LiDAR Based Trajectory-Tracking of an Autonomous Differential Drive Mobile Robot Using Fuzzy Sliding Mode Controller

HSIU-MING WU[1], (Member, IEEE), AND MUHAMMAD QOMARUZ ZAMAN[2]
[1]Department of Intelligent Automation Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Graduate Institute of Manufacturing Technology, National Taipei University of Technology, Taipei 10608, Taiwan

Corresponding author: Hsiu-Ming Wu (hmwu@mail.ntut.edu.tw)

ABSTRACT This study aims to achieve the trajectory-tracking of an autonomous differential drive mobile robot (ADDMR) in the presence of friction torques using the proposed Fuzzy Sliding-Mode Control (FSMC). First, the complete model of the ADDMR is established including kinematic, dynamic and actuator models. Then, the desired trajectory is planned and generated via Bézier curve combined with cubic time mapping. The localization of the ADDMR is performed using the Monte Carlo Localization (MCL) technique and LiDAR point cloud data. Subsequently, the proposed FSMC utilizes nonlinear sliding surfaces inferred based on fuzzy logic to asymptotically attain the pose convergence. To investigate the feasibility and robustness of the proposed FSMC, simulations and experiments are conducted under different operation conditions. The results show that the proposed FSMC using LiDAR data in the feedback leads to excellent control performance even in the presence of friction torques. It was also shown through simulations and experiments that the established ADDMR model fits the actual ADDMR model very well.

INDEX TERMS Autonomous differential drive mobile robot, fuzzy sliding-mode control, Monte Carlo localization, nonlinear sliding surface, Bézier curve.

I. INTRODUCTION

Online shopping has been drastically increasing over the past year, especially during COVID-19 pandemics [1], [2]. Factories are racing to equip their warehouses with automated logistic handling technologies [3], [4] or contactless delivery service [5], [6]. For these reasons, there has been numerous research on the control of mobile robots. The commonly used configurations are robots driven by wheels speed differences and they are controlled to follow a prescribed path from initial to final position in a determinate time. The type of robots driven by wheel velocity differences are called Differential Drive Mobile Robot. The challenge of controlling Autonomous Differential Drive Mobile Robots (ADDMRs) comes from its nonlinearity. Specifically, its nonholonomic property constrains its side movements. Many control techniques are proposed to handle it, but the important issue is that most of them only consider kinematics [7]–[22]. It is understandable because the consideration of its dynamic model will increase the complexity and dimensionality of the model

which leads to difficulties for a controller design standpoint. However, kinematics does not consider mass and when an ADDMR is driven at high speed, its dynamic model becomes a necessity in addition to the kinematic model. Thus, some controller design techniques have been taking the dynamic model of the ADDMR into consideration. For example, the works in [23]–[34] considered both kinematics and dynamics for their vehicle models.

A significant task for the above-mentioned works is trajectory tracking in which the robot must follow a pre-defined path. The studies in [24] and [28] have positioned their ADDMRs on the path initially. Keighobadi and Menhaj [24] stated that their proposed method could not fully compensate initial position outside the path. In [25]–[27], the results demonstrate tracking error convergence at arbitrary initial positions; however, the circular path just has a constant velocity reference. Xu *et al.* [27] showed that it is important to consider speed transition of the robot movement. In most previous works, the ADDMR was given a path without time constraints; therefore, the ADDMR approaches each point in the path incrementally. This is an important issue that the time arrangement along the planned

The associate editor coordinating the review of this manuscript and approving it for publication was Shihong Ding.

path can lead to speed variations that the ADDMR has to follow.

The most recent developments in trajectory-tracking using fuzzy logic [14]–[22], [24], [30], [33]–[35] and sliding mode control techniques [13], [29]–[32], [35]–[39] are the motivation behind the work presented herein, which takes advantage of findings from both methods. Therefore, the FSMC based on a LiDAR point cloud data is proposed here to control an ADDMR system to follow a trajectory at prescribed time from arbitrary initial positions under torque disturbances. The contributions of this work are as follows:

1) A comprehensive ADDMR model consists of the kinematic and dynamic models is presented here. The model is augmented with the actuator dynamics for a more realistic representation of the motion of the ADDMR which has not been addressed by Refs. [7]–[36].

2) A desired trajectory is planned using Bézier curve. The $n + 1$ way-points on the trajectory are determined using an nth-order Bézier curve. Unlike work in [27], the dynamic velocity command is embedded in the trajectory itself rather than in the control process. The advantage of the proposed approach is that it lends itself to possible combination with other control techniques in the closed-loop control design process. It also enables the trajectory to have zero velocity at the beginning and the end whereas previous works, such as those in [7]–[36] did not consider this. The close-loop performance of the ADDMR using the proposed control technique is validated experimentally using an 8th-order Bézier trajectory with cubic time mapping for the time arrangement along the path.

3) A fuzzy logic control design with 9 inference rules only combining a sliding mode control scheme leads to effective closed-loop control of the ADDMR. Fuzzy logic control offers multi-input, multi-output, model free controllers for trajectory tracking (see for example. [21] and [22]). In addition, the interpolative reasoning provides resilience against noisy measurement in the ADDMR system. However, unlike existing unlike existing techniques (e.g. [14]–[22], [30], [33]–[35]) where complex inference rules and membership functions are used, relatively simpler inference rules are used in the proposed control scheme. These lead to computational cost reduction and provide flexibility in the choice of computing devices used in the ADDMR testbed; a major advantage in the implementation phase.

4) LiDAR based Monte Carlo Localization (MCL) is used for feedback control to improve the control performance of the closed-loop system. The common approaches described in the literature (see for example [12], [29], [31]–[34]) use angular acceleration and torque feedback which renders the implementation a little more challenging. Instantaneous measurement of the orientation and position are easily accessible and

implementable on an actual testbed [22]. Therefore, a LiDAR is used here for pose feedback due to its accuracy compared to GPS or odometry sensors by Refs. [40]–[43].

5) Experiments and simulations are conducted to validate the theoretical developments. Both the simulations and experiments are conducted using different initial conditions which have not been done in previous works (see for example [10], [12], [13], [16], [17], [19]–[21], [23], [23], [25]–[33], [35], [36]).

The rest of the paper is arranged as follows: In Section 2, the system description, modelling, and problem formulation are given. In Section 3, the trajectory planning, pose measurement, and FSMC design are discussed. Discussions of the simulation and experiment results are presented in Section 4. Finally, concluding statements are made in Section 5.

## II. SYSTEM DESCRIPTION, MODELING, AND PROBLEM FORMULATION

The hardware configuration for the ADDMR experimental platform is described here for which the equations of motion will be derived. The system modeling is divided into 3 subsystems, i.e., kinematic, dynamic, and actuator models. The details of each subsystem are explained in the following subsections. The problem formulation for the ADDMR is subsequently delineated using the equations of motion of all subsystems.

A. SYSTEM DESCRIPTION

The experimental platform of the ADDMR and its diagram are shown in Fig. 1. The main CPU from ADLINK AmITX-SL-G with Intel i5-6500TE and 8GB Random Access Memory (RAM) is where the proposed controller is computed and all the data from the STM32F microcontroller is stored. Two A4950 DMOS Full-Bridge motor drivers generate voltages to actuate two DC motors responsible for the motion of the robot. The provided sensors are Quadrature hall effect encoder and Inertial Measurement Unit (IMU) used to measure the corresponding position and orientation of the ADDMR; a LiDAR is also provided for measurements of distance around the robot. The wheels are centered and with center of mass is halfway in between. In addition, two castor wheels are attached for the balance of the ADDMR.

