

Research Article

Navigation Path Following Platform for a Greenhouse Shuttle Robot Using the State-flow Method

Heonjong Yoo ¹, Donkyu Baek,¹ and Seong-gon Choi ²

¹Chungbuk National University, Cheongju-si, Republic of Korea

²Department of Electrical and Computer Engineering, Chungbuk National University, Chungdae-ro, Seowon-gu, Cheongju-si, Chungcheongbuk-do 28644, Republic of Korea

Correspondence should be addressed to Seong-gon Choi; sgchoi@chungbuk.ac.kr

Received 2 May 2023; Revised 11 January 2024; Accepted 25 January 2024; Published 19 February 2024

Academic Editor: Lei Liu

Copyright © 2024 Heonjong Yoo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Localization is an important method for autonomous indoor robots to recognize their positions. Generally, the navigation of a mobile robot is conducted using a camera, Lidar, and global positioning system. However, for an indoor environment, GPS is unavailable. Therefore, a state-trajectory tracking method is utilized based on a Lidar map. This paper presents the path following of an autonomous indoor mobile robot, that is, a shuttle robot, using a state-flow method via a robot operating system network. MATLAB and Linux high-level computers and an inertial measurement unit sensor are used to obtain the Cartesian coordinate information of a bicycle-type mobile robot. The path following problem can be solved in the state-flow block by setting appropriate time and linear and angular velocity variables. After the predetermined time, the linear and angular velocities are set based on the length of the path and radius of the quarter-circle of the left and right turns in the state-flow block, path planning, which can execute the work effectively, is established using the state-flow algorithm. The state-flow block produces time-series data that are sent to Linux system, which facilitates real-time mobile platform path following scenario. Several cases within the path-following problem of the mobile robot were considered, depending on the linear and angular velocity settings: the mobile robot moved forward and backward, turned in the right and left directions on the circular path. The effectiveness of the method was demonstrated using the desktop-based indoor mobile robot control results. Thus, the paper focuses on the application of the state-flow algorithm to the shuttle robot specifically in the narrow indoor environment.

1. Introduction

New models are often encountered in the agricultural, protection, and precision sectors. Computers, artificial intelligence, and big data technologies have been used to develop intelligent farming systems, particularly for farming robots [1]. With the rapid development of advanced technologies, many new techniques, such as big data, artificial intelligence, the Internet of Things, machine vision, and agricultural robotics, have been applied to the agricultural production [2]. These robots can be classified as grafting robots, robots for transport, spraying robots, robots to collect agricultural information, weeding robots, and seeding robots [3]. Robotics and sensing technologies have led to a technological revolution in agriculture [4]. Unmanned guided vehicle (UGV) projects are crucial for the ongoing research. Many technologies relevant to companies

(Google Car, VIP, Tesla, Mercedes, Renault, PSA, Nissan, Fiat, etc.) have already been implemented, according to the study by Khalifa et al. [5]. Stanley control and lateral speed control were introduced [5], however, they were controlled using the vector interval rather than the time interval. Recently, path tracking has become an important subject in the engineering problems. This method can be applied to both agricultural and marine environments. A conventional method for automatic path tracking was introduced [6]. Recently, linear active disturbance rejection control (LADRC) was designed for the trajectory tracking control problem of a differential-type model [7]. A recursive technique was applied to the path-tracking problem of the differential-type model by composing a chained form of the system [8]. Recently, the pure pursuit method has been widely used for the path tracking of outdoor and indoor mobile robots. However, these algorithms and methods are

controlled using the vector interval, which causes the mobile platform to vibrate during implementation. Automated navigation technology plays a crucial role in the autonomous navigation of the mobile robots in agriculture [9]. Researchers have developed several automated path-following methods based on various navigation principles. Using an electromagnetic sensor, Song [10] developed an automatic navigation of the spray robot. It is difficult to maintain a navigation system [11]. The exploration of autonomous robots is challenging, particularly in the agricultural field, where the environment is organized by half an uneven surface that changes daily owing to dust and fog, affecting sensor observations and a lack of operational feature localization [11]. Autonomous agricultural robots in this sector are an important part of efficient exploration strategies because battery problems limit their operating time. The advantage of the pure pursuit algorithm is that the path of the mobile robot follows waypoints. Several methods have been applied to the agricultural field operations for robotic localization. Localization systems such as the global positioning system (GPS) [12], real-time kinematic GPS (RTK-GPS) [13], geographic information systems [14], and Lidar-based systems have been applied to agricultural mobile robotic systems [15]. Among these methods [16], the RTK-GPS method has been proven to achieve the highest accuracy in robot localization efficiency [17]. However, this method has an extremely high cost. Hence, from the perspective of commercialization, this method is rarely used in the field applications. Before a test, state-flow algorithms can be tested for their efficiency and effectiveness through simulations. Furthermore, a test bed can be constructed to match the specific indoor utilization. The mobile platform presented in this paper is called a unicycle-type mobile platform and has been researched in several studies [6, 18–25]. A recent algorithm has been applied to unicycle-type system modeling in the several studies. The recent linear active disturbance rejection control (LADRC) has been applied to unicycle-type system modeling [26]. For optimal control, a gradient descent algorithm has been studied to solve the trajectory problem [27]. This paper introduces a novel state-flow algorithm recently introduced by MathWorks. Several researchers [28, 29] have investigated the state-flow method, namely, its applicability to flight control, hybrid energy control system design, and simulation. The benefit of the state-flow method is that the platform is controlled using time interval control rather than vector interval control, which significantly reduces the vibration of the platform via the waypoint follower: the pure pursuit and, Stanley methods. Therefore, we determined that the state-flow method is appropriate for the path-tracking problem in an indoor mobile AGV environment. This is because the space in the indoor environment is very narrow, and the path tracking method in the state-flow follows an imaginary path smoothly and accurately. Thus, stationary obstacles can be easily avoided using the state-flow method. In this study, we demonstrated that the proposed method can easily and accurately track an imaginary path by setting a rope during the experiment. The paper presents the results of the mobile platform following the imaginary path (previously, the rope was set

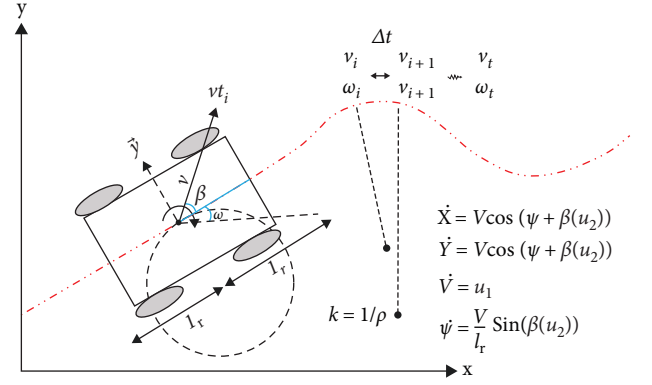


FIGURE 1: Design of the unicycle-type mobile robot.

smoothly and accurately. A similar wireless communication network is described [30].

Thus, the contribution of this paper can be expressed as follows:

- (1) Despite the various aforementioned control and estimation methods, the experiment was conducted focusing on the state-flow introduced in MATLAB/Simulink recently.
- (2) The actual and reference path is located in the Lidar map which reflects the real measured environment.
- (3) The path tracking problem can then be solved in a state-flow block by setting appropriate time, velocity, linear velocity, and angular velocity variables. Thus, the effectiveness of the method is verified by the results of a desktop-based indoor mobile robot control.

In summary, Section 2 describes the hardware configuration and how the platform connects to high-level computers, as well as modeling a bicycle-type system. The results obtained by testing the obstacle avoidance path and the indoor driving path using the radius of a circle after applying the algorithm to the state-flow are provided in Section 3. Finally, conclusions are given in Section 4.

2. Materials and Methods

2.1. Autonomous Navigation Systems and Proposed Methodology.

In this section, we introduce the trajectory control system, kinematics, and motion model of the mobile robot used in this study. The block diagram is similar to that of the proposed kinematic motion control, and the basic indoor mobile robot chassis structure and reference axis definition are shown in Figure 1. In this study, a unicycle-like mobile robot model was selected because it is the most common type of mobile robot used in various applications such as surveillance, floor cleaning, and autonomous wheelchair applications. Unicycle mobile robots are used in agricultural application [31, 32]. Path planning can be classified into global and local-path planning. The selected kinematic model for the indoor mobile robot is represented by Equation (1).

