Article

User Local Coordinate-Based Accompanying Robot for Human Natural Movement of Daily Life

Hsiao-Kuan Wu [1,2], **Po-Yin Chen** [1,3,4], **Hong-Yi Wu** [1] and **Chung-Huang Yu** [1,*]

1. Department of Physical Therapy and Assistive Technology, National Yang Ming Chiao Tung University, Taipei 11221, Taiwan; shogunwu@ym.edu.tw (H.-K.W.); pychen1206@gm.ym.edu.tw (P.-Y.C.); hongyiwu@ym.edu.tw (H.-Y.W.)
2. Center for General Education, National Yang Ming Chiao Tung University, Taipei 11221, Taiwan
3. Department of Physical Medicine and Rehabilitation, Taipei Veterans General Hospital, Taipei 11217, Taiwan
4. School of Medicine, National Yang-Ming University, Taipei 11221, Taiwan
* Correspondence: chyu@ym.edu.tw

Abstract: Considering the trend of aging societies, accompanying technology can help frail, elderly individuals participate in daily activities. The ideal accompanying robot should accompany the user in a proper position according to the activity scenarios and context; the prerequisite is that the accompanying robot should quickly move to a designated position and closely maintain it regardless of the direction in which the user moves. This paper proposes a user local coordinate-based strategy to satisfy this need. As a proof of concept, a novel "string-pot" approach was utilized to measure the position difference between the robot and the target. We implemented the control strategy and assessed its performance in our gait lab. The results showed that the robot can follow the user in the designated position while the user performs forward, backward, and lateral movements, turning, and walking along a curve.

Keywords: accompanying robot; assistive technology; autonomous following in front

1. Introduction

Many accompanying robots have been developed to help people perform predefined marine, overground, and aeronautical tasks [1–20]. Considering the trend of aging societies, we aim to extend the accompanying technology to help frail, elderly individuals participate in daily activities.

To reduce the physical burden on elderly people, shopping carts, e.g., those proposed in [2], that can automatically follow behind the user have been developed. Other devices such as smart luggage [3,4] and golf carts [5] are available on the market to facilitate certain activities. However, the "following-behind" approach sometimes imposes a psychosocial impact on users, because they cannot see the device and must frequently check whether the device is lost [6,10,13].

To solve this psychosocial problem, researchers have begun to develop robots that are located in front of or to the side of the person while accompanying him or her [6–10]. Front-accompanying robots enable the user to check the robot's location and access the robot with ease. However, an accompanying robot that is always situated in front would interfere with some of the user's activities, such as fetching items from a shelf in a supermarket, opening doors, and interacting with friends. Therefore, a robot that can accompany the user in different positions is desirable for daily living activities.

In previous studies [2,7–9,14,18–21] on autonomous following, there was limited discussion on the user's sideways walking or sharp turning [6,22], which are essential for daily activities. Moreover, it appears that when the user turns, the robot does not move around the user; instead, the user moves around the robot. The reasons include the limited degrees of freedom (DOFs) of the moving platform and the user's movement

tracking strategy. As Hu [7] noted, a user implicitly smoothly moves to cooperate with the accompanying robot movement. In other words, a user reduces the accustomed moving sideways or sharp turns to prevent the robot from failing to follow the user, which also reduces the usability of the accompanying robot in daily activities.

The ideal goal of an accompanying robot for daily living is that the robot can accompany the user in a proper position according to the activity scenarios and context and that the user can move naturally without worry. To achieve this goal, there are two levels of tasks. First, the position at which the robot should accompany the user must be determined. This step can be accomplished by explicit commands from the user [11,14], and the robot can be helped by other techniques such as obstacle avoidance, path planning [10,12,13], or artificial intelligence [15–17] in the future. The second and fundamental task is that the robot should quickly move to the designated position and closely maintain it regardless of the direction in which the user moves. This second task is the focus of this study.

The first prerequisite to closely following the user's movements is that the robot must be as agile as the human, i.e., it should at least be able to move on a plane with 3 DOFs. Many designs, e.g., Mecanum wheels [23], have been proposed and are outside the scope of this study. The other important issue in closely following the user is the tracking method. For example, if the robot tracks the course of the user's movement, it may not be able to determine whether the user is moving backward or turning around and subsequently moving forward. To date, a single method for a robot to follow a user in any designated position has not been developed. This paper proposes a user local coordinate-based method to satisfy this demand. To prove its concept, we implemented a system and assessed its performance in our gait lab.

2. Methods

2.1. Concept of the User Local Coordinate-Based Accompanying Method

The concept of the proposed accompanying method is that the robot should move to the specified position with respect to the user's local coordinate system (LCS_U). As shown in Figure 1a, the objective is to make $\vec{R_C}$ (the current position and orientation of the robot relative to the user) approach $\vec{R_T}$ (the target position and orientation relative to the user), i.e., to make the position difference $\Delta\vec{R_M}$ approach the zero vector as given in Equation (1). Its underlying tasks include determining the target position, obtaining the robot's position, calculating the position difference between the target and the accompanying robot in LCS_U, and moving the robot in the robot coordinate system (LCS_R) to reduce the position difference. By minimizing the position difference, the robot can closely accompany the user in a designated position while the user moves naturally.

The pseudocode of this algorithm is listed as follows:

1. System Initialize
2. Setup and reset user's coordinate system
3. Setup and reset robot's coordinate system
4. Set target position P_t of the accompanying robot w.r.t the user
5. Measure the position P_c of the accompanying robot w.r.t the user
6. Calculate the errors E_u between P_c and P_t
7. Convert E_u to E_r in the accompanying robot's local coordinate system
8. Move the robot to reduce the error E_r according to a control algorithm
9. Repeat steps 4~8 until system stops following

Because both local coordinate systems of the user (O_U) and robot (O_R) are moving and rotating, the coordinate transformation of the position difference between LCS_U and LCS_R must be clarified. In this study, it was assumed that the target position was predetermined; thus, the required position difference rested on determining the current position of the robot in LCS_U. The transformation equation is derived as follows:

$$\vec{R_T} = O_U\vec{O}_{RT} = \begin{bmatrix} X_T & Y_T & 1 \end{bmatrix} \quad \vec{R_C} = O_U\vec{O}_{RC} = \begin{bmatrix} X_C & Y_C & 1 \end{bmatrix}$$

Objective: move O_{RC} to O_{RT}, i.e.,

$$\Delta \vec{R_M} = \vec{R_T} - \vec{R_C} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where

$$O_w : World\ coordinate\ system\ (WCS)$$
$$O_U : User's\ local\ coordinate\ system\ (LCS_U)$$
$$O_R : Robot's\ local\ coordinate\ system\ (LCS_R)$$
$$\vec{R_T} : Robot's\ target\ position\ relative\ to\ the\ user$$
$$\vec{R_C} : Robot's\ current\ position\ relative\ to\ the\ user$$
$$\Delta \vec{R_M} : Position\ difference$$

For convenience, homogeneous coordinates are used. In general, the user is moving, so LCS_U and $\vec{R_C}$ in LCS_U will change even when the robot is stationary. The three moving scenarios are discussed below.
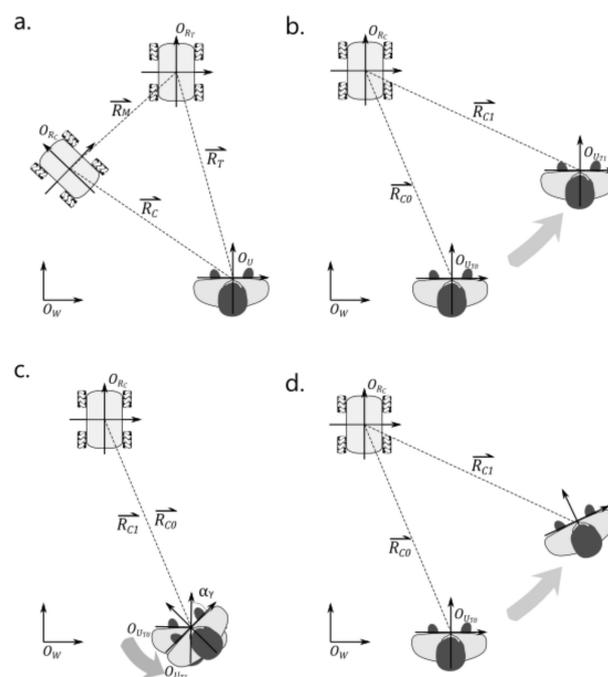
Figure 1. The movements of the accompanying robot and the user. (**a**) the robot translates and rotates to the target position; (**b**) the user translates without rotation; (**c**) the user rotates in place without translation; (**d**) the user translates and rotates.