B. SYSTEM MODELLING

The ADDMR is a type of mobile robots that the speed differences for two actuated wheels cause forward, backward, or turning movement. The illustration can be seen in the following Fig. 2. To establish geometric relationship of the ADDMR, an inertial frame $\{I\}$ and a body frame $\{B\}$ are given. Frame reference $\{I\}$ is attached to the floor and the origin of body frame $\{B\}$ is located on the center of mass of itself as described in Fig. 2. The orientation of the robot is θ with respect to the inertial frame which is calculated by taking angle measurement between the longitudinal axes of body frame and inertial frame. The linear and angular velocities of the ADDMR are calculated with respect to the point o,

(a) Experimental platform of the ADDMR system   (b) Hardware diagram
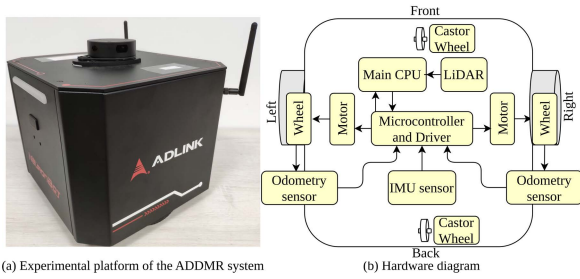
FIGURE 1. The ADDMR system used in the experiment.
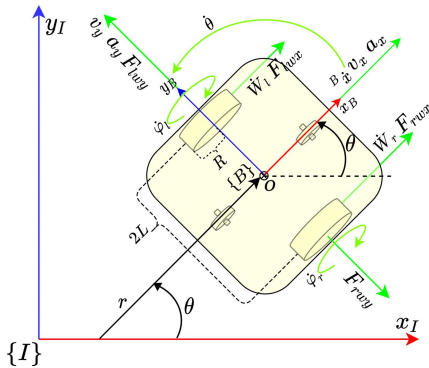
FIGURE 2. Frames and point of reference in the robot system.

i.e. the origin of body frame. Assume that there is a point of contact between each wheel and the floor, there exists resulting tangential velocities from their angular velocities $(\dot{\varphi}_r, \dot{\varphi}_l)$. The angular and linear velocities of the wheels are proportional to the ratio of the wheel radius R. Hence, the ADDMR's velocities with respect to the inertial frame can be obtained from its wheel angular rates [44]–[48] as

$$\begin{bmatrix} ^I\dot{x}_B \\ ^I\dot{y}_B \\ \dot{\theta} \end{bmatrix} = \frac{R}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \dfrac{1}{L} & -\dfrac{1}{L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} \qquad (1)$$

where θ is the orientation of the ADDMR. Moreover, the relationship between the velocity of mass $(^B\dot{x}, \dot{\theta})$ with respect to body frame and wheels' angular velocities are given by:

$$\begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & L \\ 1 & -L \end{bmatrix} \begin{bmatrix} ^B\dot{x} \\ \dot{\theta} \end{bmatrix} \qquad (2)$$

In addition to the kinematic model of Eqs. (1) and (2), the Newton-Euler approach is used to build up the dynamic model of the robot. The ADDMR is assumed to have a unified rigid body and the equations of motion are determined with respect to reference {B} as shown in Fig. 2. Moreover, two motors provide the ADDMR with longitudinal wheel forces (F_{lwx}, F_{rwx}) created by their corresponding torques (τ_l, τ_r) [45], [46], [49]. Hence, the linear and angular accelerations of the robot with respect to the body frame can be represented as

$$^B\ddot{x} = \frac{1}{MR}(\tau_l + \tau_r) \qquad (3)$$

$$\ddot{\theta} = \frac{L}{JR}(\tau_r - \tau_l) \qquad (4)$$

That means that the dynamic model of the ADDMR generates linear velocity $(^B\dot{x})$ and angular velocity $(\dot{\theta})$ after integration when motor torques τ_r and τ_l are applied.

Combining the aforementioned kinematic with dynamic models, the actuator model is derived to complete the system's equations of motion. However, designing the corresponding controller is very challenging due to the high dimensionality of the system's model. The actuated system is comprised of two identical DC motors and the relations of voltage and torque of the motor are formulated as follows:

$$\begin{cases} V_a = R_a i_a + L_a \dfrac{d\,i_a}{dt} + e_a \\ e_a = Kb\omega_m \\ \tau_m = K_t i_a \\ \tau = N\tau_m \end{cases} \qquad (5)$$

where R_a, L_a, V_a, i_a, and N are armature resistance, inductance, voltage, current, and gear ratio, respectively. The back emf e_a is proportional to the rotational velocity ω_m with back emf constant K_b. Moreover, the motor torque τ_m is proportional to the armature current i_a with constant K_t and the output torque τ is proportional to τ_m with gear ratio N. Based on Eq. (5), an actuator model can be obtained. In the subsystem, the two respective inputs are voltage V_{al} of the left motor and V_{ar} for the right motor. Here, τ_r and τ_l stand for torques of the left and right motors which are the outputs of the actuator model.

From the kinematic, dynamic, and actuator models, a comprehensive model is built for subsequently designing the trajectory-tracking controller. In addition, τ_{dr} and τ_{dl} which denote torque disturbances, are added to the dynamic model to investigate the robustness of the closed-loop control system when the proposed controller is applied.

C. PROBLEM FORMULATION
The objective of the proposed FSMC is to control the ADDMR from an arbitrary initial position along a desired trajectory in the presence of torque disturbances τ_{dr} and τ_{dl}. The pose information obtained by an MCL technique using LiDAR information to form nonlinear sliding surfaces. Then, the nonlinear sliding surfaces are reasoned to satisfy the reaching law (i.e., $\sigma\dot{\sigma} < 0$, where σ is a sliding surface) via the designed fuzzy inference rules. When control law is applied to the ADDMR system, the tracking errors asymptotically converge to zero. The overall control block diagram is shown in Fig. 3.

III. TRAJECTORY PLANNING, POSE MEASUREMENT, AND FSMC DESIGN
A. TRAJECTORY PLANNING
It is common in the robotics community to differentiate between path and trajectory. A path is a spatial function that connects an initial pose to a final one where as a trajectory is a path which is a function of time [50]. In this work, a path is a spatial function of arc length s where $s = 0$ is the initial pose and $s = 1$ is the final pose. One of the well-known ways
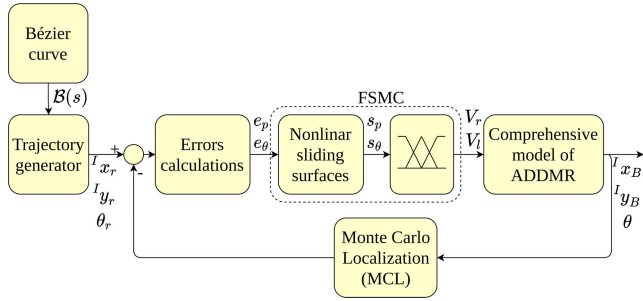
FIGURE 3. The overall control block diagram.