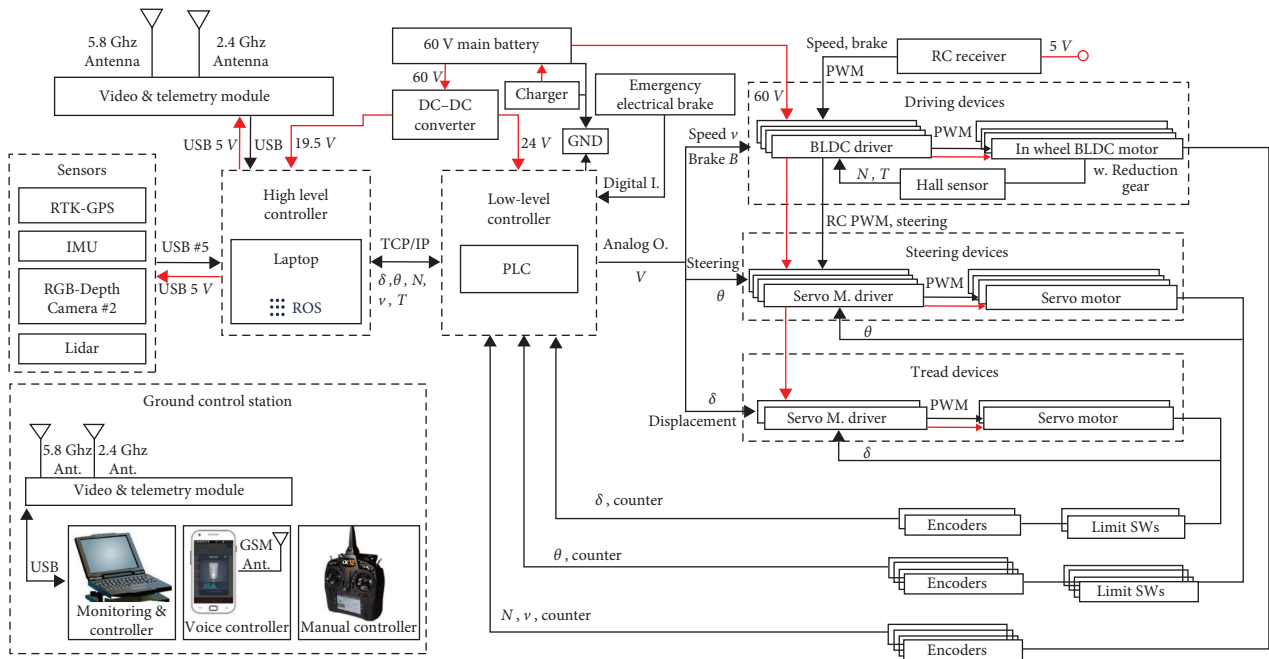


FIGURE 2: Hardware configuration.

Layout of component for a 4 wheel mobile robot platform

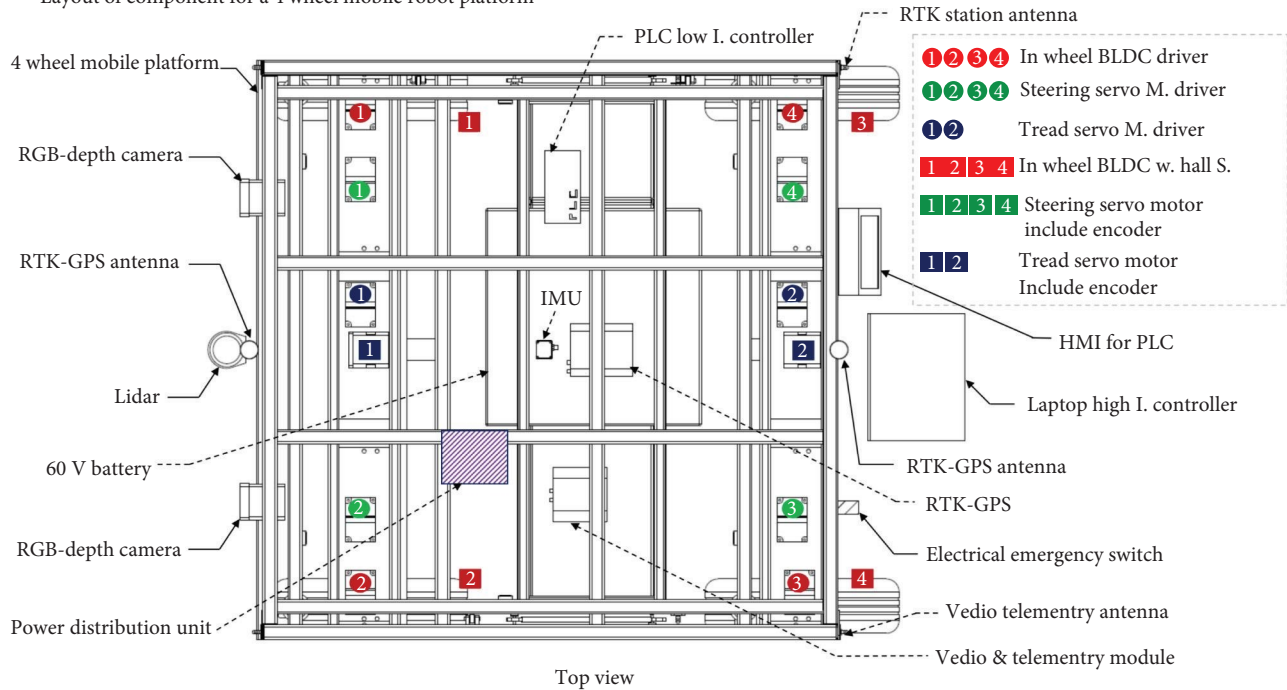


FIGURE 3: Layout of hardware on the top view.

The hardware configuration experimented in the Chungbuk National University is given as Figure 2. The function of high level computer reduces the speed error between each motors, while the function of low-level controller is to track the reference speed of the eight motors. Driving devices and steering devices are each four wheels described in Figure 2.

The top, bottom, and front view of the four wheel mobile platform is described as Figures 3 and 4.

The actual view of the four wheel mobile platform is described as Figure 5. The rear two wheels are idler type, and front two wheels can be controlled by linear and angular velocities.

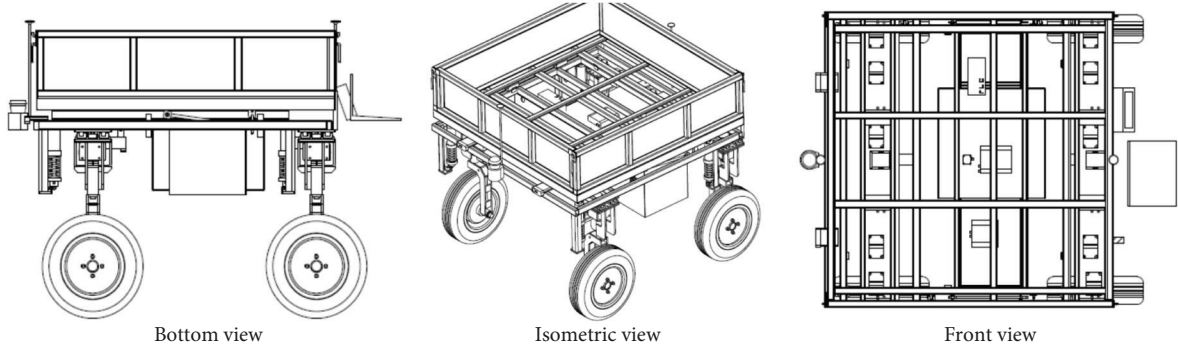


FIGURE 4: The description of four wheel mobile platform.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ 0 & \sin(\theta(t)) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix}. \quad (1)$$

Trajectory generation with constraints, a predetermined path is obtained using the state-flow method. First, the linear and angular velocities v_i and w_i are computed spatially along the entire path. Equations (2)-(4) are used for controlling the radius of the circle by changing the linear and angular velocities inputs. Using the well-known programming environment Simulink, time moments are assigned to these velocities to generate v_i and w_i . At all times, a unicycle robot has a linear velocity v_i and angular velocity w_i , and their values can be generated on a different trajectories over time. The spatial variables v_i and w_i in Simulink are determined by the curvatures K at each point along the path. In the design procedure, we begin by determining v_i and w_i spatially based on the curvature K at a specific point along the path, and it is represented by Equations (6) and (7). In the first step, the path tracking is not possible if the trajectory $K > K_{\max}$ is not trackable that explains that the value of the linear velocity is not more than two in the following experiment. If $K > K_{\min} = \omega_{\max} v_{\max}$ there will always be sufficient rotation speed to maintain the linear speed and the expected curvature. Therefore, the maximum linear speed v_{\max} is always used to define v_i and rotational speed w_i is determined by $w_i = K_{\min} \times v_{\max}$. The second step is automatically assigned the time dimension to V_i and w_i and then transform it into v_i and w_i . This can be achieved by determining a series of the small time-interval, P_i . The small time-interval is determined by the sampling time in Simulink, between P_i and P_{i+1} on the path, assuming that the spatial linear and angular velocities variables v_i and w_i are constant within the short time interval.

2.2. Motion Control of the Differential Indoor Mobile Robot.

This section presents the real-time experimental results derived in this study. A desktop-based mobile robot control simulation was performed using ROS in MATLAB. We considered the problem of tracking a straight line at a constant velocity. Using the above commands, the user's high-level computer was connected to a low-level computer using Linux. Step 1: Run the Simulink model for a specified time. Step 2: After the experiment, plot the reference and actual mobile—robot trajectories using a MATLAB plot



FIGURE 5: The actual hardware platform.

command. The following procedure completes the state-flow algorithm described in Algorithm 1.

