

# Wireless Control of Six-axis Robot Arms by Leap Motion Sensor

Chin-Chia Liu,\* Wei-Lun Hu, Chun-Kai Liang, and Chia-Chiang Hsu

Department of Industrial Education and Technology, National Changhua University of Education,  
Bao-Shan Campus, Number 2, Shi-Da Road, Changhua 500, Taiwan, R.O.C.

(Received June 2, 2022; accepted September 29, 2022)

**Keywords:** leap motion controller, noncontact control, robot arm

Most existing robot arms are controlled using wired or wearable controllers. A user must enter coordinates or control single-axis coordinate systems so that a robot arm can reach a designated destination. In this study, we developed a six-axis robot arm with a leap motion controller. The end position and coordinates of the developed robot arm changed in accordance with those of the right palm. By using the C# Web application programming interface, the coordinates of the right palm ( $X$ ,  $Y$ ,  $Z$ , pitch, yaw, and roll) were immediately uploaded to a webpage and shared in a local area network. Subsequently, the Python automatic indexing program was used to read the webpage data to construct a homogeneous matrix for the end of the robot arm. Finally, inverse kinematics was used to calculate the rotation angle of the motor of each axis. The finger bending ratio could be used to control the opening angle of the gripper of the robot arm. We measured the accuracies of the six-axis robot arm at two positions. At Position 1, the pose accuracies of the  $X$ -axis ( $AP_x$ ),  $Y$ -axis ( $AP_y$ ), and  $Z$ -axis ( $AP_z$ ) were  $-6$ ,  $0$ , and  $-9$  mm, respectively. The three-axis standard deviation of Position 1 was 2 mm, the pose accuracy ( $AP_p$ ) was 11 mm, and the pose repeatability ( $RP$ ) was 5 mm. At Position 2, the pose accuracies of the  $X$ -axis ( $AP_x$ ),  $Y$ -axis ( $AP_y$ ), and  $Z$ -axis ( $AP_z$ ) were  $10$ ,  $-7$ , and  $-12$  mm, respectively. The three-axis standard deviation of Position 2 was 2 mm, the pose accuracy ( $AP_p$ ) was 17 mm, and the pose repeatability ( $RP$ ) was 4 mm.

## 1. Introduction

One of the aims of the Fourth Industrial Revolution is to create lights-out factories, also known as dark factories, where there is zero or very low human activity on site and the factory can operate in the dark. Data are collected to train computers to analyze situations of the production line, and robot arms are used to replace humans in automatic production lines. Spray-painting tasks often involve volatile gases and dust, which may result in an explosion. Therefore, articulated robotic arms must be able to resist explosions. In addition, the required accuracy for paint spraying motion to achieve an even paint cover on the surfaces of complex workpieces is relatively low; therefore, paint spraying tasks are relatively easy for robot arms. When a controller is required for the spray painting process to protect laborers' physical health and ensure their safety, the spray painting path to be adopted will not be intuitive and overpainting

\*Corresponding author: e-mail: [ccliu@cc.ncue.edu.tw](mailto:ccliu@cc.ncue.edu.tw)  
<https://doi.org/10.18494/SAM4080>

may occur. This problem can be overcome by incorporating the hand gestures of experienced spray paint workers into robot arms. To achieve this, data must be collected from experienced spray paint workers wearing controllers on their body. Nevertheless, this creates discomfort for the workers. To reduce this discomfort and increase the intuitiveness of paint control, in this study, we integrated a leap motion controller (LMC) with a self-designed robot arm and tested the operations of the functions of this integrated system and their feasibility. A Raspberry Pi controller was coded as the robot arm controller. The LMC was connected with a desktop computer through USB 3.0. Data were transmitted through a network architecture.

Researchers have conducted extensive research on robotic manipulators on a fixed base and made many achievements in dynamic modeling.<sup>(1,2)</sup> Forward kinematics involves using the angle of the motor of each axis and matrix operations to obtain homogeneous matrix coordinates for the end position. Therefore, when using the coordinates of the homogeneous matrix of the end position to control a robot arm, inverse kinematics must be used for path planning and control mode calculations. Huang *et al.*<sup>(3)</sup> stated that robot arms with six or more degrees of freedom (DOFs) can achieve any required posture to complete tasks. The problems of singular allocation and nonlinearity may occur in inverse kinematics. Vosniakos and Kannas<sup>(4)</sup> used genetic algorithms for the inverse kinematic control of robot arms. They achieved high linear translation stability for the robot arms; however, the controller required extremely high calculation abilities. Raza *et al.*<sup>(5)</sup> carried out optimization analysis involving parametric computer-aided design modeling and finite-element analysis to design a robot arm. They designed several connection methods for robot arms and conducted a subsequent analysis. They also used Robo-Analyzer to analyze stress and strain, and they determined favorable thicknesses and areas. Park *et al.*<sup>(6)</sup> used haptic gloves to develop a tele-operated robot for remote environments. Forward kinematics of the human body was used to calculate the direction and position of each body part. By feeding this information to a humanoid's inverse kinematics, the humanoid was controlled according to the operator's position and direction.