to generate paths is Bézier curves where the path is described using a parametric equation. The $n + 1$ parameter points P_i with $i = 0, 1, \ldots, n$ are used to form an nth-order Bézier curve $\mathcal{B}(s)$ as:

$$\mathcal{B}(s) = \sum_{i=0}^{n} P_i B_i^n(s), \quad s \in [0, 1] \tag{6}$$

where $B_i^n(s)$ is a Bernstein polynomial given by

$$B_i^n(s) = \binom{n}{i} s^i (1 - s)^{n-i}, \quad i = 0, 1, 2, \ldots, n \tag{7}$$

and $\binom{n}{i}$ is a polynomial coefficient given by

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{8}$$

Because a Bézier curve only goes through the first and the last parameter points, the other points are used to determine the shape of the curve. These points V_i are spaced points in an equally divided arc length such that

$$V_i = \mathcal{B}(i/n), \quad i = 0, 1, 2, \ldots, n \tag{9}$$

This yields a Bézier curve that will always go through all the V_i. The polynomial form of the Bézier curve in Eq. (6) can be expanded into a matrix form of

$$\begin{cases} A(s) = \begin{pmatrix} s^0 & s^1 & \ldots & s^n \end{pmatrix} \\ \quad \cdot \begin{pmatrix} B_0^n B_0^n(-1)^0 & 0 & \ldots & 0 \\ B_0^n B_1^n(-1)^1 & B_1^n B_0^{n-1}(-1)^0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ B_0^n B_n^n(-1)^n & B_1^n B_{n-1}^n(-1)^{n-1} & \ldots & B_n^n B_0^0(-1)^0 \end{pmatrix} \\ \mathcal{B}(s) = A(s) \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{pmatrix} \end{cases} \tag{10}$$

where $A(s)$ is an $1 \times (n+1)$ column vector that maps parameter points P_i to Bézier points at arc length s. The corresponding P_i to the designed V_i can be calculated by expanding Eq. (9)

as follows:

$$\begin{pmatrix} V_0 \\ V_1 \\ \vdots \\ V_n \end{pmatrix} = \begin{pmatrix} \mathcal{B}(0/n) \\ \mathcal{B}(1/n) \\ \vdots \\ \mathcal{B}(n/n) \end{pmatrix}$$

$$= \begin{pmatrix} A(0/n) \\ A(1/n) \\ \vdots \\ A(n/n) \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{pmatrix}$$

$$\begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_n \end{pmatrix} = \begin{pmatrix} A(0/n) \\ A(1/n) \\ \vdots \\ A(n/n) \end{pmatrix}^{-1} \begin{pmatrix} V_0 \\ V_1 \\ \vdots \\ V_n \end{pmatrix} \tag{11}$$

To form a smooth trajectory from the planned Bézier curve path, a time mapping function of the arc length $s(t)$ is designed using a cubic polynomial as follows:

$$\begin{cases} s(t) = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \\ \dot{s}(t) = 3 a_0 t^2 + 2 a_1 t + a_2 \end{cases} \tag{12}$$

where the coefficients a_i $i \in \{0, 1, 2, 3\}$ are calculated based on the intended initial and final conditions of s, and $\dot{s}$. By combining the path with a cubic time mapping, the desired trajectory is obtained for tracking control by the ADDMR system.

B. POSE MEASUREMENT

To localize and control the ADDMR, its pose measurement for feedback is required. In the experiment, in addition to odometry, MCL is used to improve the accuracy of the pose measurement via a LiDAR. The MCL technique is inspired by a recursive Bayes filter based on importance sampling to estimate probability of robot poses using sensory measurement data [51]. The states of the robot can be defined in a probabilistic approach with the help of two distribution models i.e. motion and observation models [52]. The motion model $p(x_t \mid x_{t-1}, u_t)$ represents the transition of probability distribution of state x at time t based on the previous state x_{t-1} and the control input u_t. The observation model $p(z_t \mid x_t)$ incorporates reading of sensors z_t to calculate the likelihood. The whole MCL algorithm is described in Table 1. The continuous belief of states $bel(x_t)$ in MCL is represented by a discrete number of M particles $\mathcal{X}_t = \{x_t^{(i)}, w_t^{(i)}\}_{i=1}^M$ [53]. The weight $w_t^{(i)}$ is approximately proportional to the probability of robot at states $x_t^{(i)}$. The initial belief $bel(x_0)$ is uniformly distributed. Hence, the initial weights w_0 are also uniform and the states x_0 are initially sampled from all possible state values with uniform distribution [54]. Line 4 in Table 1 propagates the particles using the system dynamics to generate hypothetical state x_t. This process represents a sampling from distribution $p(x_t \mid x_{t-1}, u_t)$. Line 5 in Table 1 calculates the likelihood representing measurement probability z_t from particle $x_t^{(i)}$, that is $p(z_t \mid x_t^{(i)})$. In this study, LiDAR beams are utilized

TABLE 1. Monte Carlo localization algorithm.

Monte Carlo Localization Algorithm
1:    **Input**: $\mathcal{X}_{t-1}, u_t, z_t$
2:    **Output**: $\mathcal{X}_t$
3:    **for** $i = 1$ **to** M **do**
4:        $x_t^{(i)} = \text{motionModel}(x_{t-1}^{(i)}, u_t)$
5:        $w_t^{(i)} = \text{observationModel}(z_t, x_t^{(i)}, i)$
6:        $\bar{\mathcal{X}}_t = \text{temporaryParticle}(x_t^{(i)}, w_t^{(i)}, i)$
7:    **endfor**
8:    draw new indices $j = \{1, 2, \ldots, M\}$ with probability $\propto w_t$
9:    **for** $j = 1$ **to** M **do**
10:      assign $x_t^{(j)}$ to $\mathcal{X}_t$
11:   **endfor**

to measure the likelihood of each particle. Line 6 in Table 1 assigns the states and likelihoods to a temporary particles $\bar{\mathcal{X}}$. Line 8 in Table 1 commonly referred to as re-sampling step in which $j \in [1, M]$ integers are drawn proportionally to the likelihoods w_t. The newly selected particles are assigned to the reference of j in Line 10. Particles with bigger likelihood are more likely to be drawn multiple times. Hence, more particles are distributed to the most probable states.

Here, a simulated example is displayed. The result shown in Fig. 4 indicates particles (blue dots) convergence history. The robot moves in a counterclockwise direction and the position is marked in green dashed line as a ground truth. The mean position of particles is taken as a predicted location and marked by an orange line. As seen from the figure, the particles gradually concentrate and converge to the real location of the robot.

C. FUZZY SLIDING-MODE CONTROL DESIGN

Based on the established comprehensive model and localization of the ADDMR, the proposed FSMC is designed to control an ADDMR to track the desired trajectory. The proposed FSMC has two inputs; i.e., heading error e_θ and position error e_p, and two outputs; i.e., V_{al} and V_{ar}. The details are further described in the following subsections.

1) NONLINEAR SLIDING SURFACES DESIGN

Prior to designing the FSMC, nonlinear sliding surfaces are first designed. The distance error e_p and heading error e_θ information are defined to construct nonlinear sliding surfaces. To clearly understand how both errors are calculated, a geometrical schematic shown in Fig. 5 describes the relation between the inertial frame $\{I\}$, robot frame $\{B\}$, and the planned Bézier trajectory $T = \mathcal{B}(s(t))$. Vector ${}^I\mathbf{P}_B$ and ${}^I\mathbf{P}_T$ are the robot position and trajectory position relative to the inertial frame. The trajectory position relative to the body frame is denoted by ${}^B\mathbf{P}_T$. In this way, it can be calculated by utilizing homogeneous transformation matrix ${}^I\mathbf{T}_B \in \mathbf{SE}(2)$