2.3. ROS Mobile Robot System Architecture. In the Simulink ROS, the position measurements $x(t)$, $y(t)$ and heading angle $\theta(t)$ were the measurable output variables. Furthermore, the linear and angular velocities were the input variables. The aim was to create a state-flow control law; $v(t)$, and $w(t)$ enabled the mobile robot to follow the reference trajectory. The overall process is depicted in Figure 6. Linear and angular velocities were calculated and set in the state-flow based on the constraint theorem described in Section 2.2. Thereafter, the user could plot the robot's moving direction using the "PLOT" command in the MATLAB prompt. The experimental results are shown in Figure 7. The collected

Input: Robot linear velocity: v_i angular velocity: w_i output: motor output reference value: u_L, u_R . Define: The robot maximum linear velocity as u_{\max} , turning radius as radius, mobile robot chassis width as width, maximum output value: Pw_{\max}

$$X_{\text{actual}} = [X_{a1} \ X_{a2} \ X_{a3} \ \dots \ X_{an}], Y_{\text{actual}} = [y_{a1} \ y_{a2} \ y_{a3} \ \dots \ y_{an}].$$

CASE I: $V_1 > 0$ (go forward)

$$u_L = u_i \cdot (Pw_{\max}/u_{\max})$$

$$u_R = u_i \cdot (Pw_{\max}/u_{\max})$$

Else if $w_1 > 0$ (Front Left)

$$\text{radius} = v_i/w_i$$

$$\text{radius}_L = \text{radius} - \text{width}/2$$

$$\text{radius}_R = \text{radius} + \text{width}/2$$

$$v_L = w_i \times \text{radius}_L \times Pw_{\max}/v_{\max}$$

$$v_R = w_i \times \text{radius}_R \times Pw_{\max}/v_{\max}$$

CASE II: $V_1 = 0$

If $W_1 = 0$: (stop)

$$u_L = u_R = 0$$

Else if $W_1 > 0$: (Left self-rotation)

$$u_L = -u_R$$

$$v_R = w_i \times \left(\frac{\text{radius}}{2}\right) \times \frac{Pw_{\max}}{v_{\max}}$$

Else if $W_1 < 0$: (Left self-rotation)

$$u_L = -u_R$$

$$v_R = w_i \times \left(\frac{\text{radius}}{2}\right) \times \frac{Pw_{\max}}{v_{\max}}$$

$$u_R = -u_L$$

ALGORITHM 1: Complete state-flow algorithm.

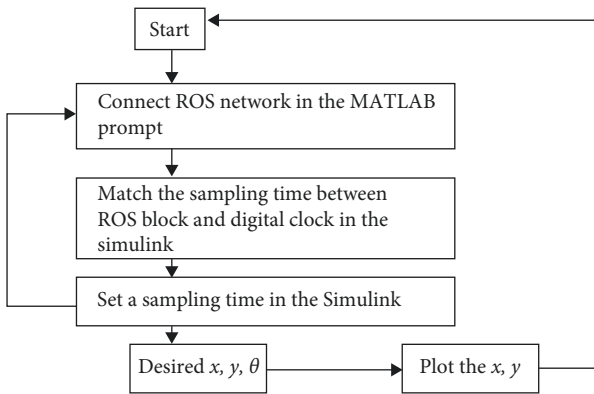


FIGURE 6: Indoor mobile robot control loop system.

data were saved using the MATLAB prompt. After the experiment, a comparison between the actual and reference trajectories of the state-flow algorithm was successfully performed.

The mobile robot navigation module was created to control all vehicle control activities and operations through the state-flow algorithm. The geometry-msgs/Twist ROS message format was used to create an ROS service that accepts the ROS topic input. The navigation module then processed the mobile robot to receive velocity commands as messages on the command/velocity topic information, which resulted in the desired controlled action of the mobile robot. The complete ROS topic and node structure were been developed according to reference [33]. The nodes were composed of two

parts: Linux system and MATLAB/Simulink. Figure 8 shows the Simulink-ROS connection to the Linux system used to control the mobile robot, and the mobile location package Lidar sensor provided an accurate estimate of the robot. The inertial measurement unit (IMU) sensor was replaced with a workspace block in Simulink. The connection between the mobile robot and the interface between each mobile robot component is described in this section. In Simulink block, the workspace was used to extract the Cartesian coordinates of the mobile robot. Till this point, the encoder data could be received by an upper computer in Simulink, and then the robot position, that is, Cartesian coordinates x and y , heading angle h , could be estimated using these data. We performed the experiment using a circular trajectory. Step 1: Type the IP command in the MATLAB prompt using the following structure `rosinit (0htt IP: == 192:168:0:xxx:113110; 0 Node-Host 0; 192.168:0:xxx:1770)`, as shown in Figure 8. The first IP address is a Linux notebook, and the second is the IP address of the computer on which the user uses Simulink. The upper and low-level computers were connected through the IP commands in the MATLAB prompt.

2.4. Autonomous Mobile Robot Hardware Systems and Sensors. The specific parameters for the mobile robot frame were a processing platform composed of NVIDIA, Jetson AGX Xavier, Linux Ubuntu 16.04 LTS ROS, and MATLAB 2023a. The ROS node diagram shows connection to the control system. A Lenovo-ThinkPad laptop equipped with a CPU Intel Core i5-6200 was used in the control system, clocked at 2.40 GHz to process the real-time MATLAB Simulink model, and

TABLE 1: Relationship between actual mobile robot's velocity v and linear and angular velocity signals (v, w) value.

Signal value in the Simulink	Actual mobile robot velocity	Radius of the circle
$v = 0.5, w = 0$	$v = 0.334$ m/s	N/A
$v = 1, w = 0$	$v = 0.668$ m/s	N/A
$v = 1.5, w = 0$	Not satisfied ($0 < V < 1$)	N/A
$v = 0.5, w = 2.2$	$v = 0.334$ m/s	R
$v = 1, w = 4.4$	$v = 0.668$ m/s	R
$v = 1, w = 2.2$	$v = 0.668$ m/s	2R

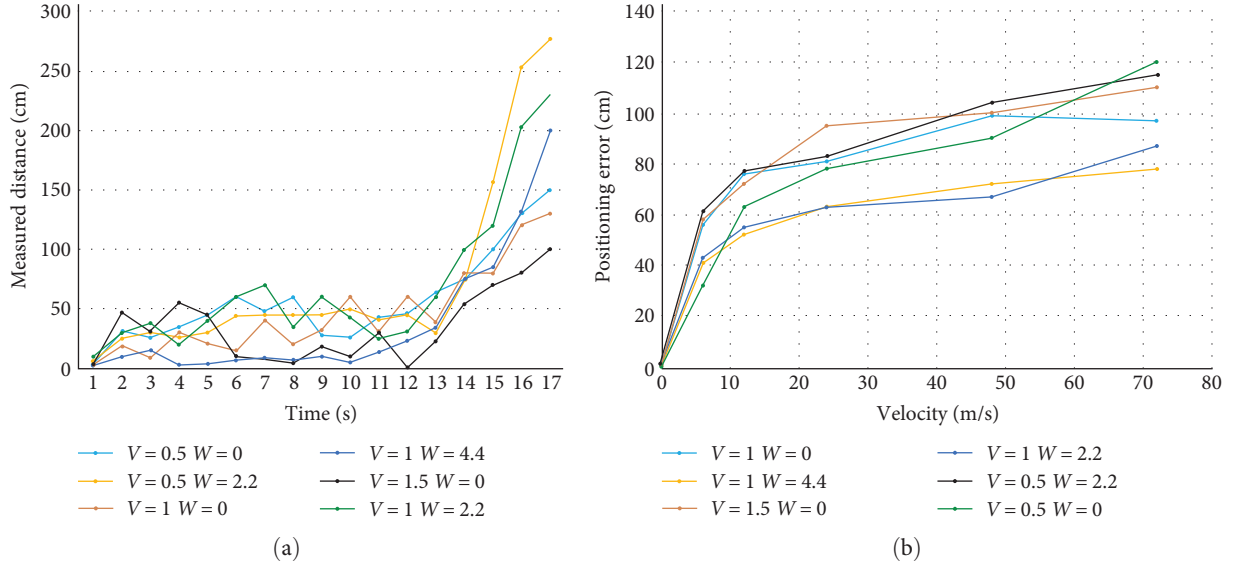


FIGURE 9: (a) Signal value in the Simulink of distance measurement and (b) actual mobile robot velocity of positioning.

hardware is described in Figure 7. During platform operation, the host computer adopted a Brushed DC Roboteq HDC 2460 motor microcomputer and sent the PWM signals amplified by the motor driver to the drive module. Finally, the right and left encoders sent feedback information from the two motors, as shown in Figure 7. An actual indoor environment was developed for the experimental setup. It had dimensions of $15.5 \text{ m} \times 6.5 \text{ m}$ and it showed an overview of the indoor environment consisting of an actual four-floor setup. The main research aim of this study was to track the predetermined path previously set by a designer using the state-flow block diagram in the MATLAB/Simulink. To evaluate the proposed state-flow algorithm for tracking the linear and angular velocities of a mobile robot on a predetermined trajectory, a series of desktop-based mobile platform experiments was conducted using MATLAB/Simulink, and an ROS connection to the mobile platform.