Noncontact control sensors include LMCs and Microsoft Kinect. The Kinect sensor is used to present the human torso posture. This sensor has applications in human body interactive games and torso posture control.<sup>(7)</sup> LMCs are used in two-hand spatial sensing, virtual-reality environments, hand gesture development, surgery training, finger rehabilitation training, and hand robot control.<sup>(8)</sup> Wu *et al.*<sup>(9)</sup> integrated a depth camera and an LMC to track a virtual-reality operator's posture and detailed movements of the operator's hands. Guerra-Segura *et al.*<sup>(10)</sup> used an LMC to record writing trajectories and a classifier to distinguish authentic signatures from fake ones. The LMC was used to increase the security of signature verification. Asakawa *et al.*<sup>(11)</sup> developed tactile educational tools and used optical capture by a leap motion module to realize the dynamic capability of the tracker device. Aguilar-Lazcano and Rechy-Ramirez<sup>(12)</sup> used an LMC to develop interactive musical games that indicate the degree of recovery of patients' fingers. They used a stable environmental light source for the LMC. Moldovan and Staretu<sup>(13)</sup> adopted an LMC to control an anthropomorphic hand that could simulate a human hand's grasping state. Ahmed *et al.*<sup>(14)</sup> used an LMC to control the displacement and rotation of a robot manufactured by KUKA. Kim *et al.*<sup>(15)</sup> compared the accuracies of the Kinect sensor and an LMC by comparing the fixed robot arm displacements obtained with them.

Robot arms can perform complex tasks and increase the effectiveness of automatic production lines. Small, high-accuracy robot arms are used in minimally invasive surgery. In surgery, a device is operated to control robot arms. Forward kinematics determines the position and direction of the robot end part.<sup>(16)</sup> Hassan *et al.*<sup>(17)</sup> used a wearable electromyography signal controller to control the lifting and lowering of a robot arm capable of seven arm postures. Huang and Lan<sup>(18)</sup> used a joystick to define functions and control a robot arm. A camera was integrated into their system to establish an ultrasound model. Musić *et al.*<sup>(19)</sup> used wearable capacitor gloves to control humanoids. The distance between the index and middle fingers controlled the strength with which the two arms of the humanoid grabbed and carried heavy objects. Li *et al.*<sup>(20)</sup> used an LMC to control a robot hand so that the joint posture of the robot hand matched that of the operator's hand. In this study, we used an LMC to obtain hand features. The position of the palm was shared on a webpage. At the end of the designed six-axis robot arm, a web crawler was used to achieve a wired or wireless remote control of the position and movement of the end effector of the designed robot arm.

The rest of this paper is structured as follows. Section 2 presents an introduction to kinematics, Sect. 3 describes the system verification, and Sect. 4 presents a discussion on the obtained results and the conclusion of this study.

## 2. Robot Arm Kinematics Theory

A tandem robot arm consists of multiple joint links that form a movement chain. The number of joints determines the DOF of a robot arm. From the relationships among the joints and the defined joint parameters, the forward kinematics of the end of a robot arm can be calculated.<sup>(21,24)</sup> By contrast, end posture matrices are used to calculate the angle of each joint in inverse kinematics.

### 2.1 Forward kinematics theory

A tandem robot arm is connected using links and arm joints to establish a motion model. Figure 1 depicts the movement chain of a robot arm. Equation (1) presents the formula for calculating the forward kinematics:

$${}^0T_n = {}^0T_1(\theta_1) {}^1T_2(\theta_2) {}^2T_3(\theta_3) {}^3T_4(\theta_4) \cdots {}^{n-1}T_n(\theta_b) = \prod_{i=1}^n {}^{i-1}T_i(\theta_i), \quad (1)$$

where  $\prod_{i=1}^n {}^{i-1}T_i(\theta_i)$  denotes the homogeneous matrix of the link coordinate system.<sup>(10)</sup>

To determine the forward kinematics, the matrix product of the link between the initial coordinate system and the end coordinate system is determined. The Denavit–Hartenberg (DH) parameter table describes the motion chain of a robot arm. This table is presented in the form of a homogeneous matrix. The parameter  ${}^{i-1}T_i(\theta_i)$  converts the original coordinate system to coordinate system  $i - 1$ , which is combined with the fixed coordinate system  $i$ . The homogeneous matrix in Eq. (2) presents the product of the four parameters.

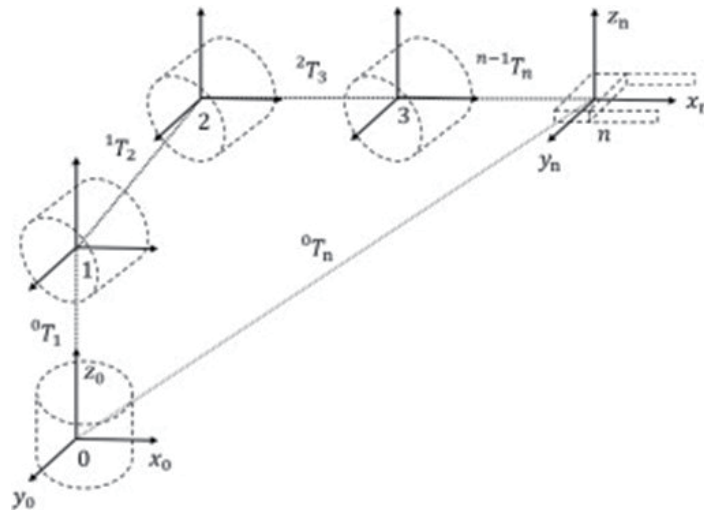


Fig. 1. Movement chain of a tandem robot arm.<sup>(8)</sup>

$${}^{i-1}T_i = T_{\dot{x}_{i-1}}(\alpha_{i-1})T_{\dot{x}R}(a_{i-1})T_{\dot{z}Q}(\theta_i)T_{\dot{z}}(d_i) = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Here,  $\theta_i$  is the rotation angle of the Z-axis,  $d_i$  is the length of the Z-axis link,  $a_i$  is the length of the X-axis link, and  $\alpha_i$  is the rotation angle of the X-axis. Sin and cos are represented by  $s$  and  $c$ , respectively, to simplify the homogeneous matrix. The homogeneous matrix is presented in Eq. (3), where  $R$  is the rotation matrix and  $P$  is the  $i$ th coordinate system expressed using vectors.