$$\begin{bmatrix} {}^B\mathbf{P}_T \\ 1 \end{bmatrix} = {}^I\mathbf{T}_B{}^{-1} \begin{bmatrix} {}^I\mathbf{P}_T \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & {}^I x_B \\ \sin\theta & \cos\theta & {}^I x_B \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} {}^I x_T \\ {}^I y_T \\ 1 \end{bmatrix} \quad (13)$$
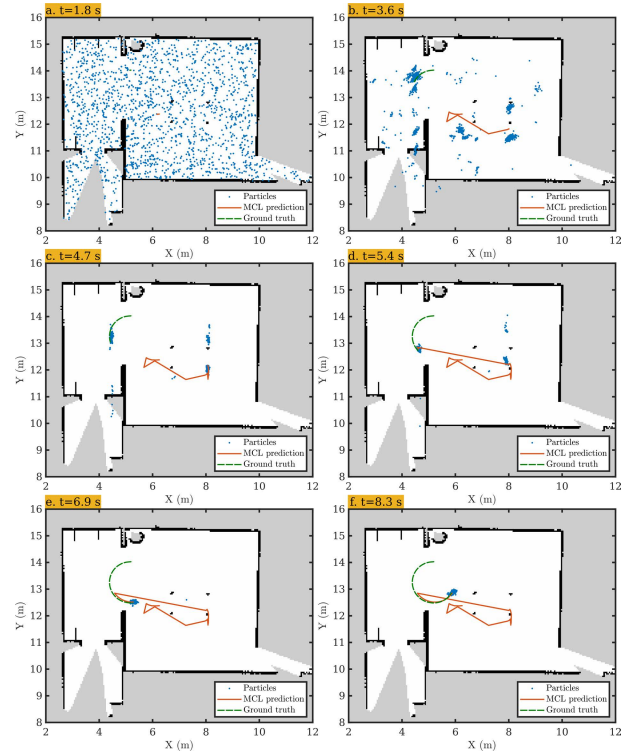
FIGURE 4. Particles convergence history in an occupancy grid map using MCL algorithm.
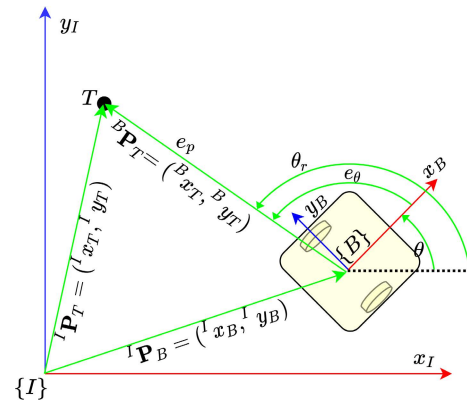
FIGURE 5. Geometric relation between the target and the robot.

and the position error is the magnitude of ${}^B\mathbf{P}_T$ as follows:

$$e_p = \sqrt{{}^B x_T{}^2 + {}^B y_T{}^2} \quad (14)$$

Using Eq. (13), the heading error e_θ is

$$e_\theta = \text{atan2}({}^B y_T, {}^B x_T) \quad (15)$$

where $\text{atan2}(\cdot)$ is a full-quadrant inverse tangent with output between $[-\pi, \pi]$ [55]–[58]. The function $\text{atan2}(\cdot)$ is preferred over the $\tan^{-1}(\cdot) = \text{atan}(\cdot)$ function for converting Cartesian coordinates to their corresponding polar coordinates.

The control objective is to make e_p and e_θ asymptotically converge to zero. Firstly, the following nonlinear sliding surfaces s_p and s_θ constructed by tracking errors are designed.

Here, the sliding surface s_p corresponds to trajectory tracking and s_θ corresponds to robot's heading tracking represented as

$$\begin{cases} s_p = \dot{e}_p + k_p e_p + f_p \mathrm{sign}(e_p) \\ s_\theta = \dot{e}_\theta + k_\theta e_\theta + f_\theta \mathrm{sign}(e_\theta) \end{cases} \quad (16)$$

where k_p, k_θ, f_p, and f_θ are positive constants and $\mathrm{sign}(\cdot)$ is a signum function [13], [15], [21], [26], [29], [31], [32], [35], [36]. Eq. (16) can be proved that once the sliding surfaces approach to zero, the corresponding tracking errors also approach to zero. Then, the fuzzy logic is proposed to regulate nonlinear sliding surfaces towards zero and the details are described in next subsection.

2) FUZZY INFERENCE LOGIC DESIGN

In fuzzy inference logic design, the sliding surfaces are inputs and the actuator voltages are outputs to control the left and right wheels. In the beginning, sliding surfaces are normalized using gains k_{np} and $k_{n\theta}$ yielding $\hat{s}_p = k_{np}s_p$ and $\hat{s}_\theta = k_{n\theta}s_\theta$. The normalized values are then evaluated using triangular fuzzy membership functions shown in Fig. 6. For each sliding surface, there are 3 membership functions comprising $\{N : Negative, Z : Zero, P : Positive\}$. In addition to the fuzzified inputs, this work also uses fuzzified voltage outputs for each actuator. The fuzzification of the outputs are the triangular type with 3 linguistic voltage representations $\{N : Negative, Z : Zero, P : Positive\}$ shown in Fig. 7. Then, fuzzy rules are designed in the inference engine as follows:

$$\begin{aligned} \mathcal{R}_1 : \ & if \ (\hat{s}_p = N \ AND \ \hat{s}_\theta = N) \quad then \ (V_{fr} = N, V_{fl} = P) \\ \mathcal{R}_2 : \ & if \ (\hat{s}_p = N \ AND \ \hat{s}_\theta = Z) \quad then \ (V_{fr} = P, V_{fl} = P) \\ & \vdots \qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ \mathcal{R}_9 : \ & if \ (\hat{s}_p = P \ AND \ \hat{s}_\theta = P) \quad then \ (V_{fr} = N, V_{fl} = P) \end{aligned}$$
$$(17)$$

The t-norm *AND* operator in this work is $MIN(\cdot)$ function. The consequence of every rule is aggregated using t-conorm $MAX(\cdot)$ function. The complete list of the rules is displayed in the following Table 2. The concluding functions, i.e. $\hat{\mu}_{fr}(V_{fr})$ and $\hat{\mu}_{fl}(V_{fl})$, are determined based on the designed rules and fuzzy set operations. Then, a crisp value is obtained by means of the centroid method expressed as follows:

$$\begin{cases} V_r = \dfrac{\sum\limits_{V_{fr}} V_{fr}\, \hat{\mu}_{fr}(V_{fr})}{\sum\limits_{V_{fr}} \hat{\mu}_{fr}(V_{fr})} \\[4mm] V_l = \dfrac{\sum\limits_{V_{fl}} V_{fl}\, \hat{\mu}_{fl}(V_{fl})}{\sum\limits_{V_{fl}} \hat{\mu}_{fl}(V_{fl})} \end{cases} \quad (18)$$

Based on the proposed FSMC design, discussion of the corresponding simulation and experiment results is presented in the next section.
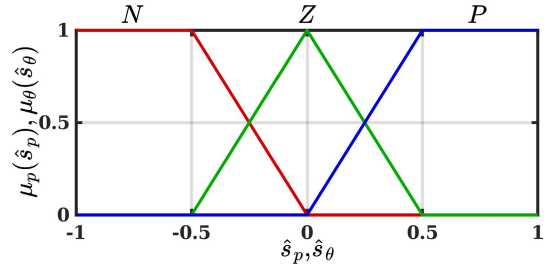
FIGURE 6. Fuzzy membership functions μ_p and μ_θ of inputs $\hat{s}_p$ and $\hat{s}_\theta$, respectively.
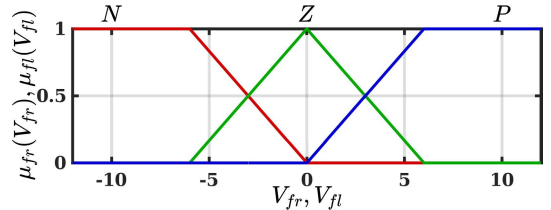
FIGURE 7. Fuzzy membership functions μ_{fr} and μ_{fl} of outputs V_{fr} and V_{fl}, respectively.