3. Results and Discussion

3.1. Setup State-Flow Algorithm. For the experiment, we must determine the relationship between the linear velocity signal and the actual mobile robot velocity. Table 1 shows the relationship shows that the actual mobile robot velocity is 0.668 m/s by inputting the signal value $v = 1, w = 0$ in the Simulink

model. In a greenhouse environment, 0.668 m/s is a sufficient velocity for automatic driving. Seder et al. [20] conducted several indoor mobile robot experiments using the scalar value, and the maximum velocity was 0.6 m/s. The experiments were divided into three parts: stationary obstacle avoidance, and a path-following experiment to prove the left- and right-turn operation of the mobile robot. We also determined the ratio between the linear and angular signal values to maintain the radius of the circle at the three tile lengths. Figure 7(b) shows that the above error dynamics can be determined as the difference between the desired trajectory of the user set and the actual trajectory of the mobile robot. The constraint on the linear velocity can be determined through various experiments, $0 < V < 1$ and $0 < W < 4.4$ (Table 1).

Figures 9(a) and 9(b) shows the relationship between linear and angular velocities signals and real in-wheel motor and steering motor's velocities.

The user sets the time, linear and angular velocities before the experiment. After setting the optimal parameters, the user implements the Simulink model, and then the mobile robot begins to move. The adjusted signal can be sent to the mobile robot's Linux computer program using ROS to publish a block, and the ROS subscription block can send the x and y position of the mobile robot, as shown in Figures 10 and 11. Furthermore, the workspace block can send the x and y

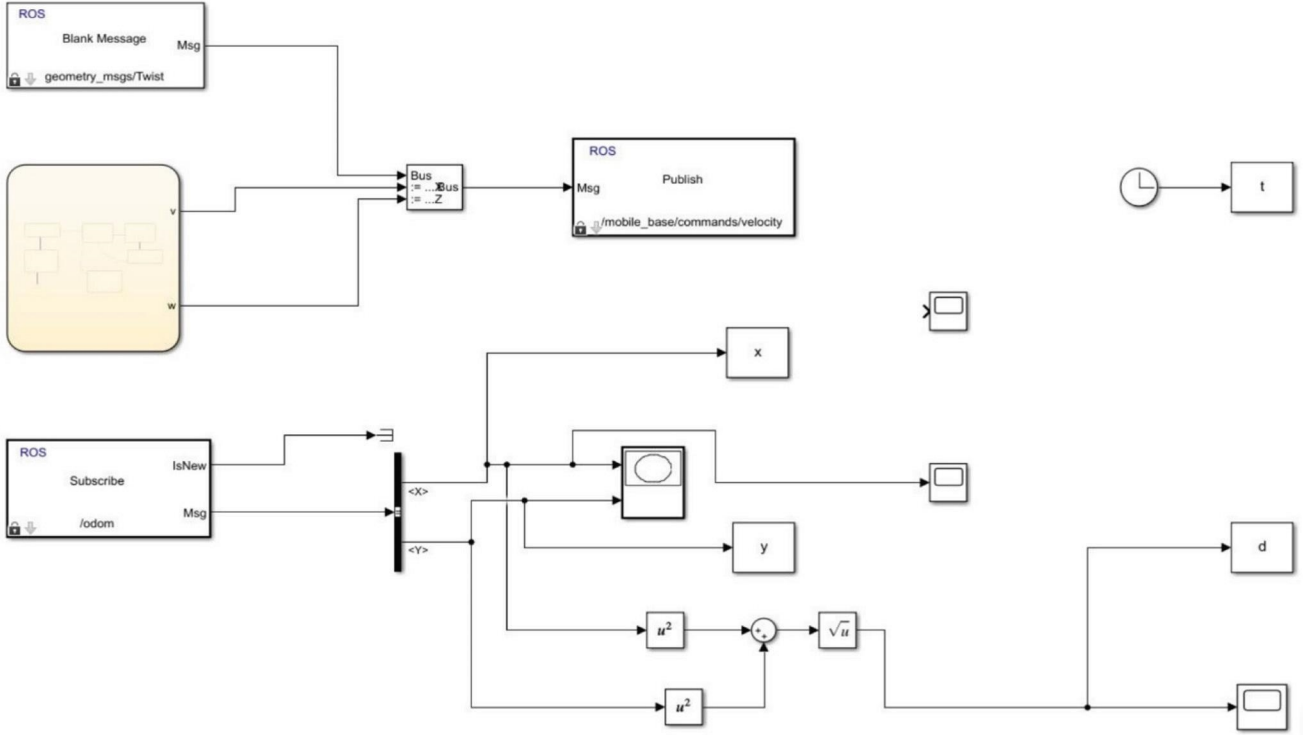


FIGURE 10: Overall structure of the State-flow method with ROS publish-and-subscribe block connection.

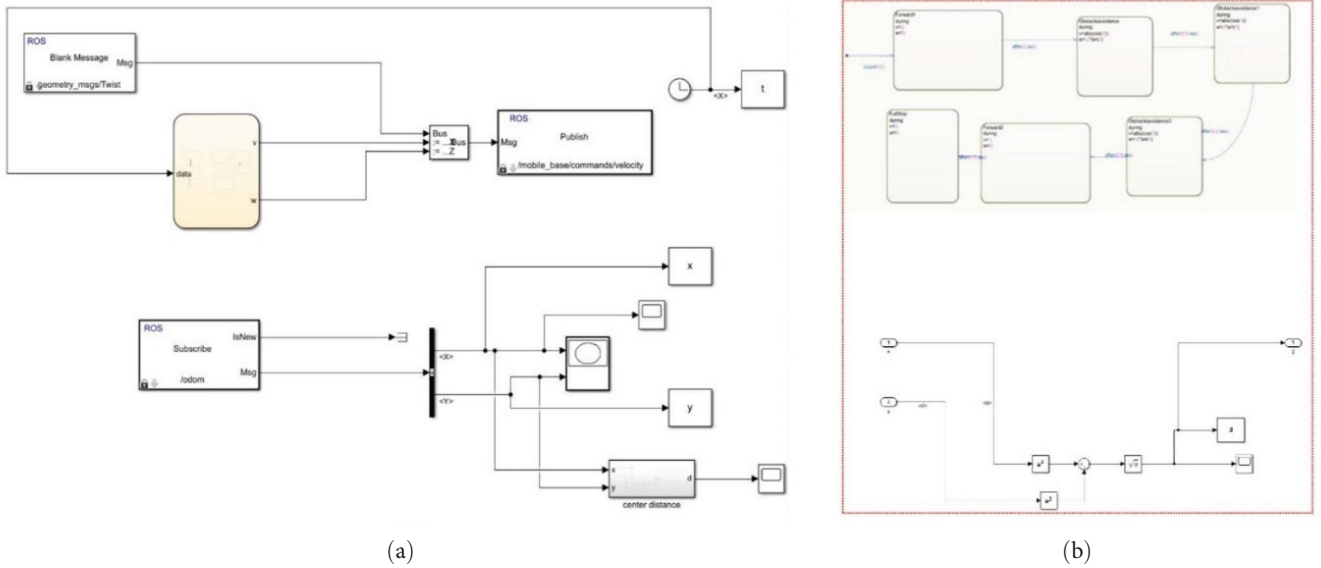


FIGURE 11: (a) Overall structure of connection for state-flow and ROS Publish block and (b) the description of the obstacle avoidance block.

coordinates of the mobile robot to the MATLAB prompt. Finally, the user can plot the figure after the experiment.

3.2. Trajectory Tracking Test 1 with State-Flow Design: Stationary Obstacle Avoidance. In Case I, only the linear velocity variables are considered as input variables, and

thus the corresponding path following is indicated as a straight line. In Case II, only the linear angular variables are negative values, which means that the mobile platform must move backward. These are simplistic scenarios in which the platform has no persistent existing problems such as left and right turns. However, there are general Cases III and IV

TABLE 2: Trajectory tracking test 1 with State-flow design for pipe line test.

Forward 1	During $v = 1, w = 2$	Time set = 1 s
1 Straight	During $v = 0.2, w = 0$	Time set = 10 s
2 Turn left	During $v = 0.2, w = 0$	Time set = 10 s
Pipe line 3 straight	During $v = 0.2, w = 0$	Time set = 15 s
Pipe line 4 backward	During $v = -0.2, w = 0$	Time set = 15 s
5 Rightward	During $v = 0.2, w = -1$	Time set = 10 s

TABLE 3: Trajectory tracking test 2 with State-flow design for obstacle-avoidance mobile robot.