$${}^0T_n = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (3)$$

The robot arm designed in this study is displayed in Fig. 2. Table 1 lists the DH parameters of this robot arm and is used to establish the homogeneous matrix of each axis from Eqs. (4)–(10). Equation (11) presents the homogeneous matrix of the gripper. The calculations performed using this matrix are presented in Appendix 1.

$${}^0T_1 = \begin{bmatrix} c[\theta_1] & -s[\theta_1] & 0 & 0 \\ s[\theta_1] & c[\theta_1] & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

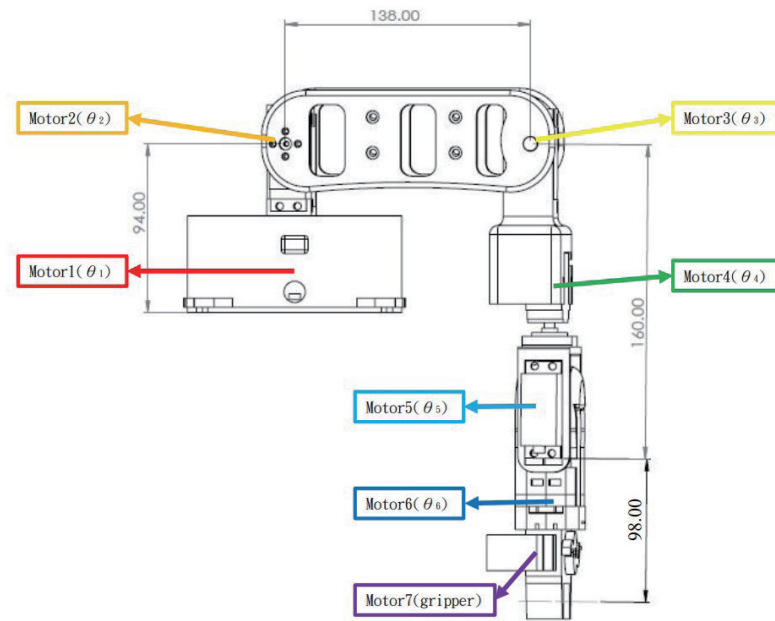


Fig. 2. (Color online) Design of the experimental robot arm.

Table 1  
DH parameters.

$i$	$a_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	$d_1$	$\theta_1$
2	90	0	0	$\theta_2$
3	0	$a_2$	0	$\theta_3$
4	90	0	$d_3$	$\theta_4$
5	-90	0	0	$\theta_5$
6	90	0	0	$\theta_6$
7	0	0	$d_7$	0

$${}^1T_2 = \begin{bmatrix} c[\theta_2] & -s[\theta_2] & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s[\theta_2] & c[\theta_2] & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^2T_3 = \begin{bmatrix} c[\theta_3] & -s[\theta_3] & 0 & a_2 \\ s[\theta_3] & c[\theta_3] & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$${}^3T_4 = \begin{bmatrix} c[\theta_4] & -s[\theta_4] & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ s[\theta_4] & c[\theta_4] & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$${}^4T_5 = \begin{bmatrix} c[\theta_5] & -s[\theta_5] & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s[\theta_5] & -c[\theta_5] & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$${}^5T_6 = \begin{bmatrix} c[\theta_6] & -s[\theta_6] & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s[\theta_6] & c[\theta_6] & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$${}^6T_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$${}^0T_7 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_7 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

## 2.2 Inverse kinematics theory

Matrices are defined using end coordinate systems. By using the algebraic, geometric, or Piper's solution method, the angle and position of each joint can be obtained. The algebraic method requires many matrix calculations and thus a high-end processor. To use Piper's solution method, the hand must meet certain conditions. Therefore, in this study, the geometric method was used as the inverse kinematics method to obtain solutions.

The DH parameter table was used to determine the inverse kinematics. The geometric method was used to calculate the angles of the first, second, and third axes. As presented in Eq. (10), the forward matrix contains critical parameters of inverse solutions. The designed six-axis robot arm comprised an elbow, an arm, and a wrist. We first defined the elbow and arm calculation positions at the centers of the fourth, fifth, and sixth axes. Because of the existence of  $d_7$ , the end matrix of the axis arm can be expressed using Eq. (12). The robot arm was projected onto the  $X$ - $Y$  plane. The angle of the first axis (Fig. 3) was calculated using Eq. (13).