TABLE 2. Complete list of fuzzy inference rules.

| (V_{fr}, V_{fl}) | | μ_θ | |
		N	Z	P
μ_p	N	(N, P)	(P, P)	(P, N)
	Z	(N, P)	(Z, Z)	(P, N)
	P	(P, N)	(N, N)	(N, P)

IV. DISCUSSION OF SIMULATION AND EXPERIMENT RESULTS

The proposed method applied to the ADDMR system is tested to follow the planned Bézier trajectory. The trajectory is designed using the following via points as

$$V = \begin{pmatrix} 0 & 0 \\ 0.5 & -0.5 \\ 0 & -1.0 \\ -0.5 & -1.5 \\ 0 & -2.0 \\ 0.5 & -2.5 \\ 0 & -3.0 \\ -0.5 & -3.5 \\ 0 & -4.0 \end{pmatrix} \quad (19)$$

The via points are chosen to mimic a sinusoidal path. Using Eq. (11), we find the parameter points and the corresponding matrix form of the Bézier curve as follows (20), as shown at the bottom of the next page. The resulting path can be seen in Fig. 8.

To follow a speed variation in the trajectory, the path is designed to end at 60 s with a nonlinear arc length function as follows:

$$\begin{cases} s(t) = -9.3 \cdot 10^{-6} t^3 + 8.3 \cdot 10^{-4} t^2 - 3.2 \cdot 10^{-18} t \\ \dot{s}(t) = -27.9 \cdot 10^{-6} t^2 + 16.6 \cdot 10^{-4} t - 3.2 \cdot 10^{-18} \end{cases}$$
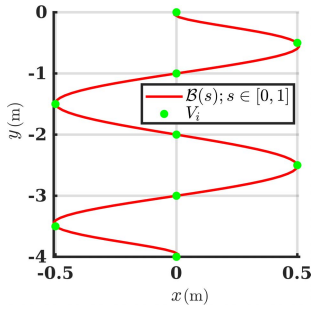$$(21)$$

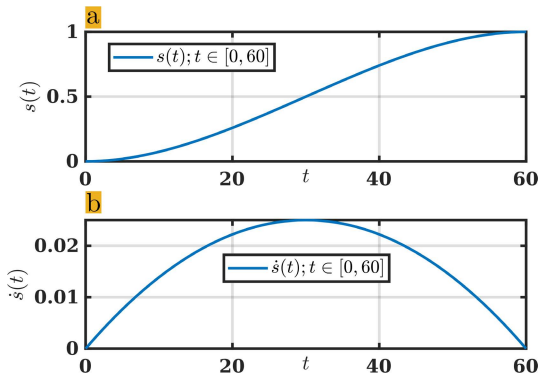**FIGURE 8.** The planned Bézier path $\mathcal{B}(s)$.

FIGURE 9. The evolution of arc length (a) $s(t)$ and (b) $\dot{s}(t)$ over time period.

TABLE 3. List of parameters in the simulation.

Parameter	Value	Unit	Parameter	Value	Unit
R_a	1	Ω	R	40	mm
L_a	0.94	mH	L	25	cm
K_b	0.03	$V/rad/s$	k_{np}	2	-
K_t	0.03	N/A	$k_{n\theta}$	0.02	-
M	8.3	kg	k_p, k_θ	1	-
J	0.098	kgm^2	f_p, f_θ	1	-
T_v	0.058	$N/rad/s$	T_c	0.055	Nm
T_s	0.120	Nm	φ_s	0.04	rad/s

and the evolution of $s(t)$ and $\dot{s}(t)$ can be seen in Fig. 9.

In practical situation, there are some torque disturbances generated from frictions or uneven floors. In order to investigate this effect, friction torque τ_f and uncertainty torque τ_u are applied to the ADDMR control system. Both disturbances
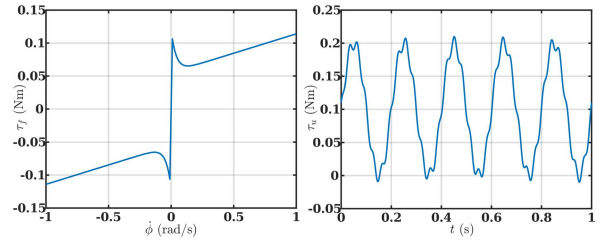

FIGURE 10. Disturbance torques τ_f and τ_u.
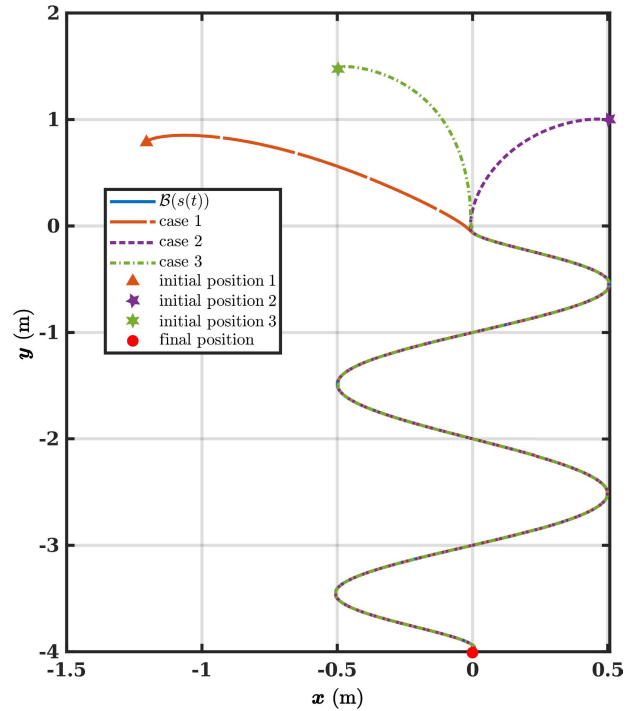

FIGURE 11. Trajectory-tracking responses of the simulated ADDMR in the presence of disturbance.

are formulated as follows:

$$
\begin{cases}
\tau_d = \tau_f + \tau_u \\
\tau_f = T_v\dot{\varphi} + \left[T_c + (T_s - T_c)e^{-\left|\frac{\varphi}{\dot{\varphi}_s}\right|} \right] \mathrm{sign}(\dot{\varphi}) \\
\tau_u = [0.1 + 0.1\sin(10\pi t) + 0.01\cos(62\pi t)]\,\mathrm{sign}(\dot{\varphi})
\end{cases}
\tag{22}
$$