Forward 1	During $v = 1, w = 2$	Time set = 1 s
Obstacle avoidance 1 turn right	During $v = \cos(1), w = -2 \times \sin(1)$	Time set = 1 s
Obstacle avoidance 2 turn left	During $v = \cos(1), w = 2 \times \sin(1)$	Time set = 1.5 s
Obstacle avoidance 1 turn left	During $v = \cos(1), w = 2 \times \sin(1)$	Time set = 1.5 s
Obstacle avoidance 1 turn right	During $v = \cos(1), w = -2 \times \sin(1)$	Time set = 1.5 s

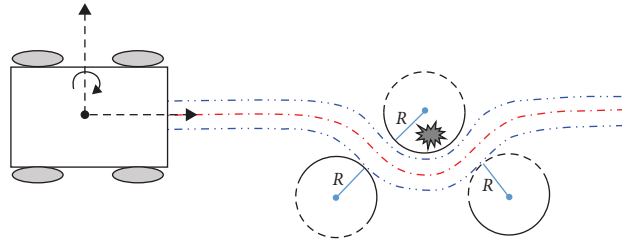


FIGURE 12: Internal components of the experimental platform; the kinematics model was design for obstacle avoidance.

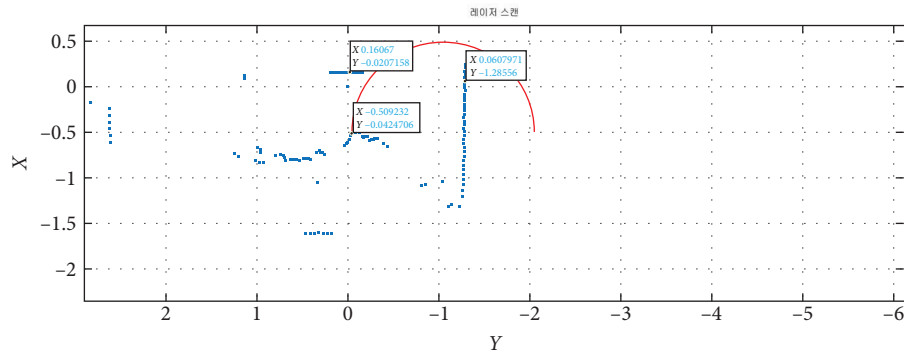


FIGURE 13: Obstacle avoidance experiment using the predetermined path by the state-flow.

in which angular variables are present, which guarantee that the mobile platform can avoid stationary obstacles within the Lidar map using the left and right turns of the mobile robot. The overall structure of the obstacle avoidance scheme is shown in Figures 11(a) and 11(b).

We set the following parameters to control the mobile robot's, state-flow production discribed as Table 2. The time-series angular velocity and linear velocity signals were generated within the state-flow block, based on a time set in the condition. Furthermore, the parameter for obstacle avoidance test 2 is given as Table 3.

The obstacle avoidance block is described as follows: the state-flow block above is built in a sequential structure owing to the change of the initial coordinates problem, which is determined by heuristic experiments from the 4th block to 5th block. The radius of the circle relevant to the right and left turns was determined only by the controlled variable, the linear velocity v , based on the kinematic equation of the bicycle-type model described in Figure 12.

The following Figure 13 is datasets achived after experimenting avoidance of the stationary obstacle. The kinematic equation is as follows:

$$x(t) = v \sin(\theta(t)), \quad (2)$$

$$y(t) = w \cos(\theta(t)). \quad (3)$$

For Cases II and III, if $v=w$, the above equation has implications for the following logic:

$$x^2(t) + y^2(t) = v^2. \quad (4)$$

The above Equation (4) is a circle with radius v .

3.3. Trajectory Tracking Test 1 with State-Flow Design: Stationary Obstacle Avoidance. An actual indoor environment was used in this study. It had a dimension of 15.5 m \times 6.5 m; the 3D simultaneous localization and mapping (SLAM) was used for localization and the Lidar measurements. Figure 14(a) shows the indoor environment of 3D SLAM, which represents Lidar mapping and odometry for mobile robot localization; the left image shows the map and trajectory estimated using the SLAM algorithm, and the right image shows the corresponding feature points. The input signals v and w were discretized signals. At a specific sampling time, the signal was captured, and then held until a certain time passed, which implied a zeroth-order holder function in Simulink. The Lidar intensity for mobile robot localization estimation is shown in Figure 14(b). The shape of the input signal is shown in Figure 14(c), which shows the estimation path and ground truth obtained using odometry and Lidar mapping. The blue path depicts the actual path of the robot, and the red path depicts the corresponding ground truth values. The difference between the actual and reference paths is indicated as root-mean-square error (RMSE). Figure 14(d) shows the estimated heading angle of the indoor mobile robot along the same path, that of the mobile robot localization.

3.4. Navigation Result of the Radius of the Circle. The state-flow method was applied to an actual mobile robot in an indoor environment. First, the poles were placed at three tiles lengths from the pillar in the indoor environment. After the poles were set, the map was scanned via 16 channel Velodyne Lidar to calculate the radius of the mobile robot's left and right turns, and the process is shown in Figures 15(a) and 15(b). The above logic implies that if w exists, it is a circle with a radius of linear velocity v . The signs of the angular velocity w , determines the direction of the circle. If w has a $(+)$ sign, the circular direction is clockwise. If w has a $(-)$ signs, then the circular direction is counterclockwise, as shown in Figure 15. Using the dynamic window approach (DWA), the signal was disrupted as described [20]. However, the state-flow method produced a constant signal, which implied that smooth trajectory tracking and accurate path-following were implemented, as shown in Figure 16(c). This indicated that the mobile platform continued moving forward or backward in the $x-y$ direction depending on the signs of linear velocity denoted by v . To show that a linear

and angular velocity signal was set for the radius of the circle, we conducted an experiment with $v = 0.5$, $\omega = 2.2$ and $v = 1$, $\omega = 2.2$, and the radius was $\sqrt{2}$ times larger than the previous one. Theoretically, the radius of the circle was $\sqrt{2}$ times by increasing the linear velocity signal twice based on Equation (5). To calculate the radius of the circle, we set a pole at point that indicated the distance from the pillar point. After scanning the 2D Lidar map, we calculated the reference path based on the radius of the first and second quarter-circles. Figure 16(d) shows the estimated path and heading angle of the mobile robot when it moved along semicircular paths when it turned right and left at right angles.

3.5. Navigation Result of the Path Following in the Narrow Environment. The reference trajectory was given by $v_r = 0.2$, $w_r = 0.2$, and $t = 10(s)$. This concerns the kinematic trajectory controller, which is not relevant to the state-flow. In addition to the mathematical review, we conducted an experiment using a digital clock given the condition linear and angular velocities condition of v in $[v_{\min}, v_{\max}]$, w in $[0, w_{\max}]$ in Equation (5). Equations (6) and (7) shows that the mobile robot reference trajectory $[X_r(t), Y_r(t), \theta_r(t)]$, can be determined based on the given angular and linear velocities $[v_r(t)]$, $[w_r(t)]$, and the poles can be set to calculate the radius of the quarter circle, which are related to the left and right turn of the mobile robot (Figure 15). Distance of the radius of the circle in the Figure 16(c) is automatically obtained from the ROS odometry block in the Simulink which matches the real distance in the Lidar map. Before the experiment, we set two poles, indicated as a desired trajectory, which can be scanned as three points using 16 channel Velodyne Lidar. Using two poles, the generated path is described in Figure 17 in the narrow indoor environment. In detail, when the mobile robot moves in a semicircle from one point to another, the radius of the circle in Figure 18(c) can be obtained by calculating the distance between first, last points and center point, previously scanned in the Lidar map, Figure 16(b). This is because the distance between the points of the Lidar map reflects the measured value in the real environment. Furthermore, the actual path signal described in Figures 16(c) and 18(c) can be read without noise, better than other waypoint method described [34].

Kinematics and odometry using MATLAB for the mobile robot, local planner, and SLAM algorithm 3D mapping to understand the environment in which the mobile robot and global planner are located are shown in the Figure 18(a). Figure 18(b) shows the Lidar intensity used to estimate mobile robot localization in different environments. After implementing the Simulink desktop-based mobile robot control simulation, the red line representing the ground truth path was plotted and compared with the blue line representing the reference path in the Figure 18(c), which shows the reference path previously generated based on the coordinates within the Lidar map. The motion equation is given by Equation (1). The tracking error dynamics is expressed in Figures 19(a) and 19(b).