The first scenario is that the user performs a translation without rotation, as shown in Figure 1b. Assuming the user translates from $O_{U_{T0}}$ to $O_{U_{T1}}$, let $\overrightarrow{O_{U_{T0}}O_{U_{T1}}} = \begin{bmatrix} \Delta X_t & \Delta Y_t \end{bmatrix}$ in the world coordinate system (*WCS*), and view the movement from the perspective of LCS_U.

$$\vec{R_{C1}} = \overrightarrow{O_{U_{T1}}O_{R_{C0}}} = \vec{R_{C0}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\Delta X_t & -\Delta Y_t & 1 \end{bmatrix}$$
$$= \begin{bmatrix} X_{C0} \\ Y_{C0} \\ 1 \end{bmatrix}^T - \begin{bmatrix} \Delta X_t \\ \Delta Y_t \\ 1 \end{bmatrix}^T \tag{2}$$

The second scenario is that the user performs a rotation in place without translation, as shown in Figure 1c. $\overrightarrow{O_{U_{T0}}O_{U_{T1}}}$ rotates with α_r in WCS and is viewed from LCS_U.

$$
\begin{aligned}
\overrightarrow{R_{C1}} &= \overrightarrow{O_{U_{T1}}O_{R_{C1}}} = \overrightarrow{O_{U_{T0}}O_{R_{C0}}} \text{ rotate } -\alpha_r \\
&= \overrightarrow{R_{C0}} \begin{bmatrix} \cos(-\alpha_r) & \sin(-\alpha_r) & 0 \\ -\sin(-\alpha_r) & \cos(-\alpha_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} X_{C0} \\ Y_{C0} \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(-\alpha_r) & \sin(-\alpha_r) & 0 \\ -\sin(-\alpha_r) & \cos(-\alpha_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} X_{C0}\cos\alpha_r + Y_{C0}\sin\alpha_r \\ -X_{C0}\sin\alpha_r + Y_{C0}\cos\alpha_r \\ 1 \end{bmatrix}^T
\end{aligned}
\tag{3}
$$

LCS_R is also rotated by α_r.

The third scenario is that the user translates and rotates, as shown in Figure 1d; then,

$$
\begin{aligned}
\overrightarrow{R_{C1}} &= \overrightarrow{R_{C0}} \begin{bmatrix} \cos(-\alpha_r) & \sin(-\alpha_r) & 0 \\ -\sin(-\alpha_r) & \cos(-\alpha_r) & 0 \\ -\Delta X_t & -\Delta Y_t & 1 \end{bmatrix} \\
&= \begin{bmatrix} X_{C0} \\ Y_{C0} \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(-\alpha_r) & \sin(-\alpha_r) & 0 \\ -\sin(-\alpha_r) & \cos(-\alpha_r) & 0 \\ -\Delta X_t & -\Delta Y_t & 1 \end{bmatrix} \\
&= \begin{bmatrix} X_{C0} \\ Y_{C0} \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(-\alpha_r) & \sin(-\alpha_r) & 0 \\ -\sin(-\alpha_r) & \cos(-\alpha_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \Delta X_t \\ \Delta Y_t \\ 1 \end{bmatrix}^T
\end{aligned}
\tag{4}
$$

The third scenario is the general case. Combined with Equation (1), this leads to $\overrightarrow{\Delta R_{MU}}$, and the position difference in LCS_U is

$$
\begin{aligned}
\overrightarrow{\Delta R_{MU}} = \begin{bmatrix} \Delta X_U \\ \Delta Y_U \\ 1 \end{bmatrix}^T &= \overrightarrow{R_T} - \overrightarrow{R_{C1}} \\
&= \begin{bmatrix} X_T \\ Y_T \\ 1 \end{bmatrix}^T - \left(\begin{bmatrix} X_{C0} \\ Y_{C0} \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(-\alpha_r) & \sin(-\alpha_r) & 0 \\ -\sin(-\alpha_r) & \cos(-\alpha_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \Delta X_t \\ \Delta Y_t \\ 1 \end{bmatrix}^T \right)
\end{aligned}
\tag{5}
$$

LCS_R is also rotated by α_r.

In implementation, the robot moves in LCS_R; therefore, $\overrightarrow{\Delta R_{MU}}$ should further rotate with (γ_{C0}) and yield $\overrightarrow{\Delta R_{MR}}$ as the position difference in LCS_R.

$$
\begin{aligned}
\overrightarrow{\Delta R_{MR}} = \begin{bmatrix} \Delta X_R \\ \Delta Y_R \\ 1 \end{bmatrix}^T &= \begin{bmatrix} \Delta X_U \\ \Delta Y_U \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(-\gamma_{C0}) & \sin(-\gamma_{C0}) & 0 \\ -\sin(-\gamma_{C0}) & \cos(-\gamma_{C0}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} X_T \\ Y_T \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(\gamma_{C0}) & -\sin(\gamma_{C0}) & 0 \\ \sin(\gamma_{C0}) & \cos(\gamma_{C0}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&\quad - \begin{bmatrix} X_{C0} \\ Y_{C0} \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(\alpha_r + \gamma_{C0}) & -\sin(\alpha_r + \gamma_{C0}) & 0 \\ \sin(\alpha_r + \gamma_{C0}) & \cos(\alpha_r + \gamma_{C0}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&\quad + \begin{bmatrix} \Delta X_t \\ \Delta Y_t \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(\alpha_r + \gamma_{C0}) & -\sin(\alpha_r + \gamma_{C0}) & 0 \\ \sin(\alpha_r + \gamma_{C0}) & \cos(\alpha_r + \gamma_{C0}) & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{6}
$$

With the error between the robot's position and the target position, i.e., Equation (6), and by applying a control law, e.g., PID, to make $\overrightarrow{R_{MR}} \to \overrightarrow{0}$, the robot can accompany

the user in any position, including in front, and in any movement direction, including forward, backward, sideways, rotating in place, etc. To maintain the orientation of the robot in LCS_U, the robot should also rotate by α_r.

2.2. Embodiment of the Accompanying Robot with the LCS_U Viewpoint

To measure the position difference between the robot and the target, a novel "string-pot" approach was utilized. It is composed of two rotary encoders, one distance sensor, and a retractable conducting wire, as shown in Figure 2. The shafts of the two rotary encoders are parallel to each other and fixed on the user and the robot. The distance sensor is attached to the housing of the robot's encoder. The retractable wire connects the housing of the encoders, pulls them to face each other, and ensures that the distance sensor always points to the user. In addition, the retractable conducting wire can transmit the encoder's signal from the user to the robot. This arrangement can easily measure the orientations of LCS_U and LCS_R with respect to each other and the distance between them. With this measurement system, $\overrightarrow{R_{c1}}$ is directly measured w.r.t. LCS_U, and Equations (4) and (6) become Equations (7) and (8), respectively.

