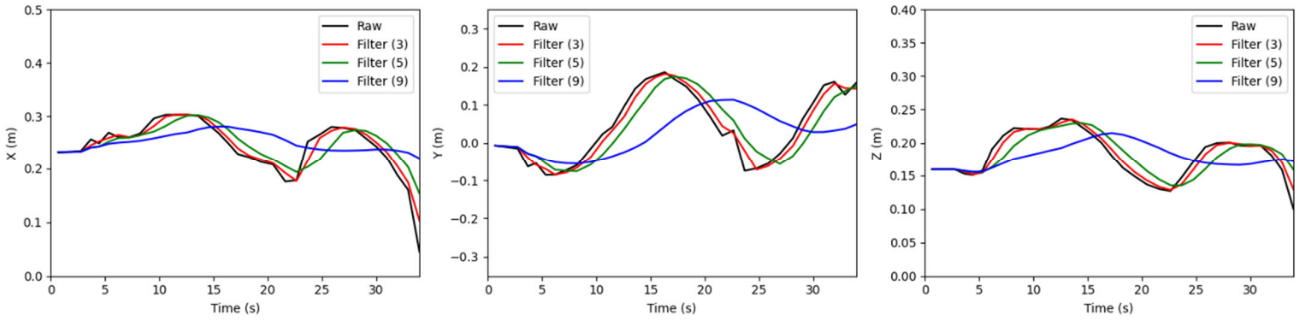
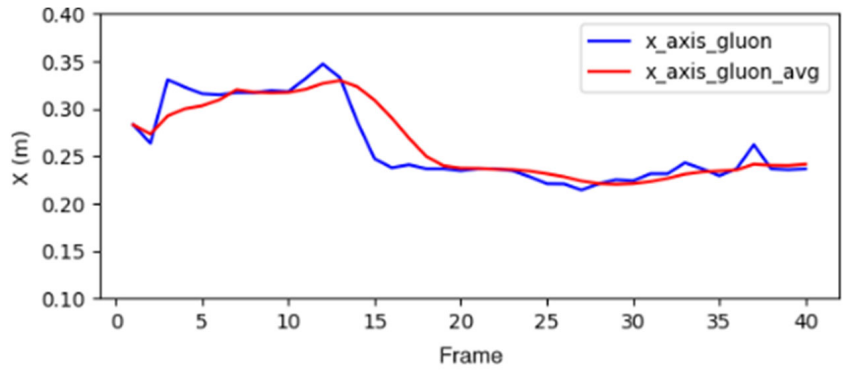
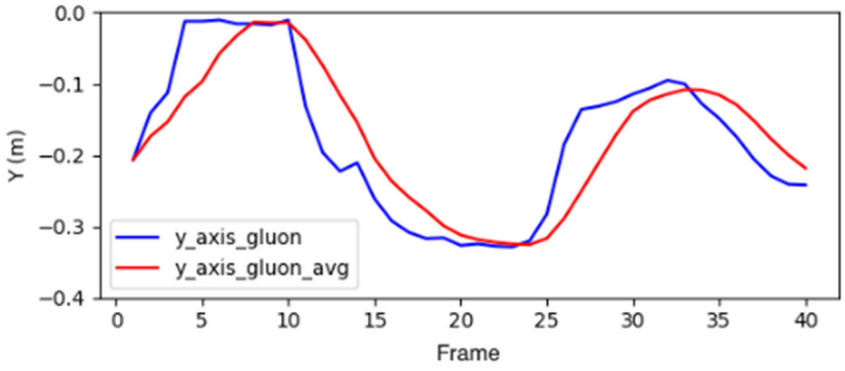


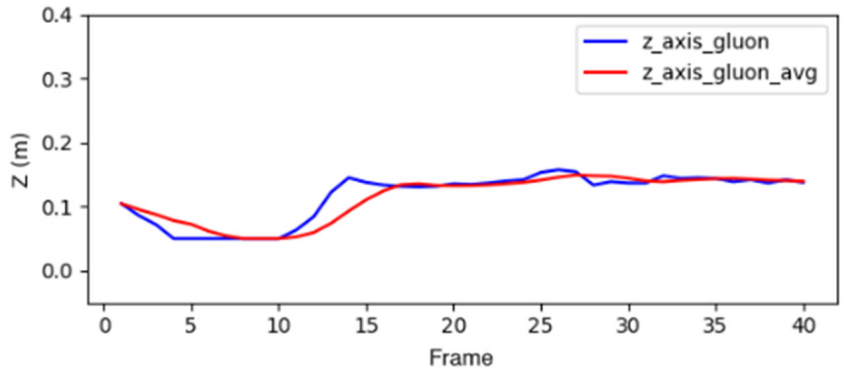
Figure 20. Trajectory responses while applying different moving-average window sizes.