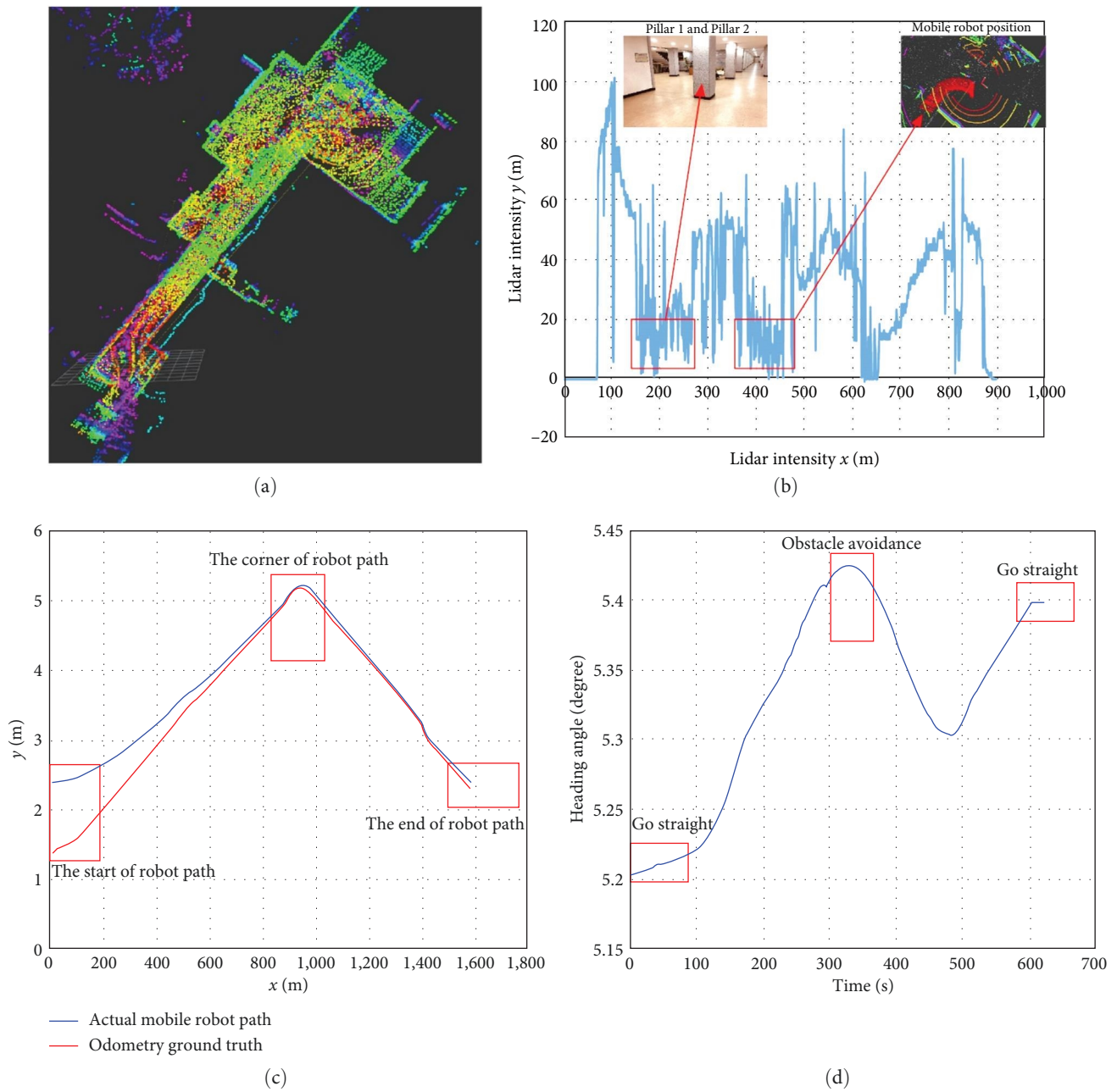


FIGURE 14: (a) 3D SLAM for mobile robot localization; (b) Lidar intensity of SLAM; (c) estimated mobile robot path and ground truth using odometry; (d) mobile robot estimated heading angle.

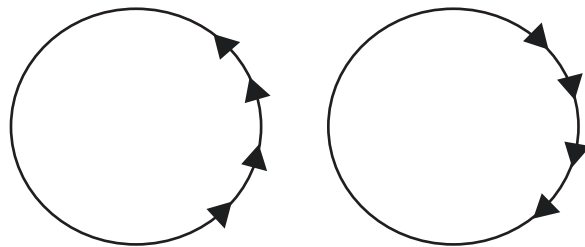


FIGURE 15: Internal components of the experimental platform depending on the signs of the angular velocity w .

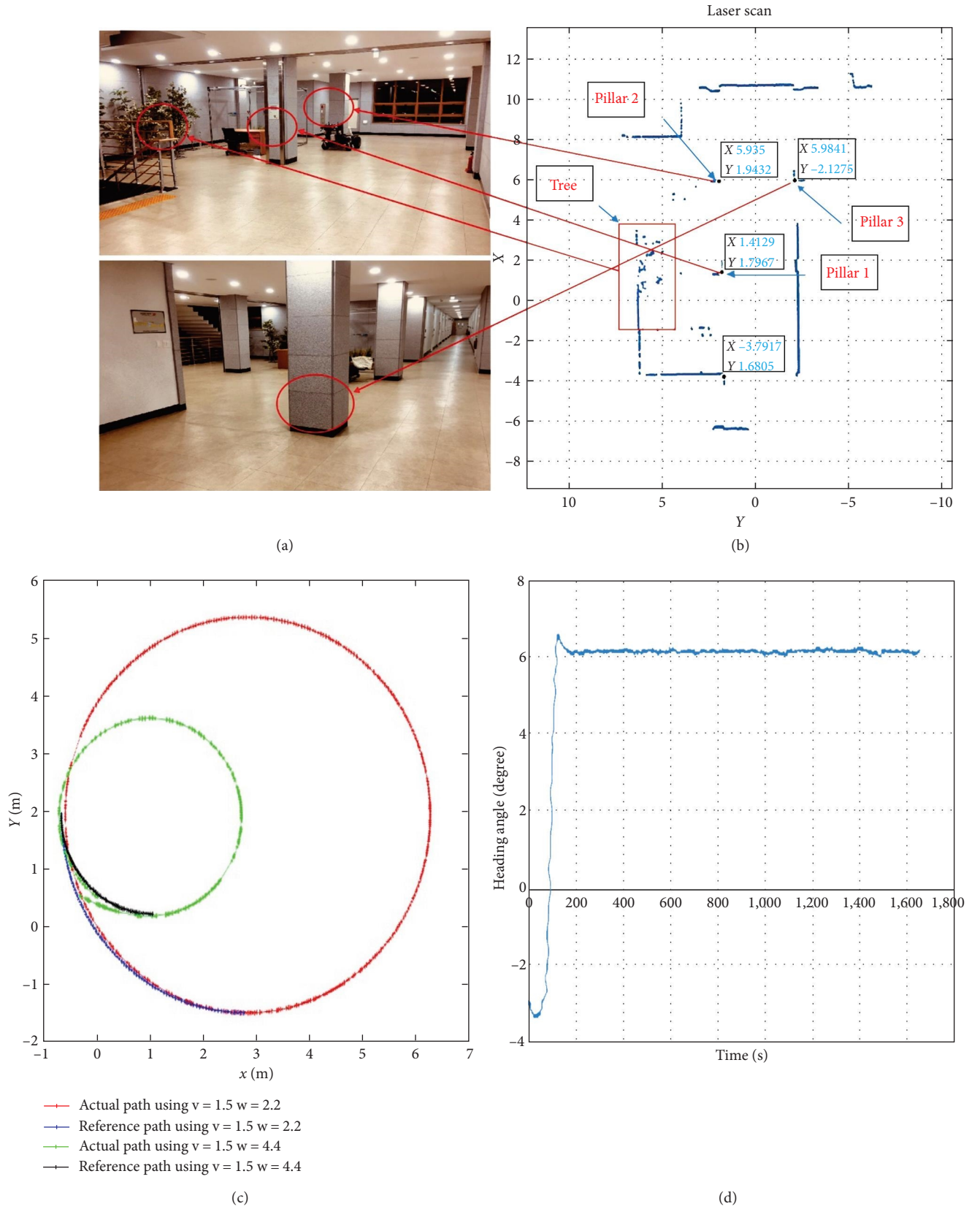


FIGURE 16: (a) Poles to the three tiles length from the pillar in the indoor environment; (b) 2D SLAM Lidar map for mobile robot estimation localization; (c) parameters for controlling the mobile robot's circular movement; (d) Estimated heading angle of the mobile robot's circle experiment.