$$\overrightarrow{R_{C1}} = \begin{bmatrix} d \cdot \cos(-\theta_U) \\ d \cdot \sin(-\theta_U) \\ 1 \end{bmatrix}^T \tag{7}$$

$$\begin{aligned} \overrightarrow{\Delta R_{MR}} &= \begin{bmatrix} \Delta X_R \\ \Delta Y_R \\ 1 \end{bmatrix}^T = \begin{bmatrix} \Delta X_U \\ \Delta Y_U \\ 1 \end{bmatrix}^T \begin{bmatrix} \cos(\theta_U + \theta_R) & \sin(\theta_U + \theta_R) & 0 \\ -\sin(\theta_U + \theta_R) & \cos(\theta_U + \theta_R) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \left(\begin{bmatrix} X_T \\ Y_T \\ 1 \end{bmatrix}^T - \begin{bmatrix} d \cdot \cos(-\theta_U) \\ d \cdot \sin(-\theta_U) \\ 1 \end{bmatrix}^T \right) \begin{bmatrix} \cos(\theta_U + \theta_R) & \sin(\theta_U + \theta_R) & 0 \\ -\sin(\theta_U + \theta_R) & \cos(\theta_U + \theta_R) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{8}$$

where

$$d : \text{ distance between the user and the robot}$$
$$\theta_U : \text{ encoder value on the user}$$
$$\theta_R : \text{ encoder value on the robot}$$

The measured encoder values are negative for the user orientation change; i.e.,

$$\overrightarrow{\Delta R_{MR}}\theta = \theta_U + \theta_R$$

where

$\overrightarrow{\Delta R_{MR}}\theta$ is the rotation angle of LCS_R to maintain the orientation w.r.t. LCS_U

As mentioned, while performing daily activities, humans utilize all 3 DOFs, including moving forward, backward, and sideways and turning in place. To simplify the movement control, we utilized Mecanum wheels [23] for the 3-DOF moving platform.

The control system structure is shown in Figure 3. A PID controller was implemented, and $\overrightarrow{\Delta R_{MR}}$ was used as its error input. The output of the PID controller was the velocity command $\begin{bmatrix} V_x & V_y & \omega_z \end{bmatrix}$ for the robot to execute, as shown in Equation (9).

$$\overrightarrow{\Delta R_{MR}} + \overrightarrow{\Delta R_{MR}}\theta = \begin{bmatrix} \Delta x_R \\ \Delta y_R \\ \Delta \theta_R \end{bmatrix}^T \xrightarrow{\text{PID}} \overrightarrow{U} = \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix}^T \tag{9}$$

Since the four Mecanum wheels were used, the individual rotational speeds of the wheels could be calculated for the intended movement, and the pulse-width modulation (PWM) signals could be converted for their motor drivers using the pulse-width (PW) mapper as shown in Equation (10) [24].

$$\text{PW mapper}: \begin{bmatrix} PWM_1 \\ PWM_2 \\ PWM_3 \\ PWM_4 \end{bmatrix} = k \begin{bmatrix} \omega_1 \\ -\omega_2 \\ \omega_3 \\ -\omega_4 \end{bmatrix} = \frac{k}{R} \begin{bmatrix} +1 & +1 & (L_1+L_2) \\ +1 & -1 & (L_1+L_2) \\ -1 & +1 & (L_1+L_2) \\ -1 & -1 & (L_1+L_2) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_z \end{bmatrix} \tag{10}$$

where

V_x, V_y, ω_z: velocity of the robot
ω_1, ω_2, ω_3, ω_4: angular velocity of the four Mecanum wheels
R: radius of the Mecanum wheels
k: motor constant
PWM: pulse width modulation
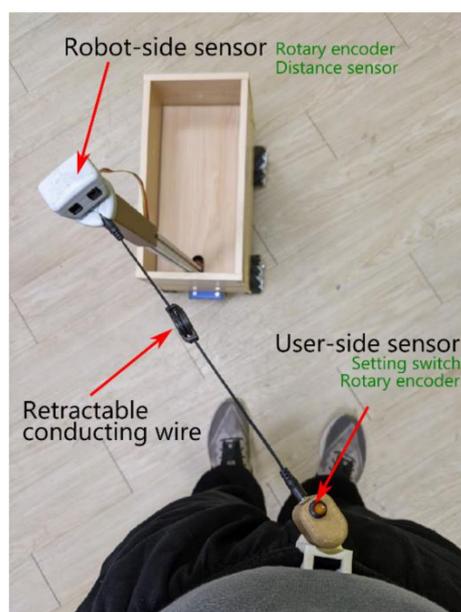
Figure 2. "String-pots" to sense LCS_U and LCS_R.
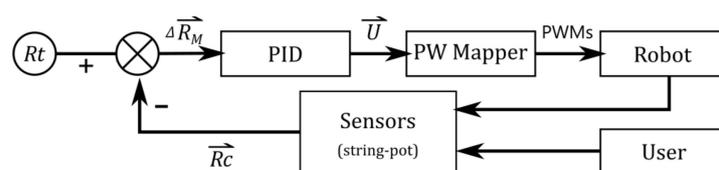
Figure 3. Control system structure.

2.3. Detailed System Hardware

The string-pot system in Figure 2 was realized with an infrared distance sensor (SHARP GP2Y0A02YK0F) and two absolute rotary encoders (P3015 Series Hall Rotary Encoder). Their signals were acquired by a programmable system-on-chip (PSoC) microcontroller (Cypress, CY8CKIT-059 PSoC 5 LP prototyping kit) and a built-in 12-bit analog-to-digital converter (ADC) with a sampling rate of 2000 samples per second (s/s). The data were filtered with a median filter and subsequently downsampled to 50 s/s to calculate the position difference $\Delta \overrightarrow{R_{MR}}$. Then, the position difference was further converted to a pulse width through a PID controller and a PW mapper. Through the built-in PWM module of the PSoC, the pulse widths were transmitted to the motor drivers (DC 5–12 V 0–30 A Dual-channel H Bridge Motor Driver Controller) to drive the DC geared motor (GW4058-31ZY DC worm gear motor) and Mecanum wheels. The system structure block diagram was shown in Figure 4. The appearance of the tested accompanying robot was shown in Figure 5.
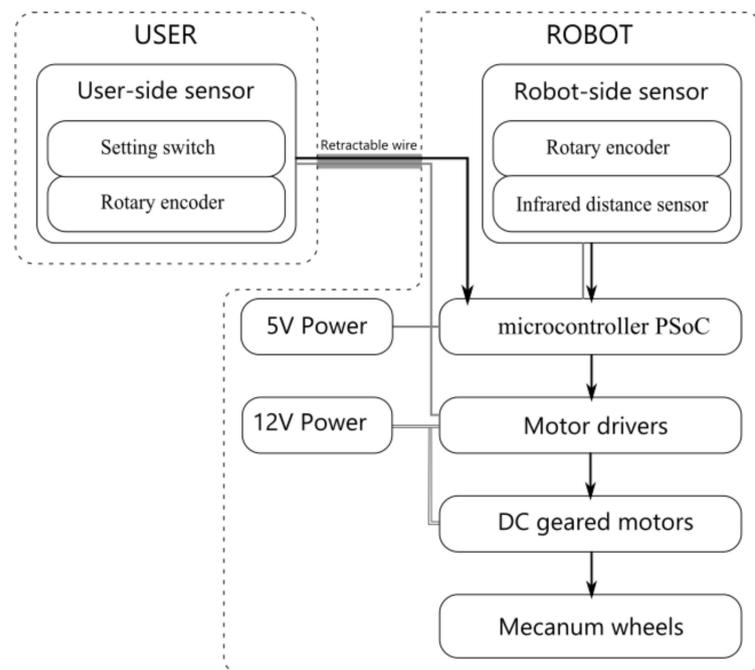
Figure 4. System structure block diagram.
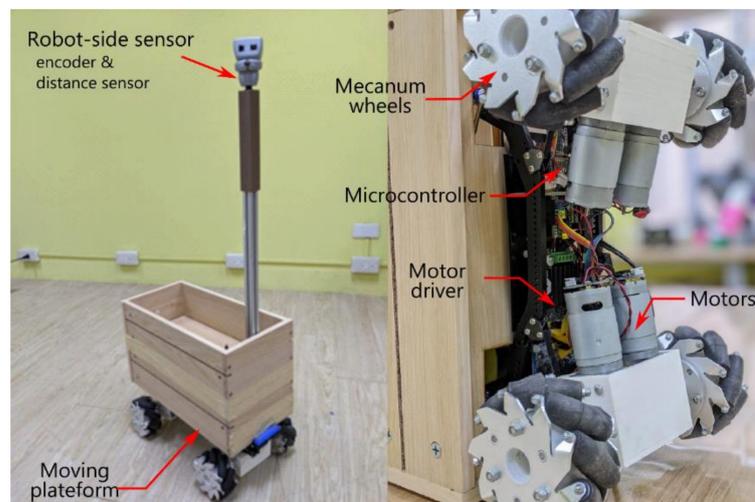
Figure 5. Assembly of the tested accompanying robot.