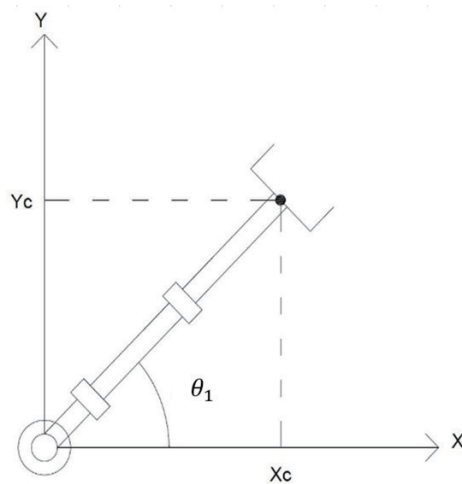


Fig. 3. Schematic of the calculation of the first axis of the inverse kinematics.

$${}^0T_6 = {}^0T_7 \cdot {}^6T_7^{-1} \quad (12)$$

$$\theta_1 = \arctan(y_c / x_c) \quad (13)$$

To obtain the angles of the second and third axes, the robot arm was projected onto the  $r$ - $Z$  plane (Fig. 4). The height of the base was subtracted from  $P_z$ , as presented in Eq. (14). Then, the plane distance  $r$  was calculated using Eq. (15).  $\theta_3$  was solved using the cosine rule [Eq. (16)]. As presented in Fig. 4 and Eqs. (17) and (18),  $\theta_3$  had two solutions, one positive and one negative. The negative value was the upper arm solution, and the positive value was the lower arm solution. Therefore, the geometric method [Eq. (19)] was used to calculate  $\theta_2$ . The upper arm solution is more suitable when the bottom area of the field contains interference, whereas the lower arm solution is more suitable when the top area of the field has a height limit or barriers.

$$v = P_z - d_1 \quad (14)$$

$$r^2 = P_x^2 + P_y^2 \quad (15)$$

$$c^2 = a^2 + b^2 - 2ab \times \cos \theta \quad (16)$$

$$\cos \theta_3 = \frac{r^2 + v^2 - a_2^2 - d_3^2}{2a_2d_3} = D \quad (17)$$

$$\theta_3 = \arctan(\pm \sqrt{1 - D^2} / D) \quad (18)$$

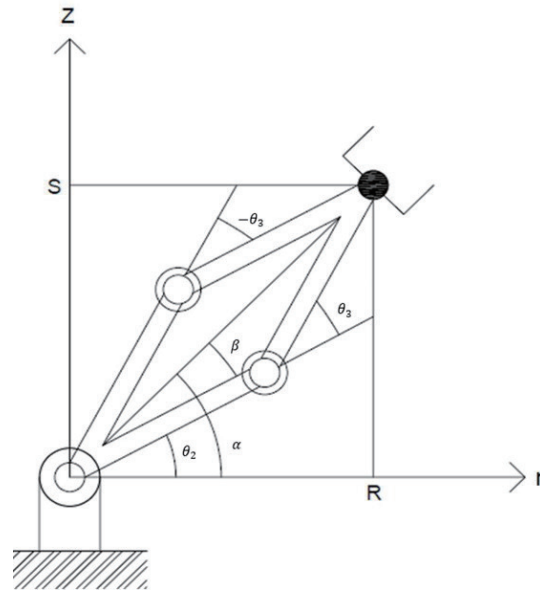


Fig. 4. Schematic of the calculation of the second and third axes of the inverse kinematics.

$$\theta_2 = \arctan(v/r) - \arctan(d_3 \sin \theta_3 / (a_2 + d_3 \cos \theta_3)) \quad (19)$$

A solution was substituted into the original forward equation to obtain  ${}^0T_3$ . From  ${}^0T_3$ ,  ${}^0R_3$  was obtained for subsequent calculations. The parameters  ${}^0R_3$  and  ${}^0R_6$  were calculated using Eq. (20). The parameters  $\theta_4$ ,  $\theta_5$ , and  $\theta_6$  were obtained using Euler angles.

$${}^3R_6 = ({}^0R_3)^T \cdot {}^0R_6 = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & C_4S_5 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 \\ -S_5S_6 & S_5S_6 & C_5 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (20)$$

By using the geometric method and Euler angles, the end matrix of each axis was obtained.<sup>(21–24)</sup>

### 3. Data Transmission and Integrated System Verification

#### 3.1 Construction of robot arm for experiment

The parameters in Table 1 are  $d_1 = 94$  mm,  $a_2 = 138$  mm,  $d_3 = 160$  mm, and  $d_7 = 98$  mm. The control system was Raspberry Pi 4. Python was used to code the forward and inverse kinematics of the control-position-type  $180^\circ$  servo motor. The servo motor's control signals were transmitted using a PCA9685 driver board (Fig. 5). Because the small servo motor only covered the range of  $0$ – $180^\circ$ , some special angles could not be achieved. This limitation did not affect the experiment.



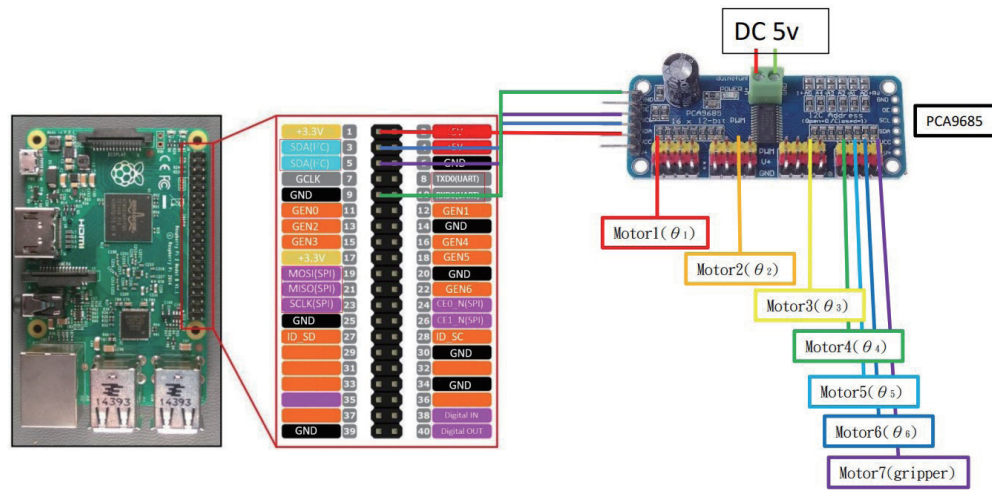


Fig. 5. (Color online) Circuit layout of the experimental robot arm.