(**a**)



(**b**)



(**c**)

Figure 21. Analysis of experimental motion-trajectory filtering for the Gluon robotic manipulator: (**a**) *X*-axis motion trajectory of the end-effector, (**b**) *Y*-axis motion trajectory of the end-effector, (**c**) *Z*-axis motion trajectory of the end-effector.

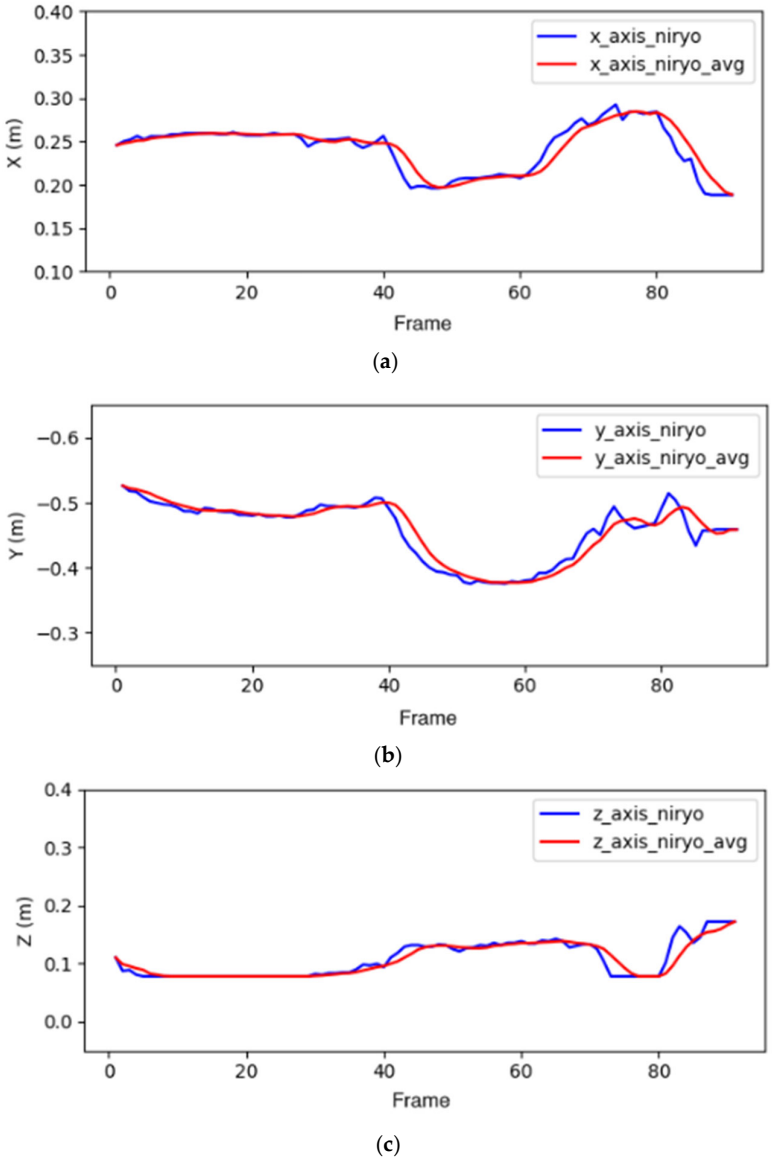


Figure 22. Analysis of experimental motion-trajectory filtering for the Niryo Ned robotic manipulator: (**a**) *X*-axis motion trajectory of the end-effector, (**b**) *Y*-axis motion trajectory of the end-effector, (**c**) *Z*-axis motion trajectory of the end-effector.

In summary, precise coordinate transformation and filtering are significant for the remote control of robotic manipulators. The experimental results demonstrate that, within the task trajectories, the robotic manipulators can accurately replicate the positions of the operator's hands and complete the object transfer task. Furthermore, although the filtered positional movements exhibit a slight delay, the target positions are still successfully achieved.

6. Conclusions

This study proposes a human–machine interaction method that employs MediaPipe hand landmark detection combined with a depth camera to track the positional information of the operator's hand. In the experiment, the result demonstrates the effectiveness of the proposed control framework. By analyzing the relationship between the thumb and index finger, remote control of the end-effectors of double robotic manipulators is achieved, enabling precise calculation of position and orientation in three-dimensional space with acceptable errors. As a result, the operator's hands can remotely control the double robotic

manipulators for collaboratively transferring the task of an object with imitation. In the future, the integration of additional sensors could extend the tracking range beyond hands, facilitating the execution of more complex tasks.