3. Methods of System Verification

3.1. Testing Tasks

To assess the performance of the proposed accompanying method, six basic walking tasks and two types of combined movement were performed. The six basic walking tasks were: forward walking, backward walking, left lateral walking, right lateral walking, a left pivot of 90°, and a right pivot of 90°. These tasks were performed with a 2.5 m × 1.8 m rectangle of the force plate border as a reference. The two types of combined maneuvers were (1) walking along the force plate while the user maintained a front-facing orientation and (2) walking around the force plate clockwise or counterclockwise. The border of the force plate was used as the walking reference. In addition, the user arbitrarily set or selected the target position. Although the robot's performance should be identical with different accompanying positions, it was purposely set in front of the user to demonstrate our proposed method capability.

3.2. Testing Environment

The tests were performed in our motion lab, which was equipped with an 8-camera VICON motion capture system (VICON Motion Systems, Oxford, UK). Four markers were placed on the user's pelvis and bilateral anterior and posterior iliac spine (ASIS and PSIS) to calculate the user's orientation with respect to the lab's coordinates; four markers were placed on the shelf above the four Mecanum wheels of the robot to calculate the orientation of the robot with respect to the lab's coordinates. Two other markers were placed on the housing of the user-end and robot-end rotary encoders as the origins of LCS_U and LCS_R, respectively. The origins were not at the centers of the user and robot.

The data from the VICON system were processed to obtain the trajectory of the user and robot by LabVIEW to calculate the orientation of the user and robot, distance between the origins of LCS_U and LCS_R, and relative orientations of LCS_R and LCS_U with respect to each other. The data were also used to calculate the robot's trajectory in LCS_U.

4. Results of System Verification

Table 1 lists the positions and displacements of the user and robot in the *WCS*. The origin of the WCS was set at the user's start position. The direction and orientation of the WCS were determined with respect to the lab coordinates. First, the user faced the positive Y axis. The robot's start position was the setting position of the WCS. The displacement was the stop position subtracted from the start position. The movement displacements of the user and robot in the Y direction were 1914 mm and 1933 mm in the forward-walking task, respectively. The movement displacements of the user and robot in the Y direction were −1980 mm and −1985 mm in the backward-walking task, respectively. The displacements of the user and robot in the X direction were −1520 mm and −1732 mm in the left lateral walking task, respectively. The displacements of the user and robot in the X direction were 1558 mm and 1666 mm in the right lateral walking task, respectively. The turning angles of the user and robot changed by 73.7 degrees and 71.6 degrees in the left pivot turning task, respectively. The turning angles of the user and robot changed by −92.1 degrees and −88.4 degrees in the right pivot turning task, respectively.

Table 1. Result of Walking Tasks in the *WCS*.

Tests	Start Position			Stop Position			Displacement		
	X (mm)	Y (mm)	γ (°)	X (mm)	Y (mm)	γ (°)	X (mm)	Y (mm)	γ (°)
Forward walking									
user	0	0	99.6	−3	1914	91.8	−3	1914	−7.8
robot	24	438	78.1	90	2361	77.9	66	1933	−0.2
Backward walking									
user	0	0	92.4	28	−1980	92.8	28	−1980	0.4
robot	88	451	78.8	82	−1534	81.9	−6	−1985	3.1
Left lateral walking									
user	0	0	97.4	−1520	64	92.0	−1520	64	−5.4
robot	220	583	95.7	−1512	582	108.4	−1732	−2	12.7
Right lateral walking									
user	0	0	92	1558	−34	90.4	1558	−34	−1.8
robot	−8	447	83.8	1658	363	77.9	1666	−84	−5.9
Left pivot turning									
user	0	0	97.2	−265	−222	170.9	−266	−222	73.7
robot	10	417	95.9	−796	−279	167.5	−807	−696	71.6
Right pivot turning									
user	0	0	95.5	309	−165	3.4	309	−165	−92.1
robot	34	588	98.6	871	−209	10.1	837	−797	−88.4

Table 2 lists the robot's start, target, and stop positions in LCS_U. The stop position is defined as the point where the user stopped walking and the robot settled. X_C is the robot

position w.r.t. LCS_U in the lateral direction, Y_C is the robot position w.r.t. LCS_U in the front direction, and γ_C is the robot's orientation w.r.t. LCS_U.

Table 2. Result of Walking Tasks in LCS_U.

Tests	Target/Start Position			Stop Position			Average Position during Walking		
	X_C (mm)	Y_C (mm)	γ_C (°)	X_C (mm)	Y_C (mm)	γ_C (°)	X_C (mm)	Y_C (mm)	γ_C (°)
Forward walking	95	418	−12.5	107	443	−13.6	111 ± 33	320 ± 40	−19.3 ± 6.5
Backward walking	106	447	−13.2	75	443	−9.7	128 ± 32	574 ± 44	−12.5 ± 3.1
Left lateral walking	292	550	−28.0	26	517	−2.9	259 ± 31	482 ± 36	−28.3 ± 3.3
Right lateral walking	9	447	−0.7	101	397	−14.3	−159 ± 50	422 ± 13	20.7 ± 6.5
Left pivot turning	63	412	−8.8	−9	533	1.0	310 ± 167	440 ± 26	−32.6 ± 14.9
Right pivot turning	91	582	−8.8	−36	563	3.7	−106 ± 189	536 ± 54	11.5 ± 19.7
Walking along a rectangle	−121	384	17.5	−98	376	14.6	−115 ± 125	323 ± 85	19.8 ± 20.3
Clockwise-curve walking	42	541	−4.4	73	514	−8.0	11 ± 72	347 ± 53	−1.8 ± 11.6
Counterclockwise-curve walking	−51	530	5.5	−147	492	16.6	14 ± 241	315 ± 74	−1.6 ± 37.8

4.1. Basic Walking Tasks

In the basic walking tasks, the user was asked to perform the tasks inside the rectangular force plate in the center of the motion lab. Figure 6 shows the trajectories of the walking tasks w.r.t. the WCS, where the corresponding origins of LCS_U (symbolized '○') and LCS_R (symbolized '●') are linked by line segments. The protruding hairline symbols represent medial-lateral directions. The interval between consecutive corresponding point pairs was 200 milliseconds. Lags and overshoots were observed at the beginning and end, respectively. The target/starting position was arbitrarily selected by the user at the beginning of each test. Figure 6a shows the result of the user walking forward 6 steps. Figure 6b shows the result of the user walking backward 8 steps. Figure 6c shows the result of the user performing left lateral walking for 8 steps. Figure 6d shows the result of the user performing right lateral walking for 8 steps.

Figure 7 shows the time series of the sensing values for the walking tasks that correspond to the distance between the robot and the user and the encoder values θ_{Robot}, θ_{User}. The gaps between the set distance and actual distance and between θ_{User} and the target angle represent the lag and overshoot of the track. Lag tracking can be demonstrated in the increased gap between θ_{User} and θ_{Robot}. The distance between the user and the robot decreased while the user walked forward and increased while the user walked backward due to the phase lag.