### 3.2 Data transmission structure

The structure used to transmit experimental data is shown in Fig. 6. The red frame labeled 1 is the desktop computer, which is connected to the LMC through USB3.0 and uploaded to the local area network using the C# Web application programming interface. The red frame labeled 2 is the router, which is connected to the LMC through the cable or wireless network. The red frame labeled 3 is the robot arm system, and the final position is reached by calculating the inverse kinematics and controlling the angle of the servo motor through Raspberry Pi 4.

### 3.3 Verification of inverse kinematics programming

After coding with Python and defining the motor angle of each axis, the relevant parameters were substituted into Eq. (11) to calculate the homogeneous matrix of the end effector, which was used in the inverse kinematics geometric method to calculate the angles of each axis [Eqs. (13)–(20)]. The initial angles were set as  $\theta_1 = 0^\circ$ ,  $\theta_2 = 0^\circ$ ,  $\theta_3 = 0^\circ$ ,  $\theta_4 = 0^\circ$ ,  $\theta_5 = 0^\circ$ , and  $\theta_6 = 0^\circ$ . The homogeneous matrix of the end effector was

$${}^0T_7 = \begin{bmatrix} 1 & 0 & 0 & 138 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -164 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (21)$$

In the inverse kinematics calculations, the angle sequence of each axis was  $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$  and the solution of the angle of each axis was  $[0, 0, 0, 0, 0, 0]$ .

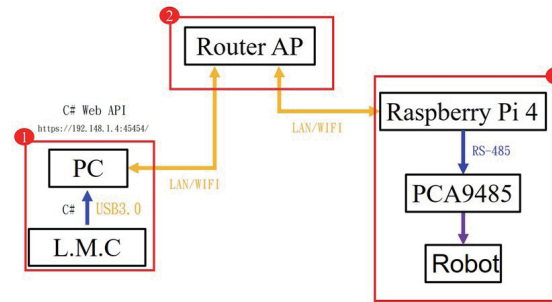


Fig. 6. (Color online) Structure used to transmit experimental data.

The position angles were set as  $\theta_1 = 50^\circ$ ,  $\theta_2 = 120^\circ$ ,  $\theta_3 = 150^\circ$ ,  $\theta_4 = 0^\circ$ ,  $\theta_5 = -50^\circ$ , and  $\theta_6 = 90^\circ$ . The homogeneous matrix of the end effector was

$${}^0T_7 = \begin{bmatrix} 0.77 & 0.49 & -0.41 & -187.69 \\ -0.64 & 0.59 & -0.49 & -223.68 \\ 0 & 0.64 & 0.77 & 288.58 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (22)$$

In the inverse kinematics calculations, the angle sequence of each axis was  $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$  and the solution of each axis was  $[50.0, 120.2, 149.9, 0, -50.4, 90]$ . The third, fourth, and fifth axes were different mainly owing to their different resolutions in the calculations.

By using a wireless network, a Python request was adopted to index the webpage XML format and edit the values on the index webpage to verify the positioning repeatability and error of the designed robot arm. The verification method involved drawing a cross at the origin. The end effector was equipped with laser light, as shown in Fig. 7. By using light directed perpendicular to the plane, the locations on the  $X$ -,  $Y$ -, and  $Z$ -axes were determined. The coordinates obtained when the robot arm moved from Position 1 to Position 2 are presented in Table 2.

The measurement process was repeated 10 times. After the robot was moved to the designated coordinates, the robot arm was stabilized and measurement was conducted. The test method of ISO9283 was used to calculate the positioning repeatability of the robot arm in the experiment. The corresponding measurement data are displayed in Fig. 8. Tables 3 and 4 present the accuracies obtained for Positions 1 and 2, respectively. The three-axis standard deviation ( $RP_i$ ) of Positions 1 and 2 was 1 mm, and the difference in their accuracies ( $AP_p$ ) was 5 mm.

### 3.4 Integrating LMC with robot arm for verification

An LMC has a small sensing space. To control a six-axis robot arm larger than an LMC, the spatial scope of the LMC must be enlarged. The system was set to allow rescaling. In this study, the verification setting position should be twice the initial LCM spatial scope. Therefore, the LMC error also doubled. By using the LMC, the six-axis robot arm, the posture in each quadrant,

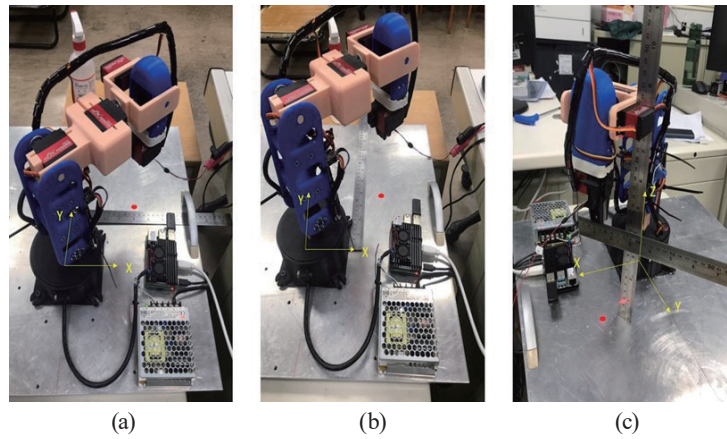


Fig. 7. (Color online) Measurement of the (a) X-, (b) Y-, and (c) Z-axis positions.