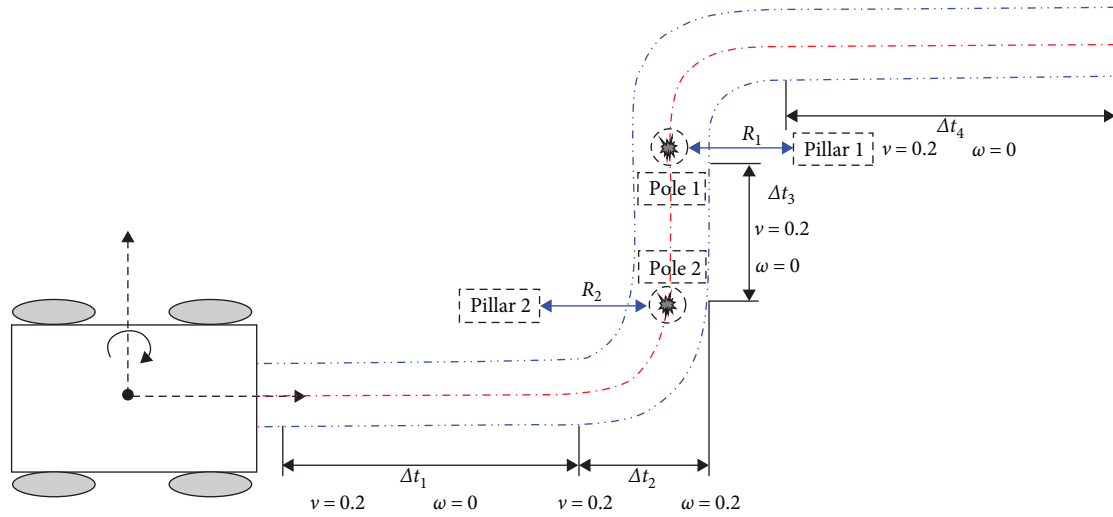
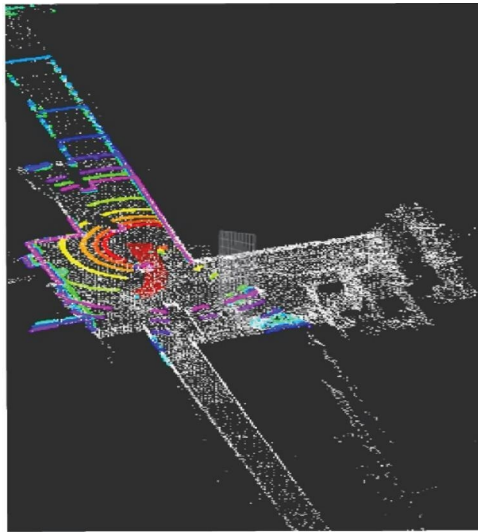
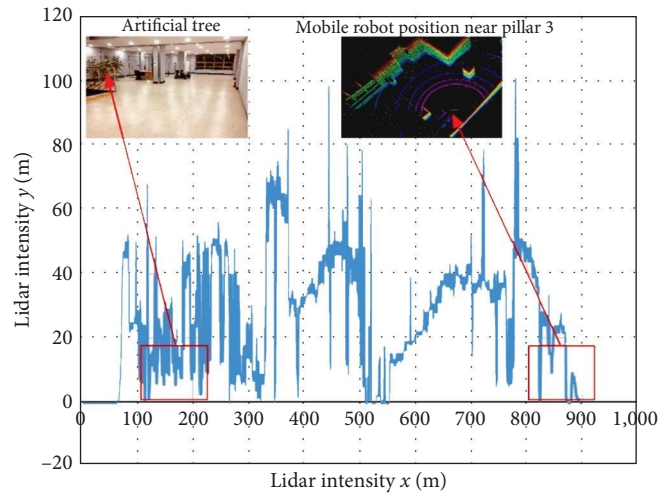


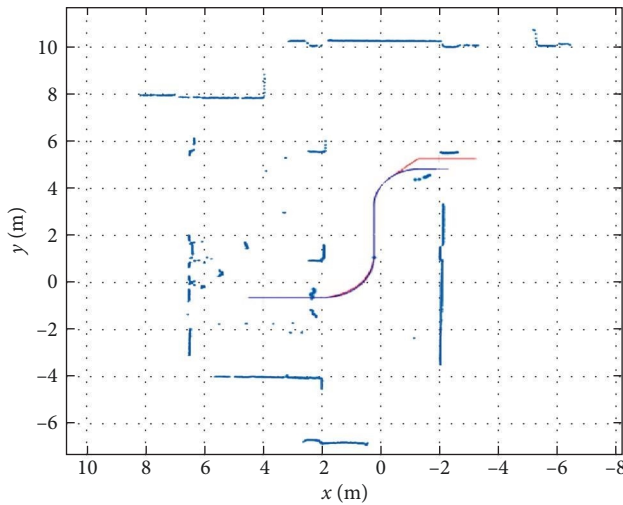
FIGURE 17: Reference trajectory $x_r(t)$, $y_r(t)$ and $\theta_r(t)$ plotted using MATLAB prompt quarter circle command given the reference and straight line plot.



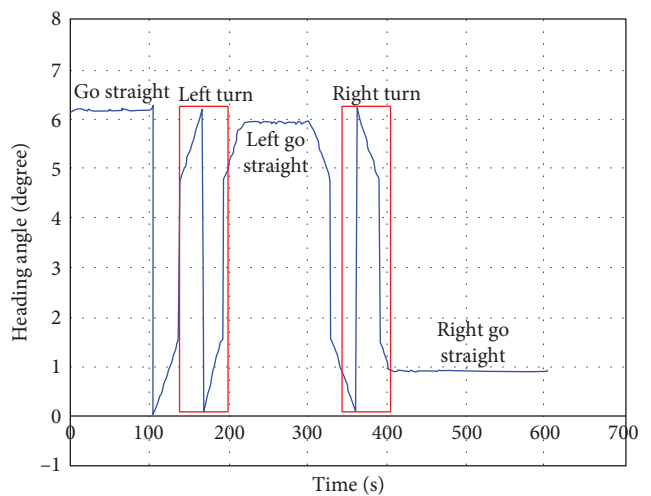
(a)



(b)



(c)



(d)

FIGURE 18: (a) 3D SLAM for mobile robot localization; (b) Lidar intensity of the SLAM; (c) estimated mobile robot path; (d) estimated the mobile robot path and heading angle.

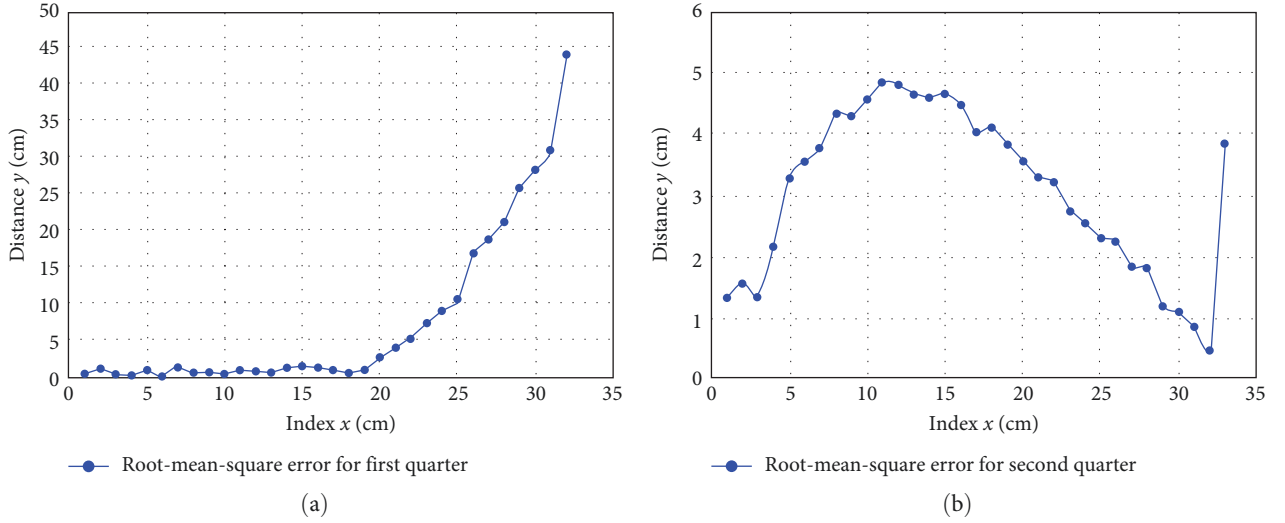


FIGURE 19: (a) First estimated path following of the indoor mobile robot's quarter-circle left turn and (b) the second estimated path following of the indoor mobile robot's quarter-circle right turn.

TABLE 4: RMSE between mean and max values of mobile robot localization.

RMSE of D_E (cm)	Left turn	Right turn	Straight line
Mean value	3.0853	7.4028	0.2024
Max value	4.8576	43.8816	0.5321

$$\begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r(t) - x(t) \\ y_r(t) - y(t) \\ \theta_r(t) - \theta(t) \end{bmatrix}, \quad (5)$$

$$v = v_r + \frac{c_1 x_e}{\sqrt{1 + x_e^2 + y_e^2}}, \quad (6)$$

$$w = w_r + \frac{c_2 v_r (y_e \cos(\frac{\theta_e}{2}) - x_e \sin(\frac{\theta_e}{2}))}{\sqrt{1 + x_e^2 + y_e^2}} + c_3 \sin\left(\frac{\theta_e}{2}\right). \quad (7)$$

The above control law holds on the condition that the controlled input are bounded to $v \in [v_{\min}, v_{\max}]$, $w \in [w_{\min}, w_{\max}]$. The coefficients C_1 , C_2 , and C_3 are positive constants that can be adjusted in response to the velocity constraints. At first, the linear and angular signal, $v(t)$, $w(t)$, were sent in the form of an open loop to extract the actual path, $x(t)$, $y(t)$, and then the error dynamic, $x_e(t)$, $y_e(t)$, from the reference path, $x_r(t)$, $y_r(t)$, could be obtained. In the second closed loop experiment, the position controlled v and w are set in the state-flow block, then we experiment based on the parameter calculated from Equations (6) and (7). The aforementioned process is described in Figure 5.