4.2. Combined Tasks

4.2.1. Walking along a Rectangle

In the condition of walking along a rectangle, the user was instructed to walk along the rectangular force plate in the center of the motion lab. The movements along the rectangle in this condition included walking forward, laterally right, backward, and laterally left with eight steps per direction. The results are presented in Figure 8. The accompanying robot could follow the user walking along the rectangle. Shown are the trajectories of the walking tasks where the corresponding origins of LCS_U (symbolized '○') and LCS_R (symbolized '●') are linked by line segments. The protruding hairline symbols represent the medial-lateral directions.

4.2.2. Clockwise-Curve Walking Test

In the clockwise-curve walking test, the user was instructed to walk in a circle around the force plate clockwise, which combined forward walking and small right lateral and right turning movements. As shown in Figure 9, the starting position of the robot was on the front right side of the user, and the trajectory was almost on the user's walking curve.

Shown are the trajectories of the walking tasks where the corresponding origins of LCS_U (symbolized '○') and LCS_R (symbolized '●') are linked by line segments. The protruding hairline symbols represent medial-lateral directions. The total walking steps were 23 steps in this clockwise-curve walking test.
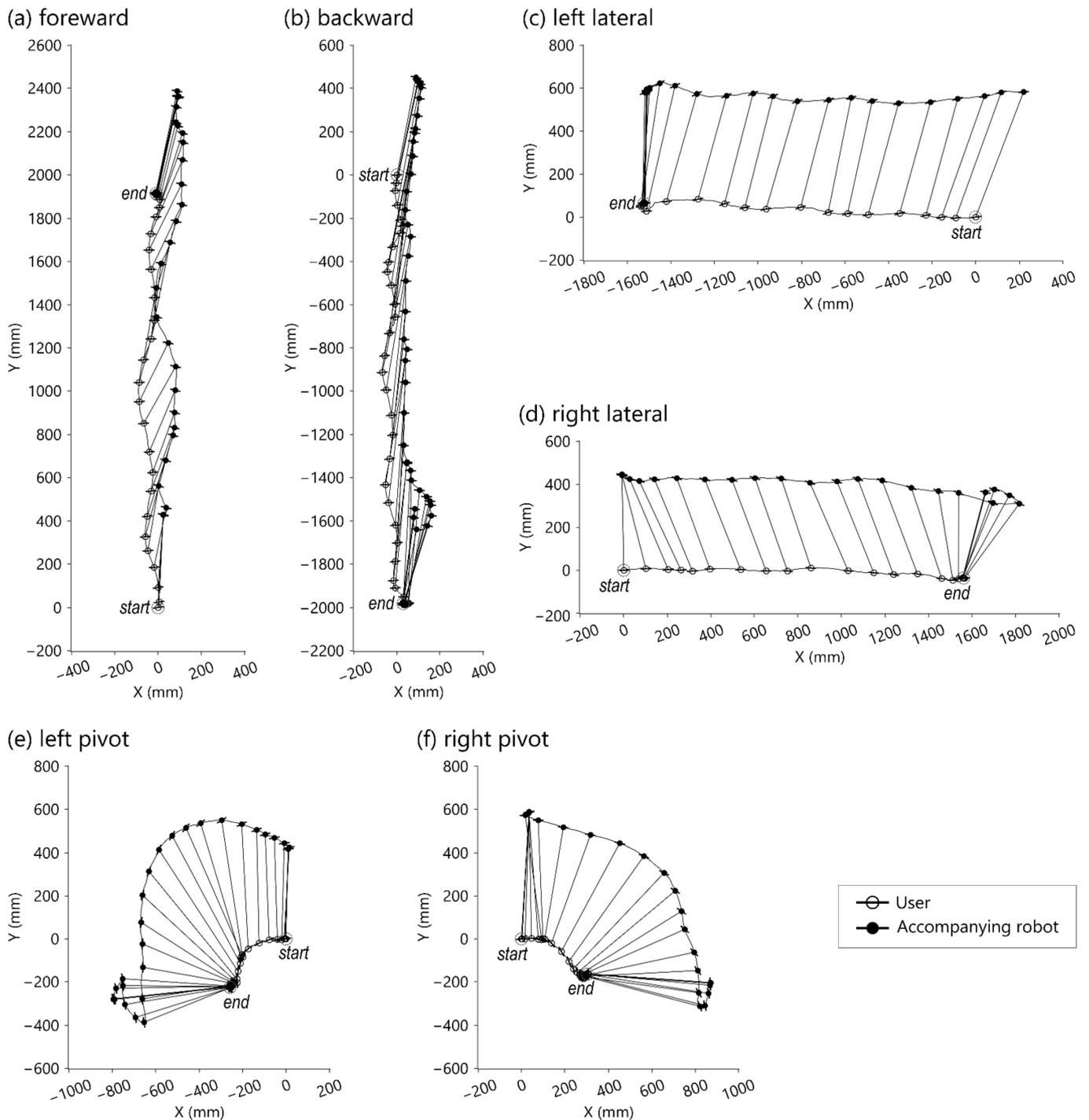
Figure 6. Top-view motion graph of the movements of the robot and user w.r.t. the WCS. (**a**) Forward walking test; (**b**) backward walking test; (**c**) left lateral walking test; (**d**) right lateral walking test; (**e**) walking test with pivot turning to the left; (**f**) walking test with pivot turning to the right.
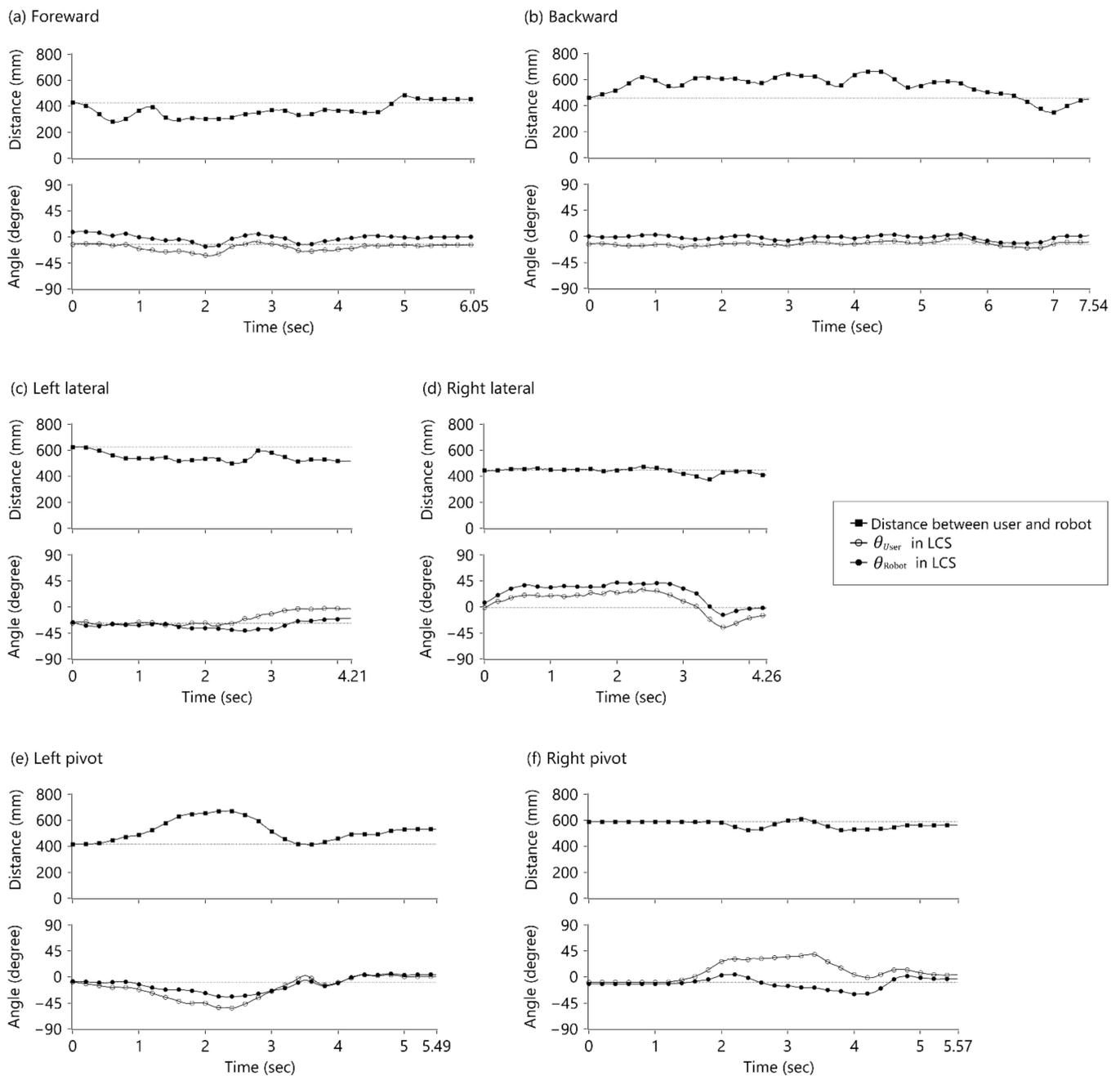
Figure 7. Relative distance between the robot and the user w.r.t. the LCS and encoder values θ_{Robot}, θ_{User} in the LCS. (**a**) Forward walking test; (**b**) backward walking test; (**c**) left lateral walking test; (**d**) right lateral walking test; (**e**) walking test with pivot turning to the left; (**f**) walking test with pivot turning to the right.
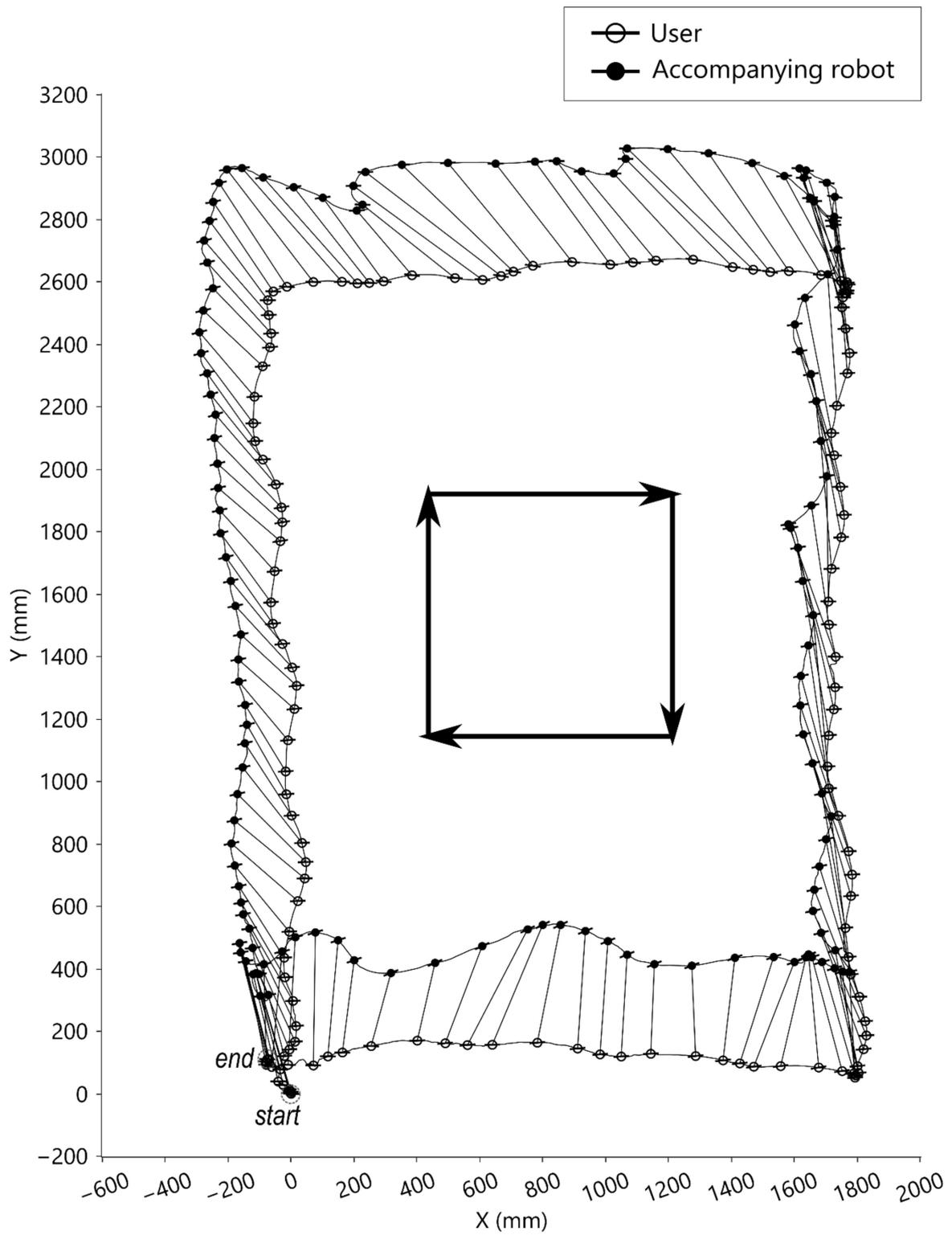
Figure 8. Top-view motion graph of walking along the rectangle.
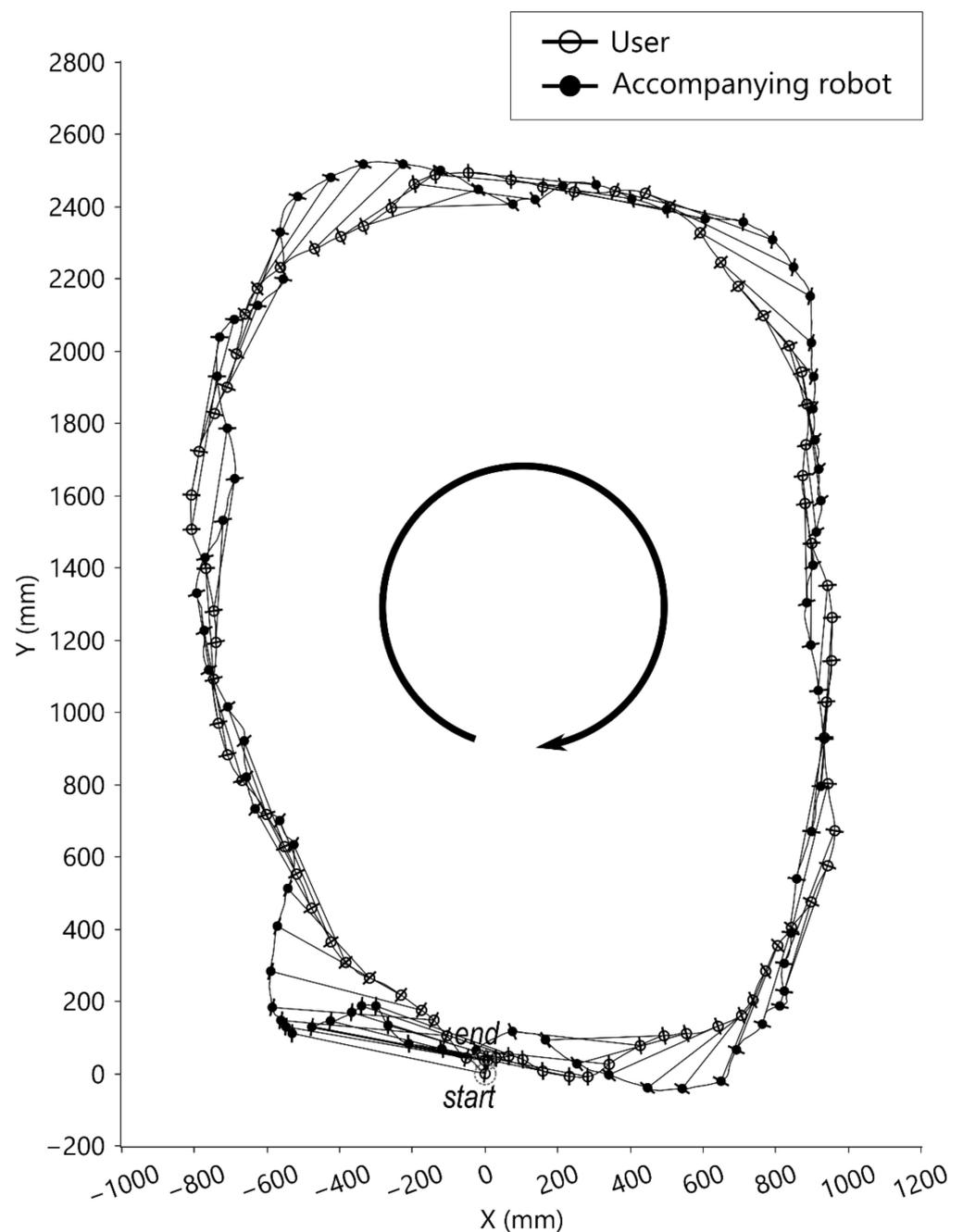
Figure 9. Top-view motion graph of clockwise walking.