Table 2  
Settings for Positions 1 and 2.

	Coordinates
Position 1	(113, 113, 136)
Position 2	(-113, 113, 136)

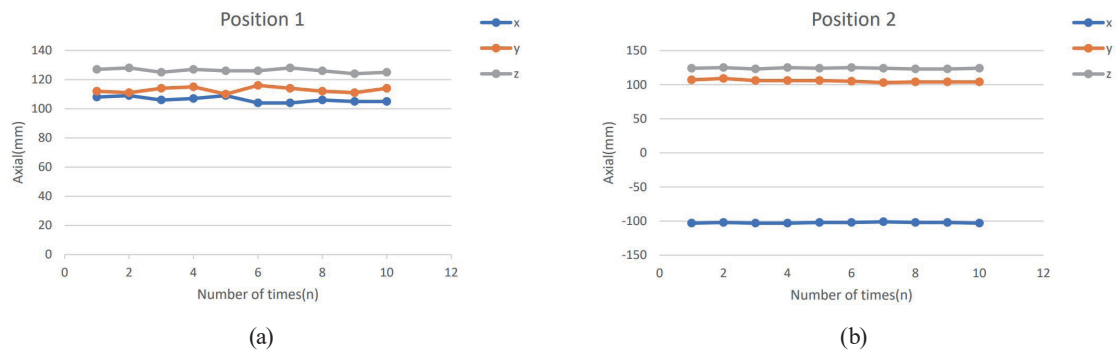


Fig. 8. (Color online) Repeated measurements at Positions (a) 1 and (b) 2.

Table 3  
Calculated accuracies for Position 1.

$\bar{x}$	106
$\bar{y}$	112
$\bar{z}$	126
$\bar{l}$	2
$RP_l$	5
$AP_x$	-6
$AP_y$	0
$AP_z$	-9
$3S_l$	2
$AP_p$	11

Table 4  
Calculated accuracies for Position 2.

$\bar{x}$	-102
$\bar{y}$	105
$\bar{z}$	124
$\bar{l}$	1
$RP_l$	4
$AP_x$	10
$AP_y$	-7
$AP_z$	-12
$3S_l$	2
$AP_p$	17

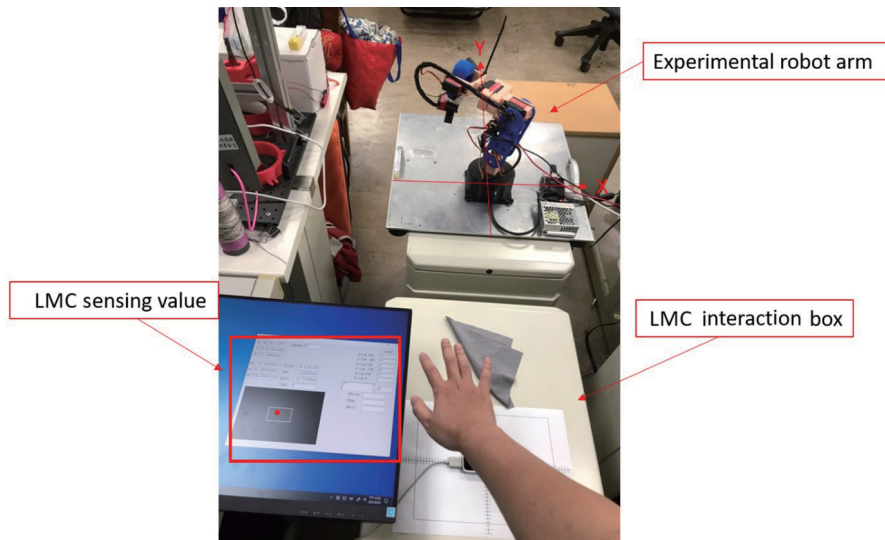


Fig. 9. (Color online) Side view of the LMC with the capability of remote wireless control.

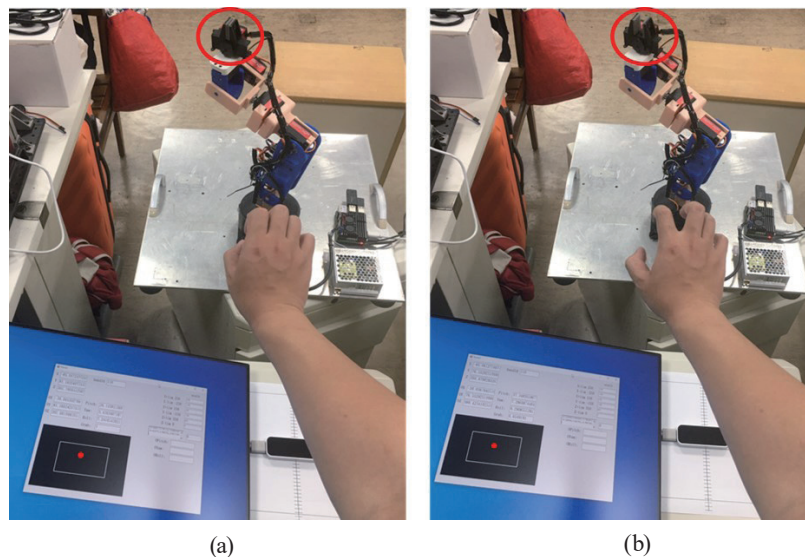


Fig. 10. (Color online) Bending with the fingers curled at (a) 100 and (b) 60%.

the wrist rotation, and the gripper opening size could be controlled. As displayed in Fig. 9, the operator's palm was located in the second quadrant of the LMC. An interface (i.e., the screen on the left) exhibited the palm position and ratio transition position information for the LMC. These data were uploaded to a webpage and the six-axis robot arm. When the fingers curled at 100%, the gripper was completely closed. When the fingers curled at 60%, the distance between the ends of the gripper was 15 mm (Fig. 10).