Author Contributions: Conceptualization, H.-M.W. and S.-H.W.; methodology, H.-M.W. and S.-H.W.; software, H.-M.W. and S.-H.W.; validation, H.-M.W. and S.-H.W.; formal analysis, H.-M.W. and S.-H.W.; investigation, H.-M.W. and S.-H.W.; resources, H.-M.W. and S.-H.W.; data curation, H.-M.W. and S.-H.W.; writing—original draft preparation, H.-M.W. and S.-H.W.; writing—review and editing, H.-M.W. and S.-H.W.; visualization, H.-M.W. and S.-H.W.; supervision, H.-M.W.; project administration, H.-M.W. and S.-H.W.; funding acquisition, H.-M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The relevant data is provided by a project. Due to the privacy promised in the contract, the data cannot be disclosed at this time.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Y.; Wan, G. Techniques for Selecting and Manipulating Object in Virtual Environment Based on 3-DOF Trackers and Data Glove. In Proceedings of the 16th International Conference on Artificial Reality and Telexistence—Workshops (ICAT'06), Hangzhou, China, 29 November–1 December 2006; pp. 662–665.
2. Chang, T.; Wu, Y.; Han, C.; Chang, C. Real-time Arm Motion Tracking and Hand Gesture Recognition Based on a Single Inertial Measurement Unit. In Proceedings of the 11th International Conference on Internet of Things, Malmö, Sweden, 2–5 September 2024; pp. 44–49.
3. MediaPipe. Available online: https://ai.google.dev/edge/mediapipe/solutions/guide?hl=zh-tw (accessed on 1 September 2025).
4. Sengar, S.S.; Kumar, A.; Singh, O. Efficient Human Pose Estimation: Leveraging Advanced Techniques with MediaPipe. *arXiv* **2024**, arXiv:2406.15649. [CrossRef]
5. Peta, K.; Wiśniewski, M.; Kotarski, M.; Ciszak, O. Comparison of Single-Arm and Dual-Arm Collaborative Robots in Precision Assembly. *Appl. Sci.* **2025**, *15*, 2976. [CrossRef]
6. Motoda, T.; Hanai, R.; Nakajo, R.; Murooka, M.; Erich, F.; Domae, Y. Learning Bimanual Manipulation via Action Chunking and Inter-Arm Coordination with Transformers. *arXiv* **2025**, arXiv:2503.13916. [CrossRef]
7. Wu, K.; Lu, Z.; Chen, X. A Bimanual Teleoperation System of Humanoid Robots for Dexterous Manipulation. In Proceedings of the 2024 IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, 10–14 December 2024; pp. 814–819.
8. Alsawadi, M.S.; El-kenawy, E.M.; Rio, M. Using BlazePose on Spatial Temporal Graph Convolutional Networks for Action Recognition. *Comput. Mater. Contin.* **2023**, *74*, 19–36. [CrossRef]
9. Mroz, S.; Baddour, N.; McGuirK, C.; Juneau, P.; Tu, A.; Cheng, K.; Lemaire, E. Lemaire Comparing the Quality of Human Pose Estimation with BlazePose or OpenPose. In Proceedings of the 4th International Conference on Bio-Engineering for Smart Technologies, Paris, France, 8–10 December 2021; pp. 1–4.
10. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef] [PubMed]
11. Osokin, D. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. *arXiv* **2018**, arXiv:1811.12004.
12. Gao, Z.; Chen, J.; Liu, Y.; Jin, Y.; Tian, D. A Systematic Survey on Human Pose Estimation: Upstream and Downstream Tasks, Approaches, Lightweight Models, and Prospects. *Artif. Intell. Rev.* **2025**, *58*, 68. [CrossRef]
13. RealSense D435. Available online: https://www.intelrealsense.com/depth-camera-d435/ (accessed on 1 September 2025).
14. Paterson, J.; Aldabbagh, A. Gesture-Controlled Robotic Arm Utilizing OpenCV. In Proceedings of the IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 27–30 October 2021; pp. 161–166.
15. Marić, F.; Jurin, I.; Marković, I.; Kalafatić, Z.; Petrović, I. Robot Arm Teleoperation via RGBD Sensor Palm Tracking. In Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 30 May–3 June 2016; pp. 1093–1098.
16. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.; Grundmann, M. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv* **2020**, arXiv:2006.10214.

17. Bazarevsky, V.; Martynnik, Y.; Vakunov, A.; Raveendran, K.; Grundmann, M. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs. *arXiv* **2019**, arXiv:1907.05047.
18. Grishchenko, I.; Ablavatski, A.; Kartynnik, Y.; Raveendran, K.; Grundmann, M. Attention Mesh: High-fidelity Face Mesh Prediction in Real-time. *arXiv* **2020**, arXiv:2006.10962. [CrossRef]
19. Niryo Ned. Available online: https://docs.niryo.com/robots/ned/ (accessed on 1 September 2025).
20. Gluon. Available online: https://mintasca.com/tc/product-16.html (accessed on 1 September 2025).
21. ROS. Available online: https://wiki.ros.org/ros (accessed on 1 September 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.