$$
\mathcal{B}(s) = \begin{pmatrix} s^0 & s^1 & \cdots & s^8 \end{pmatrix} \cdot \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-8 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
28 & -56 & 28 & 0 & 0 & 0 & 0 & 0 & 0 \\
-56 & 168 & -168 & 56 & 0 & 0 & 0 & 0 & 0 \\
70 & -280 & 420 & -280 & 70 & 0 & 0 & 0 & 0 \\
-56 & 280 & -560 & 560 & -280 & 56 & 0 & 0 & 0 \\
28 & -168 & 420 & -560 & 420 & -168 & 28 & 0 & 0 \\
-8 & 56 & -168 & 280 & -280 & 168 & -56 & 8 & 0 \\
1 & -8 & 28 & -56 & 70 & -56 & 28 & -8 & 1
\end{pmatrix} \begin{pmatrix}
0 & 0 \\
-0.3 & -0.5 \\
5.7 & -1.0 \\
-9.0 & -1.5 \\
0 & -2.0 \\
9.0 & -2.5 \\
-5.7 & -3.0 \\
0.3 & -3.5 \\
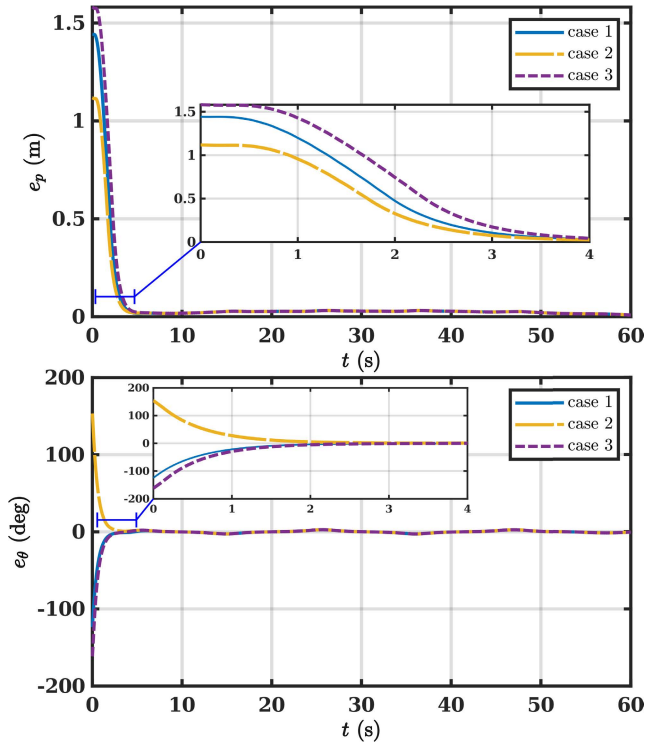0 & -4.0
\end{pmatrix}
\tag{20}
$$

FIGURE 12. e_p and e_θ responses of the simulated ADDMR in the presence of disturbance starting from different initial positions.
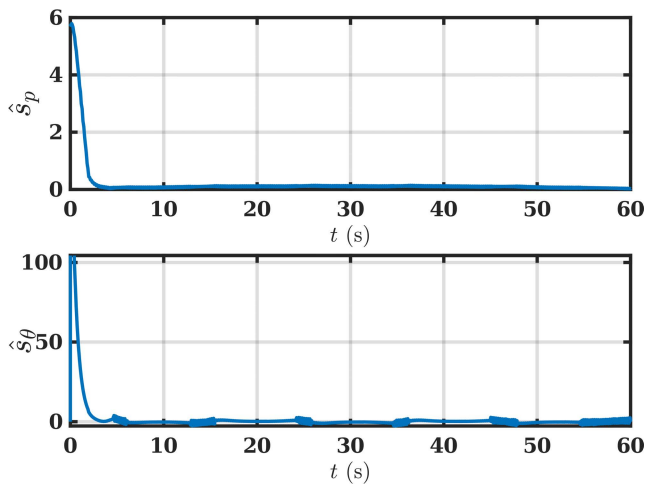

FIGURE 13. Sliding surfaces responses of the simulated ADDMR in the presence of disturbance for case 1.

FIGURE 14. Tracking responses of each case in the experimental implementation.

FIGURE 15. Linear velocity v_x and angular velocity $\dot{\theta}$ responses of each case in the experimental implementation.

The friction torque is based on friction model with stribeck effect [59], [60] where T_v, T_c, T_s, and $\dot{\varphi}_s$ are viscous friction, coulomb friction, stiction friction, and stribeck velocity constants, respectively. The friction model gives variation of disturbance torque with respect to angular velocity of the wheels $\dot{\varphi}$. Furthermore, the uncertainty torque gives variation of disturbance torque with respect to time which includes low and high frequency components. The system and control parameters in the simulation are listed in Table 3 and the illustration of the disturbance is shown in Fig 10.
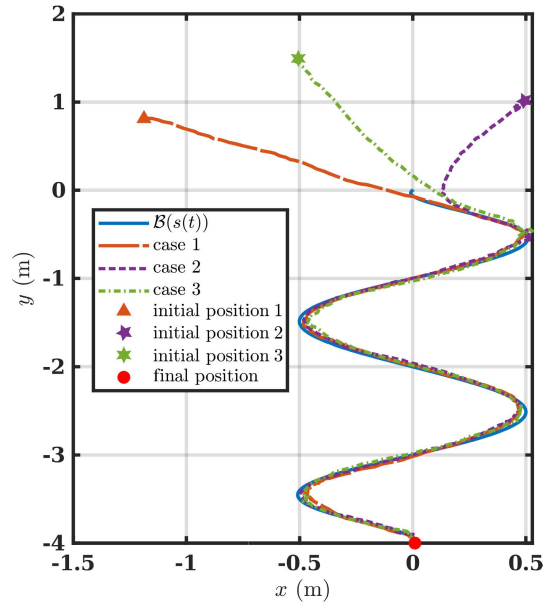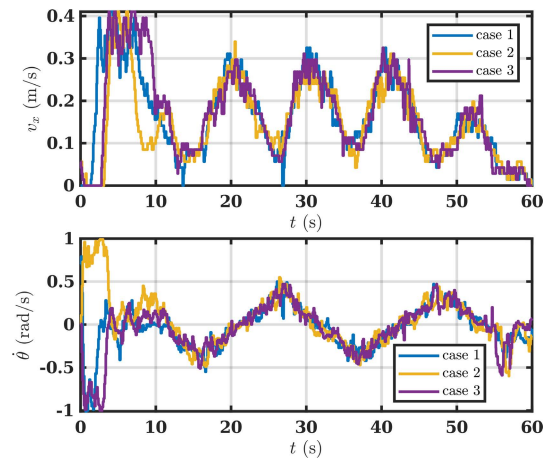
During simulations and experiments, the ADDMR is initially placed outside the path. The initial positions for each case are $(x_{ci}, y_{ci}) = \{(-1.2, 0.8), (0.5, 1), (-0.5, 1.5)\}$ and the initial orientation is $\theta_{ci} = 90°$ for all cases.

It is clearly seen from Fig. 11 that the proposed FSMC can achieve excellent trajectory-tracking of the ADDMR despite the presence of disturbance torques. Similarly, the orientation tracking is also good and displayed in Fig. 12 where the rise time of e_p is 2 s and the rise time of e_θ is 1.26 s. Moreover, Fig. 13 shows that the sliding surface of case 1 converges to zero as the robot successfully follows the planned trajectory. The provided simulations validate that the proposed method is capable of controlling the ADDMR to follow the planned trajectory under torque disturbances.
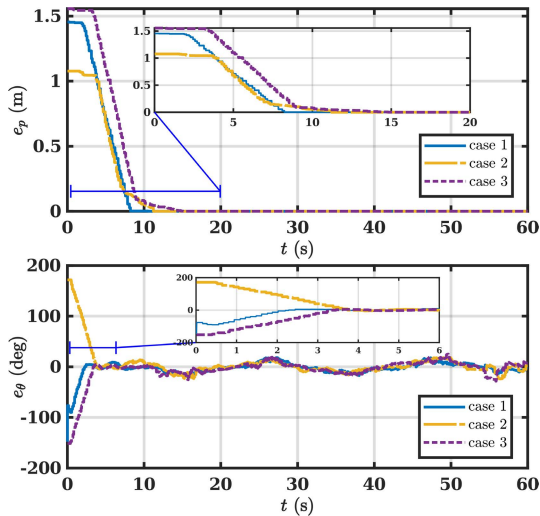
FIGURE 16. e_p and e_θ responses of each case in the experimental implementation.