#### 4. Discussion and Conclusion

An LMC was used in this study to control a six-axis robot arm. However, the accuracy during the verification process was unsatisfactory. The main reasons for this result were as follows: (1)

the small servo motor did not provide feedback to the position signals and (2) a large gap existed between the motor and the deceleration mechanism. Moreover, the robot arm was controlled by open-circuit pulse width modulation; therefore, the accuracy of the signals could not be verified. The links of the robot arm were produced through 3D printing; thus, the assembly error was large. Because of the method developed for the stabilization of the second-axis motor, this motor lacked support. Additionally, the motor shaft was short. Consequently, the second-axis motor was dragged by inertia to lean forward, which resulted in unsatisfactory accuracy for the position of each axis.

The LMC was used to sense the information of the operator's right hand. This controller used a wireless network to transmit data to the remote six-axis robot arm. The position and direction of the end effector of the robot arm instantly followed those of the palm of the right hand, to translate or rotate. The opening size of the gripper was controlled using the finger curving ratio, which enabled the noncontact control of the robot arm. The position accuracy and positioning repeatability in the experiment were lower than expected because the designed robotic arm lacked accuracy. However, the forward and inverse kinematics calculation results were accurate.

The designed system can be used in dangerous environments to remotely control robot arms to complete dangerous tasks, thus ensuring operator safety. For example, the designed system can be used to search for highly reactive metals or control robot arms in spray painting. Thus, the designed system can reduce the discomfort involved in using wearable devices to control robot arms for long periods.

### Acknowledgments

We would like to express our sincere gratitude to the Ministry of Science and Technology of Taiwan (project number MOST110-2221-E-018-015) for supporting this research.

### References

- 1 T. Iwai, T. Miyazaki, T. Kawase, T. Kanno, and K. Kawashima: *Sens. Mater.* **33** (2021) 1009. <https://doi.org/10.18494/SAM.2021.3153>
- 2 S. Kaitwanidvilai, V. Chanarungruengkij, and P. Konghuayrob: *Sens. Mater.* **32** (2020) 499. <https://doi.org/10.18494/SAM.2020.2428>
- 3 H. C. Huang, C. P. Chen, and P. R. Wang: *Proc. 2012 IEEE Int. Conf. Systems, Man, and Cybernetics (SMC) (IEEE 2012)* 3105–3110.
- 4 G. C. Vosniakos and Z. Kannas: *Robot Comput. Integr. Manuf.* **25** (2009) 417. <https://doi.org/10.1016/j.rcim.2008.02.005>
- 5 K. Raza, T. A. Khan, and N. Abbas: *J. King Saud Univ., Eng. Sci.* **30** (2018) 218. <https://doi.org/10.1016/j.jksues.2018.03.005>
- 6 S. Park, Y. Jung, and J. Bae: *Mechatronics* **55** (2018). 54. <https://doi.org/10.1016/j.mechatronics.2018.08.011>
- 7 C. P. Quintero, R. T. Fomena, A. Shademan, N. Wolleb, T. Dick, and M. Jagersand: *Proc. 2013 IEEE Int. Conf. Robotics and Automation (IEEE 2013)* 1166–1171.
- 8 S. A. Jadhav, R. R. Yashod, A. P. Sabne, Y. K. Rokde, and C. R. Ghuge: *IJCSMC* **9** (2020) 50. <https://www.ijcsmc.com/docs/papers/December2020/V9I12202012.pdf>
- 9 Y. Wu, Y. Wang, S. Jung, S. Hoermann, and R. W. Lindeman: *Entertain Comput.* **31** (2019) 100303. <https://doi.org/10.1016/j.entcom.2019.100303>
- 10 E. Guerra-Segura, A. Ortega-Pérez, and C. M. Travieso: *Expert Syst. Appl.* **165** (2021) 113797. <https://doi.org/10.1016/j.eswa.2020.113797>