Table 3 shows the states in which the above state-flow method can achieve the required accuracy in an indoor environment, and summarize the RMSEs of the position error based on MATLAB/Simulink, with semicircle turns in one run for two different turning paths. Figures 16(a) and 16(b) are the surrounding environment during experiment scanned

from Velodyne 16 channel Lidar. Figure 19(a) shows that the first position error in the quarter-circle of the rows from this table was less than 5 cm, owing to the high slippage of the rollers in the headland. Based on the obtained results, the state-flow method can be applied to the indoor mobile robot path-following problem. Figure 19(b) shows that the second quarter-circle right-turn problem had a significant RMSE for the end points, because the ROS publish block was calculated from the IMU sensor, in that case, significant drift occurred, as described in [22]. However, in the actual world, we observed that the mobile platform followed the imaginary path within a 5 cm RMSE error. the reference and actual path described in Figure 18(c) is automatically obtained at ROS odometry block in MATLAB/Simulink during experiment. Figure 18(d) is corresponding heading angle during experiment in the indoor environment. The initial point in the Lidar map was found and the initial point of the actual path was moved to the mobile robot's starting point indicated in the Lidar map. By placing the reference and actual path on the Lidar map, they show how the robot moved in real environment. RMSE during experiment is given in Table 4.

4. Conclusions

In this study, the state-flow method was applied to an actual mobile robot path following an the imaginary path. First, the predetermined path was set, and then the sampling time and, linear and angular velocities were calculated using the linear and angular constraints determined by various experiments. Furthermore, the reference path was created within the Lidar map by setting multiple poles, and the RMSE was then

calculated based on the poles and pillar coordinates of the Lidar map, proving that smooth trajectory tracking can be implemented via the state-flow method. State-flow path following has advantages over waypoint path following such as pure pursuit, which implies vibration is damped and smooth following is possible. We demonstrated that the latitude x , longitude y , and heading angle h is smoothly path followed.

Data Availability

The data can be available to the author, hence ask a datasets to 622061@chungbuk.ac.kr.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All listed authors must have made a significant scientific contribution to the research in the manuscript approved its claims and agreed to be an authors.

Acknowledgments

This work was partly supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2024-2020-0-01462, 50%) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2020R1A6A1A12047945, 50%).

References

- [1] D. H. Kang, S. Y. Lee, J. K. Kim, M. J. Park, J. K. Son, and S.-W. Yun, "Development of an automatic grafting robot for fruit vegetables using image recognition," *Protected horticulture and Plant Factory*, vol. 28, no. 4, pp. 322–327, 2019.
- [2] K. Sabanci, "Image processing based precision spraying robot," *Tarım Bilimleri Dergisi*, vol. 20, no. 4, pp. 406–414, 2014.
- [3] T. Bak and H. Jakobsen, "Agricultural robotic platform with four wheel steering for weed detection," *Biosystems Engineering*, vol. 87, no. 2, pp. 125–136, 2004.
- [4] K. H. Choi, S. K. Han, S. H. Han, K.-H. Park, K.-S. Kim, and S. Kim, "Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields," *Computers and Electronics in Agriculture*, vol. 113, pp. 266–274, 2015.
- [5] A. Khalifa, O. Kermorgant, S. Dominguez, and P. Martinet, "Platooning of car-like vehicles in urban environments: longitudinal control considering actuator dynamics, time delays, and limited communication capabilities," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2670–2677, 2021.
- [6] M. Samuel, M. Hussein, and M. Binti, "A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle," *International Journal of Computer Applications*, vol. 135, no. 1, pp. 35–38, 2016.
- [7] X. Wang, F. Wang, and W. Wei, "Linear active disturbance rejection control of dissolved oxygen concentration based on benchmark simulation model number 1," *Mathematical Problems in Engineering*, vol. 2015, Article ID 178953, 9 pages, 2015.
- [8] Zhong-Ping Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 265–279, 1999.
- [9] B. Åstrand and A. J. Baerveldt, "An agricultural mobile robot with vision-based perception for mechanical weed control," *Autonomous Robots*, vol. 13, no. 3378, 2002.
- [10] H. S. Song, "Capital structure determinants an empirical study of swedish companies," in *CESIS Working Paper Series in Economics and Institutions of Innovation*, p. 25, 2005.
- [11] O. C. Barawid Jr., A. Mizushima, K. Ishii, and N. Noguchi, "Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application," *Biosystems Engineering*, vol. 96, no. 2, pp. 139–149, 2007.
- [12] L. S. Guo and L. Feng, "A low-cost integrated positioning system of GPS and inertial sensors for autonomous agricultural vehicles," in *ASAE Annual Meeting (p. 1)*, American Society of Agricultural and Biological Engineers, No. 033112, 2003.
- [13] T. Bakker, K. van Asselt, J. Bontsema, J. Müller, and G. van Straten, "Autonomous navigation using a robot platform in a sugar beet field," *Biosystems Engineering*, vol. 109, no. 4, pp. 357–368, 2011.
- [14] R. Earl, G. Thomas, and B. S. Blackmore, "The potential role of GIS in autonomous field operations," *Computers and Electronics in Agriculture*, vol. 25, no. 1–2, pp. 107–120, 2000.
- [15] J. Xue, L. Zhang, and T. E. Grift, "Variable field-of-view machine vision based row guidance of an agricultural robot," *Computers and Electronics in Agriculture*, vol. 84, pp. 85–91, 2012.
- [16] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3D LIDAR sensor," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 265–273, 2011.
- [17] T. Torii, "Research in autonomous agriculture vehicles in Japan," *Computers and Electronics in Agriculture*, vol. 25, no. 1–2, pp. 133–153, 2000.
- [18] E. Panteley and A. Loria, "On global uniform asymptotic stability of nonlinear time-varying systems in cascade," *Systems & Control Letters*, vol. 33, no. 2, pp. 131–138, 1998.
- [19] A. A. J. Lefeber, "Tracking control of nonlinear mechanical systems," in *Doctoral Dissertation*, The Netherlands, 2000.
- [20] M. Seder, K. Macek, and I. Petrovic, "An integrated approach to real-time mobile robot control in partially known indoor environments," in *31st Annual Conference of IEEE Industrial Electronics Society, IECON*, No. 8923423, 2005.
- [21] P. Polack, F. Altché, B. d'Andréa-Novet, and A. de La Fortelle, "The kinematic bicycle model: a consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, No. 17066205, 2017.
- [22] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020.
- [23] A. Ollero and G. Heredia, "Stability analysis of mobile robot path tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots*, IEEE, No. 5100108, 1995.
- [24] L. S. Guo, Q. Zhang, and S. Han, "Position estimate of off-road vehicles using a low-cost GPS and IMU," in *American Society of Agricultural and Biological Engineers*, No. 021157, 2002.
- [25] S. Li, G. Zhang, X. Lei, X. Yu, H. Qian, and Y. Xu, "Trajectory tracking control of a unicycle-type mobile robot with a new planning algorithm," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, No. 17651557, 2017.

- [26] W. Han, P. Yuan, D. Chen et al., "A novel tracking control method based on LADRC for autonomous mobile robot," in *IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, No. 15617195, 2015.
- [27] S. Miah, W. Gueaieb, D. Spinello, and K. Khan, "Neighboring optimal control for mobile robot trajectory tracking with range-limited sensors," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE, No. 15291488, 2015.
- [28] B. Liu and P. Li, "A stateflow based simulation of UAV multi-mode flight control," in *Proceedings of the 8th World Congress on Intelligent Control and Automation*, No. 11511884, 2010.
- [29] M. P. Bakht, Z. Salam, A. R. Bhatti, W. Anjum, S. A. Khalid, and N. Khan, "Stateflow-based energy management strategy for hybrid energy system to mitigate load shedding," *Applied Science*, vol. 11, no. 10, 2021.
- [30] M. H. Anisi, G. Abdul-Salaam, and A. H. Abdullah, "A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture," *Precision Agriculture*, vol. 16, no. 2, pp. 216–238, 2015.
- [31] A. S. Thair and Z. N. Mustafa, "Intelligent surveillance robot," in *The First International Conference of Electrical, Computer, Power and Control Engineering ICECCPCE'13*, pp. 17-18, 2013.
- [32] P. Jordi, A. S. Jose, V. Ignasi, and C. Xavi, "Buiding a mobile robot for floor-cleaning operation in domestic environment," *IEEE Transactions on Instrument and Measurement*, vol. 53, no. 5, 2004.
- [33] Y. Heonjong, L. Kyeonghwan, and C. Tean, "Development of the path following platform for a unicycle-type mobile robot in the indoor environment based on state flow method," *Korean Society of Agricultural Machine*, 2020.
- [34] M. S. Jarrod, "Automatic steering methods for autonomous automobile path tracking," Doctoral Dissertation, USA, 2009.