In the experiments, the proposed FSMC is implemented on ADLINK NeuronBot. The robot follows the same trajectory at the same initial conditions as the simulations. The implementation result can be seen in Fig. 14 where the trajectory-tracking is satisfactory despite the presence of the floor disturbances. Compared to the simulations shown in Fig. 11, the curvatures are smaller when approaching the desired trajectory. This is because the robot operates indoors and its speed limit is needed for safety consideration. As seen in Fig. 15, the linear velocity of the robot does not exceed 0.4 m/s. As a result, e_θ is reduced much faster than e_p as shown in Fig. 16 where the robot made a sharper move compared to the simulations. Furthermore, it can be seen from error responses that the robot completes the trajectory-tracking in a specified time.

V. CONCLUSION

In this study, the FSMC technique is proposed to control an ADDMR subject to torque disturbances. In addition, the comprehensive model of the ADDMR consisting of kinematics, dynamics, and actuator dynamics is established. The trajectory is planned via Bézier curve. In addition, the pose measurement of the ADDMR is measured through MCL technique based on a LiDAR point cloud data. The numerical and experimental validation results prove the effectiveness and robustness of the proposed control scheme.

REFERENCES

[1] H.-H. Chang and C. D. Meyerhoefer, "COVID-19 and the demand for online food shopping services: Empirical evidence from Taiwan," *Amer. J. Agricult. Econ.*, vol. 103, no. 2, pp. 448–465, 2021.

[2] J. Grashuis, T. Skevas, and M. S. Segovia, "Grocery shopping preferences during the COVID-19 pandemic," *Sustainability*, vol. 12, no. 13, p. 5369, Jul. 2020.

[3] A. Bolu and O. Korcak, "Adaptive task planning for multi-robot smart warehouse," *IEEE Access*, vol. 9, pp. 27346–27358, 2021.

[4] M.-K. Ng, Y.-W. Chong, K.-M. Ko, Y.-H. Park, and Y.-B. Leau, "Adaptive path finding algorithm in dynamic environment for warehouse robot," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13155–13171, Sep. 2020.

[5] A. Pani, S. Mishra, M. Golias, and M. Figliozzi, "Evaluating public acceptance of autonomous delivery robots during COVID-19 pandemic," *Transp. Res. D, Transp. Environ.*, vol. 89, Dec. 2020, Art. no. 102600.

[6] Y. Shen, D. Guo, F. Long, L. A. Mateos, H. Ding, Z. Xiu, R. B. Hellman, A. King, S. Chen, C. Zhang, and H. Tan, "Robots under COVID-19 pandemic: A comprehensive survey," *IEEE Access*, vol. 9, pp. 1590–1615, 2021.

[7] Y.-H. Lee, S.-G. Kim, T.-Y. Kuc, J.-K. Park, S.-H. Ji, Y.-S. Moon, and Y.-J. Cho, "Virtual target tracking of mobile robot and its application to formation control," *Int. J. Control, Autom. Syst.*, vol. 12, no. 2, pp. 390–398, Apr. 2014.

[8] A. Morro, A. Sgorbissa, and R. Zaccaria, "Path following for unicycle robots with an arbitrary path curvature," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 1016–1023, Oct. 2011.

[9] F. Heidari and R. Fotouhi, "A human-inspired method for point-to-point and path-following navigation of mobile robots," *J. Mech. Robot.*, vol. 7, no. 4, Nov. 2015, Art. no. 041025.

[10] R. Anushree and B. K. S. Prasad, "Design and development of novel control strategy for trajectory tracking of mobile robot: Featured with tracking error minimization," in *Proc. IEEE Annu. India Conf. (INDICON)*, Dec. 2016, pp. 1–6.

[11] M. Asai, G. Chen, and I. Takami, "Neural network trajectory tracking of tracked mobile robot," in *Proc. 16th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2019, pp. 225–230.

[12] N. Y. Allagui, D. B. Abid, and N. Derbel, "Autonomous navigation of mobile robot with combined fractional order PI and fuzzy logic controllers," in *Proc. 16th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2019, pp. 78–83.

[13] Y. Su and Y. Liang, "Research on double closed-loop trajectory tracking control algorithm of wheeled mobile robot based on global stability," in *Proc. 12th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Aug. 2020, pp. 180–183.

[14] M. Faisal, M. Algabri, B. M. Abdelkader, H. Dhahri, and M. M. Al Rahhal, "Human expertise in mobile robot navigation," *IEEE Access*, vol. 6, pp. 1694–1705, 2018.

[15] D. Chwa and J. Boo, "Adaptive fuzzy output feedback simultaneous posture stabilization and tracking control of wheeled mobile robots with kinematic and dynamic disturbances," *IEEE Access*, vol. 8, pp. 228863–228878, 2020.

[16] M. H. Falsafi, K. Alipour, and B. Tarvirdizadeh, "Tracking-error fuzzy-based control for nonholonomic wheeled robots," *Arabian J. Sci. Eng.*, vol. 44, no. 2, pp. 881–892, Feb. 2019.

[17] W. Benaziza, N. Slimane, and A. Mallem, "Disturbances elimination with fuzzy sliding mode control for mobile robot trajectory tracking," *Adv. Elect. Electron. Eng.*, vol. 16, no. 3, pp. 297–310, 2018.

[18] A. Alouache and Q. Wu, "Fuzzy logic PD controller for trajectory tracking of an autonomous differential drive mobile robot (ie Quanser Qbot)," *Ind. Robot: Int. J.*, vol. 45, no. 1, pp. 23–33, Jan. 2018.

[19] K. Akka and F. Khaber, "Optimal fuzzy tracking control with obstacles avoidance for a mobile robot based on takagi-sugeno fuzzy model," *Trans. Inst. Meas. Control*, vol. 41, no. 10, pp. 2772–2781, Jun. 2019.

[20] T. Q. Khai, Y.-J. Ryoo, W.-R. Gill, and D.-Y. Im, "Design of kinematic controller based on parameter tuning by fuzzy inference system for trajectory tracking of differential-drive mobile robot," *Int. J. Fuzzy Syst.*, vol. 22, no. 6, pp. 1972–1978, Sep. 2020.

[21] J. Zhang, Q. Gong, Y. Zhang, and J. Wang, "Finite-time global trajectory tracking control for uncertain wheeled mobile robots," *IEEE Access*, vol. 8, pp. 187808–187813, 2020.

[22] M. Abdelwahab, V. Parque, A. M. R. Fath Elbab, A. A. Abouelsoud, and S. Sugano, "Trajectory tracking of wheeled mobile robots using Z-number based fuzzy logic," *IEEE Access*, vol. 8, pp. 18426–18441, 2020.

[23] R. Fierro and F. L. Lewis, "Control of a nonholomic mobile robot: Backstepping kinematics into dynamics," *J. Robot. Syst.*, vol. 14, no. 3, pp. 149–163, 1997.

[24] J. Keighobadi and M. B. Menhaj, "From nonlinear to fuzzy approaches in trajectory tracking control of wheeled mobile robots," *Asian J. Control*, vol. 14, no. 4, pp. 960–973, Jul. 2012.

[25] D. Huang, J. Zhai, W. Ai, and S. Fei, "Disturbance observer-based robust control for trajectory tracking of wheeled mobile robots," *Neurocomputing*, vol. 198, pp. 74–79, Jul. 2016.

[26] S. Wang and J. Zhai, "A trajectory tracking method for wheeled mobile robots based on disturbance observer," *Int. J. Control, Autom. Syst.*, vol. 18, no. 8, pp. 2165–2169, Aug. 2020.