- 11 N. Asakawa, H. Wada, Y. Shimomura, and K. Takasugi: *Sens. Mater.* **32** (2020) 3617. <https://doi.org/10.18494/SAM.2020.2939>
- 12 C. A. Aguilar-Lazcano and E. J. Rechy-Ramirez: *Measurement* **157** (2020) 107677. <https://doi.org/10.1016/j.measurement.2020.107677>
- 13 C. C. Moldovan and I. Staretu: *Procedia Eng.* **181** (2017) 575. <https://doi.org/10.1016/j.proeng.2017.02.436>
- 14 S. Ahmed, V. Popov, A. Topalov, and N. Shakev: *IFAC-PapersOnLine* **52** (2019) 321. <https://doi.org/10.1016/j.ifacol.2019.12.543>
- 15 Y. Kim, P. C. Kim, R. Selle, A. Shademan, and A. Krieger: *Proc. 2014 IEEE Int. Conf. Robotics and Automation (ICRA) (IEEE 2014)* 3502–3509.
- 16 M. Uddin, V. Kumar, and V. K. Yadav: *Mater. Today* **47** (2021) 3761. <https://doi.org/10.1016/j.matpr.2021.02.429>
- 17 H. F. Hassan, S. J. Abou-Loukh, and I. K. Ibraheem: *J. King Saud Univ., Eng. Sci.* **32** (2020) 378. <https://doi.org/10.1016/j.jksues.2019.05.001>
- 18 Q. Huang and J. Lan: *Biomed. Signal Process. Control.* **54** (2019) 101606. <https://doi.org/10.1016/j.bspc.2019.101606>
- 19 S. Musić, G. Salvietti, F. Chinello, D. Prattichizzo, and S. Hirche: *IEEE T. Haptics* **12** (2019) 350. <https://hal.archives-ouvertes.fr/hal-02360685>
- 20 C. Li, A. Fahmy, and J. Sienz: *IEEE Access* **7** (2019) 136914. <https://ieeexplore.ieee.org/abstract/document/8845589>
- 21 M. W. Spong, S. Hutchinson, and M. Vidyasagar: *Robot Modeling and Control* (Wiley, New York, 2006).
- 22 R. Kelly, V. S. Davila, and J. A. L. Perez: *Control of Robot Manipulators in Joint Space* (Springer Science & Business Media, Berlin, 2005).
- 23 M. W. Spong and M. Vidyasagar: *Robot Dynamics and Control* (John Wiley & Sons, New Yor, 2008).
- 24 L. W. Lee, L. Y. Lu, I. H. Li, C. W. Lee, and T. J. Su: *Sens. Mater.* **33** (2021) 3081. <https://doi.org/10.18494/SAM.2021.3250>

## Appendix

$$\begin{aligned}
 n_x &= s[\theta_1] \left( c[\theta_5] c[\theta_6] s[\theta_4] + c[\theta_4] s[\theta_6] \right) \\
 &\quad + c[\theta_1] \left( -c[\theta_6] s[\theta_2 + \theta_3] s[\theta_5] + c[\theta_2 + \theta_3] \left( c[\theta_4] c[\theta_5] c[\theta_6] - s[\theta_4] s[\theta_6] \right) \right) \\
 n_y &= c[\theta_6] \left( c[\theta_5] \left( c[\theta_2] c[\theta_3] c[\theta_4] s[\theta_1] - c[\theta_4] s[\theta_1] s[\theta_2] s[\theta_3] - c[\theta_1] s[\theta_4] \right) - s[\theta_1] s[\theta_2 + \theta_3] s[\theta_5] \right) \\
 &\quad - \left( c[\theta_1] c[\theta_4] + c[\theta_2 + \theta_3] s[\theta_1] s[\theta_4] \right) s[\theta_6] \\
 n_z &= c[\theta_6] \left( c[\theta_3] \left( c[\theta_4] c[\theta_5] s[\theta_2] + c[\theta_2] s[\theta_5] \right) + s[\theta_3] \left( c[\theta_2] c[\theta_4] c[\theta_5] - s[\theta_2] s[\theta_5] \right) \right) \\
 &\quad - s[\theta_2 + \theta_3] s[\theta_4] s[\theta_6] \\
 o_x &= c[\theta_4] c[\theta_6] s[\theta_1] - c[\theta_1] c[\theta_2 + \theta_3] c[\theta_6] s[\theta_4] \\
 &\quad - c[\theta_5] \left( c[\theta_1] c[\theta_2 + \theta_3] c[\theta_4] + s[\theta_1] s[\theta_4] \right) s[\theta_6] + c[\theta_1] s[\theta_2 + \theta_3] s[\theta_5] s[\theta_6] \\
 o_y &= c[\theta_1] \left( -c[\theta_4] c[\theta_6] + c[\theta_5] s[\theta_4] s[\theta_6] \right) \\
 &\quad + s[\theta_1] \left( -c[\theta_2 + \theta_3] c[\theta_6] s[\theta_4] \right) \\
 &\quad + \left( -c[\theta_2] c[\theta_3] c[\theta_4] c[\theta_5] + c[\theta_4] c[\theta_5] s[\theta_2] s[\theta_3] + s[\theta_2 + \theta_3] s[\theta_5] \right) s[\theta_6] \\
 o_z &= -c[\theta_6] s[\theta_2 + \theta_3] s[\theta_4] - \left( c[\theta_4] c[\theta_5] s[\theta_2 + \theta_3] + c[\theta_2 + \theta_3] s[\theta_5] \right) s[\theta_6] \\
 a_x &= s[\theta_1] s[\theta_4] s[\theta_5] + c[\theta_1] \left( c[\theta_5] s[\theta_2 + \theta_3] + c[\theta_2 + \theta_3] c[\theta_4] s[\theta_5] \right)
 \end{aligned}$$

$$a_y = c[\theta_5]s[\theta_1]s[\theta_2 + \theta_3] + (c[\theta_2]c[\theta_3]c[\theta_4]s[\theta_1] - c[\theta_4]s[\theta_1]s[\theta_2]s[\theta_3] - c[\theta_1]s[\theta_4]])s[\theta_5]$$

$$a_z = -c[\theta_2 + \theta_3]c[\theta_5] + c[\theta_4]s[\theta_2 + \theta_3]s[\theta_5]$$

$$p_x = c[\theta_1](a_2c[\theta_2] + d_3s[\theta_2 + \theta_3])$$

$$p_y = s[\theta_1](a_2c[\theta_2] + d_3s[\theta_2 + \theta_3])$$

$$p_z = d_1 - d_3c[\theta_2 + \theta_3] + (a_2s[\theta_2])$$