4.2.3. Counterclockwise-Curve Walking Test

In the counterclockwise-curve walking test, the user was instructed to walk in a circle around the force plate counterclockwise, which combined forward, left lateral, and left turning movements. As shown in Figure 10, the starting position of the robot was on the front right side of the user. When the user walked counterclockwise, the trajectory of the robot enclosed the trajectory of the user. Shown are the trajectories of the walking tasks where the corresponding origins of LCS_U (symbolized '○') and LCS_R (symbolized '●') are linked by line segments. The protruding hairline symbols represent the medial-lateral directions. There were 28 walking steps in total in this counterclockwise-curve walking test.
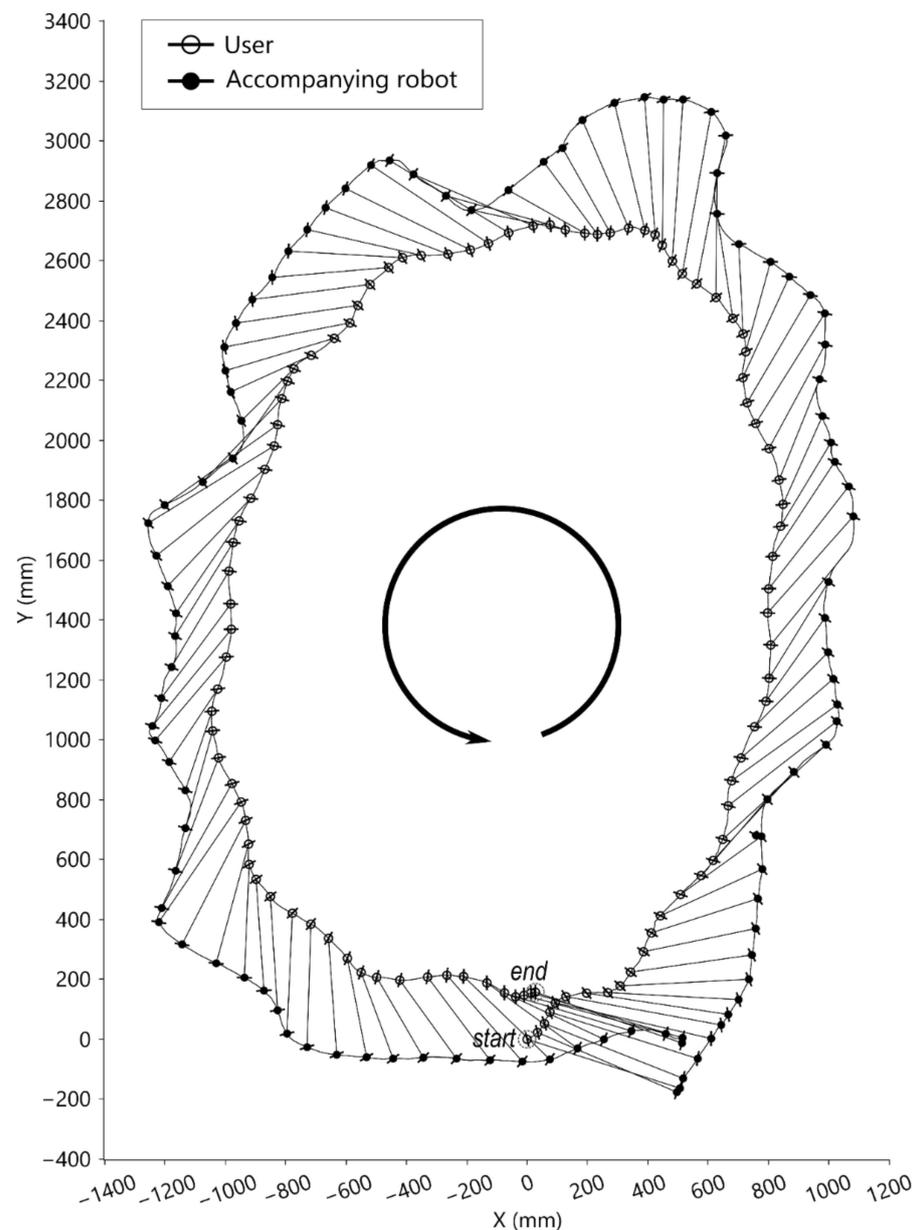
Figure 10. Top-view motion graph of counterclockwise walking.

5. Discussion

The results show that the robot can follow the user in the designated position while the user performs forward, backward, and lateral movements, turning, and curve walking. In other words, this study has demonstrated that with the proposed tracking strategy, the accompanying robot can closely follow the user in sideways or sharp turn movements. In addition, it can accompany the user in front even when the user moves backwards.

Each test had different start/target positions, which were arbitrarily set. Regardless, the difference between the end position and the set/target position in each test was within an acceptable range for daily accompanying. There were phase lags and overshoots at the end of the tests. Further tuning the PID controller or replacing it with other control algorithms could improve the tracking performance. For example, there appeared to be a small phase lag and no overshoot in the left lateral walking test (Figure 6c and Table 1). After checking the data, we found that the robot was further to the right of the user, and the distance sensor measured the user's forearm instead of the trunk when setting the target position. In other words, unlike the other tests, in the left lateral walking test, the target

distance was shorter than the actual distance between the robot and the user from the beginning. This mistake suggests a control method where changing the target distance may reduce the phase lag and overshoot, i.e., when the user is approaching the robot, e.g., the robot is in front of the user and the user is walking forward, the target distance can be set to a greater value; when the user is moving away from the robot, the target distance can be set shorter. Further control techniques, including system identification, modeling, and simulation, to improve system performance and stability should be examined in the future.

In this study, the selected distance sensor (SHARP GP2Y0A02YK0F) is less affected by environmental light. However, the sensing range was limited from 20 cm to 150 cm, and some measuring errors may occur when the user walks too close to the robot. The absolute rotary encoder (P3015 series hall rotary encoder) is less affected by friction. However, the retractable conducting wire may swing, and the user can rotate his/her body while moving. Consequently, the sensor may not always measure the same point of the user. The trajectory figures (Figures 6 and 8–10) show that there were natural rotations and sideways movements of the user's pelvis; thus, the user's coordinates were not steady, which caused fluctuating target positions in the WCS. There were three possible solutions to remove unnecessary fluctuations in the robot: (1) use a laser range finder; (2) reduce the weight of the retractable wire or become wireless; and (3) reduce the loop rate of the controller. However, a low loop rate can compromise the trajectory tracking performance. Adding a movement filter to smoothen the motion may be better.

In addition to the controller and transducers, the mechanical design of the robot affects the tracking performance. First, the inertia and driving torque of the robot affect the acceleration. Actuator saturation is experienced when the user moves too fast. It can be solved by using high-torque motors or by an algorithm [25]. Driving motors with higher torque can respond more quickly to the user's sudden movements, but consume more power and increase system weight. Second, the 3 DOFs of the moving platform simplify the control strategy and save time and space in tracking the target position. In this study, this aspect was implemented with Mecanum wheels [23], but other biomimetic techniques [21,26] or unmanned aerial vehicles [22,27] can also be used.