[27] Z. Xu, S. X. Yang, and S. A. Gadsden, "Enhanced bioinspired backstepping control for a mobile robot with unscented Kalman filter," *IEEE Access*, vol. 8, pp. 125899–125908, 2020.

[28] C. Ben Jabeur and H. Seddik, "Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot," *Asian J. Control*, vol. 23, no. 1, pp. 23–41, 2021.

[29] H. Yu, N. Sheng, and Z. Ai, "Sliding mode control for trajectory tracking of mobile robots," in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 13–17.

[30] L. Morales, M. Herrera, O. Camacho, P. Leica, and J. Aguilar, "LAMDA control approaches applied to trajectory tracking for mobile robots," *IEEE Access*, vol. 9, pp. 37179–37195, 2021.

[31] H. Ye and S. Wang, "Trajectory tracking control for nonholonomic wheeled mobile robots with external disturbances and parameter uncertainties," *Int. J. Control, Autom. Syst.*, vol. 18, no. 12, pp. 3015–3022, Dec. 2020.

[32] A. Azzabi and K. Nouri, "Design of a robust tracking controller for a nonholonomic mobile robot based on sliding mode with adaptive gain," *Int. J. Adv. Robotic Syst.*, vol. 18, no. 1, 2021, Art. no. 1729881420987082.

[33] T. A. Mai, T. S. Dang, D. T. Duong, V. C. Le, and S. Banerjee, "A combined backstepping and adaptive fuzzy PID approach for trajectory tracking of autonomous mobile robots," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 43, no. 3, pp. 1–13, Mar. 2021.

[34] J. Nie, Y. Wang, Z. Miao, Y. Jiang, H. Zhong, and J. Lin, "Adaptive fuzzy control of mobile robots with full-state constraints and unknown longitudinal slipping," *Nonlinear Dyn.*, vol. 106, no. 4, pp. 3315–3330, Dec. 2021.

[35] B. Moudoud, H. Aissaoui, and M. Diany, "Fuzzy adaptive sliding mode controller for electrically driven wheeled mobile robot for trajectory tracking task," *J. Control Decis.*, vol. 9, no. 1, pp. 71–79, Jan. 2022.

[36] S. Peng and W. Shi, "Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot," *IEEE Access*, vol. 6, pp. 43414–43424, 2018.

[37] K. Shao, J. Zheng, H. Wang, X. Wang, R. Lu, and Z. Man, "Tracking control of a linear motor positioner based on barrier function adaptive sliding mode," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7479–7488, Nov. 2021.

[38] K. Shao, R. Tang, F. Xu, X. Wang, and J. Zheng, "Adaptive sliding mode control for uncertain Euler–Lagrange systems with input saturation," *J. Franklin Inst.*, vol. 358, no. 16, pp. 8356–8376, 2021.

[39] C. Yan, K. Shao, X. Wang, J. Zheng, and B. Liang, "Reference governor-based control for active rollover avoidance of mobile robots," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 429–435.

[40] Y.-T. Wang, C.-C. Peng, A. Ravankar, and A. Ravankar, "A single LiDAR-based feature fusion indoor localization algorithm," *Sensors*, vol. 18, no. 4, p. 1294, Apr. 2018.

[41] S. Wang, X. Chen, G. Ding, Y. Li, W. Xu, Q. Zhao, Y. Gong, and Q. Song, "A lightweight localization strategy for LiDAR-guided autonomous robots with artificial landmarks," *Sensors*, vol. 21, no. 13, p. 4479, Jun. 2021.

[42] J. Palacín, D. Martínez, E. Rubies, and E. Clotet, "Mobile robot self-localization with 2D push-broom LIDAR in a 2D map," *Sensors*, vol. 20, no. 9, p. 2500, Apr. 2020.

[43] Y. Shi, W. Zhang, F. Li, and Q. Huang, "Robust localization system fusing vision and lidar under severe occlusion," *IEEE Access*, vol. 8, pp. 62495–62504, 2020.

[44] J. Velagic, B. Lacevic, and N. Osmic, *Nonlinear Motion Control of Mobile Robot Dynamic Model*. Rijeka, Croatia: IntechOpen, Jun. 2008.

[45] R. Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton–Euler methodologies: A unified framework," *Adv. Robot. Automat.*, vol. 2, no. 2, pp. 1–7, 2013.

[46] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Amsterdam, The Netherlands: Elsevier, 2013.

[47] M. A. Ali and M. Mailah, "A simulation and experimental study on wheeled mobile robot path control in road roundabout environment," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419834778.

[48] A. Stefek, T. V. Pham, V. Krivanek, and K. L. Pham, "Energy comparison of controllers used for a differential drive wheeled mobile robot," *IEEE Access*, vol. 8, pp. 170915–170927, 2020.

[49] S. Nurmaini, "Differential drive mobile robot control using variable fuzzy universe of discourse," in *Proc. Int. Conf. Electr. Eng. Comput. Sci. (ICECOS)*, Aug. 2017, pp. 50–55.

[50] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, vol. 118, 2nd ed. Berlin, Germany: Springer, 2017.

[51] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.*, vol. 128, nos. 1–2, pp. 99–141, May 2001.

[52] N. Akai, S. Hoshino, K. Inoue, and K. Ozaki, "Monte Carlo localization using magnetic sensor and LIDAR for real world navigation," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Dec. 2013, pp. 682–687.

[53] L. Tiancheng, B. Miodrag, and D. M. Petar, "Resampling methods for particle filtering: Classification, implementation, and strategies," *Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.

[54] A. Yilmaz and H. Temeltas, "Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LIDAR data," *Robot. Auto. Syst.*, vol. 122, Dec. 2019, Art. no. 103285.

[55] S. J. Chapman, *Fortran 90/95 for Scientists and Engineers*, 1st ed. New York, NY, USA: McGraw-Hill, 1998.

[56] MathWorks, *MATLAB The Language of Technical Computing: Function Reference: A—E Version 7*, MathWorks, Natick, MA, USA, 2004, vol. 1.

[57] V. Torres and J. Valls, "A fast and low-complexity operator for the computation of the arctangent of a complex number," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2663–2667, Sep. 2017.

[58] V. Torres, J. Valls, and R. Lyons, "Fast- and low-complexity atan2(a,b) approximation [Tips and Tricks]," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 164–169, Nov. 2017.

[59] A. G. Conceicao, M. D. Correia, and L. Martinez, "Modeling and friction estimation for wheeled omnidirectional mobile robots," *Robotica*, vol. 34, no. 9, pp. 2140–2150, 2016.

[60] F. Dai, X. Gao, S. Jing, W. Guo, and Y. Liu, "A two-wheeled inverted pendulum robot with friction compensation," *Mechatronics*, vol. 30, pp. 116–125, Sep. 2015.

HSIU-MING WU (Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, National Taiwan University of Science and Technology, in 2009. He was a Postdoctoral Fellow with the Opto-Mechatronics Technology Center, National Taiwan University of Science and Technology. In early 2013, he worked as a Postdoctoral Fellow with Texas A&M University at Qatar for four years. In early 2018, he joined the Department of Mechanical Engineering, National Chin-Yi University of Technology, Taichung, for two years. Currently, he is an Assistant Professor with the Department of Intelligent Automation Engineering, National Taipei University of Technology. His research interests include mechatronics, nonlinear control, and robotic control.

MUHAMMAD QOMARUZ ZAMAN received the bachelor's and master's degrees in electrical engineering from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2018. He is currently pursuing the Ph.D. degree with the National Taipei University of Technology. From 2015 to 2018, he was a Research Assistant with the Microcontroller Laboratory, Institut Teknologi Sepuluh Nopember. His research interests include robotics and the IoT.

● ● ●