The key point of the proposal was determining the positions of the user and robot w.r.t to each other. We achieved this task using the novel "string-pot" sensor system. It is compact, reliable, and immune to environmental noise such as light. However, with this "string-pot" sensor system, the user-side sensor is not suitable to be worn on the lateral side of the user's body because the sensor and wire will interfere with the user's hand. To remove the wire, i.e., create a wireless system, further challenges must be overcome, such as target user identification and user orientation determination. Facial recognition [28] may solve part of this problem, but issues such as the user looking around without changing the body's position while the robot is following behind must be considered. Obtaining the relative distances and orientations of the user and robot without physical wires is possible. Two inertial measurement units (IMUs) [18,19], mounted on both user side and robot side, and a laser range finder [20] can provide information on the distance and orientation between the robot and the user.

The excursion trajectories of the robot are greatly affected by the target position and user movement tracking. As shown in Figure 6e,f, the robot took a longer path while the user pivoted. Furthermore, as shown in Figures 9 and 10, the user performed similar lengths of circular walks, but the robot trajectories were different. One of the robot's trajectories was similar to the user's trajectory, and the other of the robot's trajectories was larger than the user's trajectory. The reason was that one of the robot's target positions was on the walking curve/circle, and the other was outside of the walking curve/circle. If the target point is outside of the instance center's circle of the user's moving curve, the robot is expected to take a longer path than the user.

6. Conclusions

We have demonstrated that a robot with one algorithm and simple controller can accompany the user in a selected position while the user performs forward, backward, and sideways or sharp turn movements. i.e., the user can move naturally, and the robot can maintain a designed position w.r.t. the user. This goal was achieved using three main factors: (1) knowledge of the positions of the user and robot w.r.t their local coordinate systems; (2) agile movement of the robot; and (3) a quickly updating feedback loop. In the future, using wireless sensors that combine the obstacle avoidance technique and a high-level method to determine the set position of the robot in LCS_U, the accompanying robot can further help the user in daily life activities.

References

1. Islam, M.J.; Hong, J.; Sattar, J. Person-following by autonomous robots: A categorical overview. *Int. J. Robot. Res.* **2019**, *38*, 1581–1618. [CrossRef]
2. Nishimura, S.; Takemura, H.; Mizoguchi, H. Development of attachable modules for robotizing daily items -Person following shopping cart robot. In Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 15–18 December 2008.
3. Travelmate Robotics. Available online: https://travelmaterobotics.com/ (accessed on 20 May 2021).
4. Piaggio Fast Forward. Available online: https://www.piaggiofastforward.com/ (accessed on 19 May 2021).
5. CaddyTrek. Available online: https://caddytrek.com/ (accessed on 20 May 2021).
6. Jung, E.-J.; Yi, B.-J.; Yuta, S. Control algorithms for a mobile robot tracking a human in front. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012.
7. Hu, J.-S.; Wang, J.-J.; Ho, D. Design of Sensing System and Anticipative Behavior for Human Following of Mobile Robots. *IEEE Trans. Ind. Electron.* **2014**, *61*, 1916–1927. [CrossRef]
8. Ferrer, G.; Garrell, A.; Sanfeliu, A. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013.
9. Mi, W.; Wang, X.; Ren, P.; Hou, C. A system for an anticipative front human following robot. In Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering, Lisbon, Portugal, 29–31 July 2016.
10. Nikdel, P.; Shrestha, R.; Vaughan, R. The hands-free push-cart: Autonomous following in front by predicting user trajectory around obstacles. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018.
11. Fritsch, J.; Kleinehagenbrock, M.; Lang, S.; Fink, G.; Sagerer, G. Audiovisual person tracking with a mobile robot. 2004. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.7741&rep=rep1&type=pdf (accessed on 4 June 2020).
12. Huskić, G.; Buck, S.; González, L.A.I.; Zell, A. Outdoor person following at higher speeds using a skid-steered mobile robot. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.

13. Triebel, R.; Arras, K.; Alami, R.; Beyer, L.; Breuers, S.; Chatila, R.; Chetouani, M.; Cremers, D.; Evers, V.; Fiore, M.; et al. SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports. In *Field and Service Robotics*; Springer: Cham, Switzerland, 2016; pp. 607–622.
14. Jevtic, A.; Doisy, G.; Parmet, Y.; Edan, Y. Comparison of interaction modalities for mobile indoor robot guidance: Direct physical interaction, person following, and pointing control. *IEEE Trans. Hum.-Mach. Syst.* **2015**, *45*, 653–663. [CrossRef]
15. Chen, B.X.; Sahdev, R.; Tsotsos, J. Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot. In *International Conference on Computer Vision Systems*; Springer: Cham, Switzerland, 2017.
16. Jiang, S.; Yao, W.; Hong, Z.; Li, L.; Su, C.; Kuc, T.-Y. A Classification-Lock Tracking Strategy Allowing a Person-Following Robot to Operate in a Complicated Indoor Environment. *Sensors* **2018**, *18*, 3903. [CrossRef] [PubMed]
17. Wang, X.; Zhang, L.; Wang, D.; Hu, X. Person detection, tracking and following using stereo camera. In Proceedings of the Ninth International Conference on Graphic and Image Processing (ICGIP 2017), Qingdao, China, 14–16 October 2017. [CrossRef]
18. Cifuentes, C.A.; Frizera, A.; Carelli, R.; Bastos, T. Human–robot interaction based on wearable IMU sensor and laser range finder. *Robot. Auton. Syst.* **2014**, *62*, 1425–1439. [CrossRef]
19. Brookshire, J. Person Following Using Histograms of Oriented Gradients. *Int. J. Soc. Robot.* **2010**, *2*, 137–146. [CrossRef]
20. Chung, W.; Kim, H.; Yoo, Y.; Moon, C.; Park, J. The Detection and Following of Human Legs Through Inductive Approaches for a Mobile Robot With a Single Laser Range Finder. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3156–3166. [CrossRef]
21. Dürr, V.; Arena, P.P.; Cruse, H.; Dallmann, C.J.; Drimus, A.; Hoinville, T.; Krause, T.; Mátéfi-Tempfli, S.; Paskarbeit, J.; Patanè, L.; et al. Integrative Biomimetics of Autonomous Hexapedal Locomotion. *Front. Neurorobotics* **2019**, *13*, 88. [CrossRef] [PubMed]
22. Xiao, X.; Fan, Y.; Dufek, J.; Murphy, R. Indoor UAV Localization Using a Tether. In Proceedings of the 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, USA, 6–8 August 2018.
23. Dickerson, S.L.; Lapin, B.D. Control of an omni-directional robotic vehicle with Mecanum wheels. In Proceedings of the NTC '91—National Telesystems Conference Proceedings, Atlanta, GA, USA, 26–27 March 1991.
24. Taheri, H.; Qiao, B.; Ghaeminezhad, N. Kinematic model of a four mecanum wheeled mobile robot. *Int. J. Comput. Appl.* **2015**, *113*, 6–9. [CrossRef]
25. Su, Y.; Zheng, C.; Mercorelli, P. Global Finite-Time Stabilization of Planar Linear Systems With Actuator Saturation. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 947–951. [CrossRef]
26. Delcomyn, F.; Nelson, M.E. Architectures for a biomimetic hexapod robot. *Robot. Auton. Syst.* **2000**, *30*, 5–15. [CrossRef]
27. Chakrabarty, A.; Morris, R.; Bouyssounouse, X.; Hunt, R. Autonomous indoor object tracking with the Parrot AR. Drone. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016.
28. Fukui, K.; Yamaguchi, O. Face recognition using multi-viewpoint patterns for robot vision. In *Robotics Research. The Eleventh International Symposium*; Springer: Berlin/Heidelberg, Germany, 2005.