TRAJECTORY TRACKING CONTROL FOR FOUR-MECANUM-WHEELED ROBOTS CONSIDERING MASS ECCENTRICITY: A FUZZY ADAPTIVE AND DYNAMIC SURFACE CONTROL APPROACH

MINH DONG NGUYEN[1], QUANG HIEP DO[1], PHUONG NAM DAO[2], MANH TIEN NGO[3],*

[1]*Faculty of Electrical-Automation, University of Economics-Technology for Industries, 456 Minh Khai Street, Hai Ba Trung District, Ha Noi, Viet Nam*
[2]*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, 1 Dai Co Viet Street, Hai Ba Trung District, Ha Noi, Viet Nam*
[3]*Institute of Physics, Vietnam Academy of Science and Technology, 10 Dao Tan Street, Ba Dinh District, Ha Noi, Viet Nam*

Abstract. Mass eccentricity, the displacement of a robot's center of mass, presents significant challenges for precise control and stability. This paper introduces a novel approach that combines fuzzy adaptive control with dynamic surface control to improve trajectory tracking in omnidirectional four-Mecanum-wheeled mobile robots, accounting for mass eccentricity. The proposed control strategy tackles these challenges by leveraging fuzzy logic's adaptability and dynamic surface control's robustness. Simulation results demonstrate that this approach substantially enhances trajectory tracking accuracy and overall system stability.

Keywords. Four-Mecanum-wheeled mobile robot, fuzzy adaptive control, dynamic surface control, trajectory tracking, mass eccentricity, nonlinear control.

1. INTRODUCTION

The advancement of mobile robotics has greatly benefited from the development of various wheel configurations and control strategies aimed at improving maneuverability and stability. Among these, omnidirectional robots equipped with Mecanum wheels have attracted considerable attention due to their exceptional ability to execute complex trajectories and navigate confined spaces with high precision. This versatility makes Mecanum-wheeled robots particularly valuable in a wide range of applications, including smart wheelchairs, forklifts, industrial material handling, and collaborative robots equipped with arms for tasks in manufacturing environments and exploration.

The omnidirectional movement capability of Mecanum-wheeled robots stems from the unique design of their wheels. Each wheel features passive rollers arranged at a 45-degree angle relative to the wheel's main axis. When the wheel rotates, part of the translational

force is converted into lateral sliding force, enabling the wheel to move forward, backward, and sideways simultaneously. A typical Mecanum robot is equipped with four independently driven Mecanum wheels, each powered by a separate motor. By controlling the rotation of each wheel in different directions, various movement vectors are achieved, allowing the robot to navigate in multiple directions, as demonstrated in the study [1, 2]. Kinematic equations and wheel configurations for Mecanum wheels, along with their various applications, are outlined in studies [3–5]. The dynamic equations are discussed in papers [6–9], but these studies only consider robots with constant mass and center of gravity. However, in some applications, adding a payload alters both the total mass and the center of gravity, which can affect stability and control performance. Therefore, our study introduces a more comprehensive model for a four-Mecanum-wheeled mobile robot, incorporating both kinematic and dynamic equations that account for mass eccentricity.

Recently, robot motion control algorithms have been explored in studies [10–12], which propose sliding mode control techniques for robots with uncertain parameters and external disturbances. Among these, an adaptive terminal sliding mode control scheme has been suggested for robots equipped with suction mechanisms. Furthermore, the authors in [13] proposed a PID control algorithm combined with fuzzy logic for systems with variable loads. Furthermore, the combination of PID and fast terminal sliding mode control, as proposed in [14, 15], incorporates a disturbance observer, where a sliding mode observer is used in the feedback loop to address instability issues in the feedback channel. Studies [16–18] discuss motion control methods for robots using fuzzy logic algorithms. Specifically, in [16] an IMU sensor is used to determine the robot orientation angle, which informs the application of a fuzzy logic control algorithm. This fuzzy controller consists of 25 rules, with the system's errors serving as input signals. The Sugeno model with Mamdani functions is employed for the fuzzy logic system. The output of the control algorithm is a PWM signal sent to the wheel motors.

Dynamic surface control has emerged as a robust method of managing non-linear systems, effectively mitigating the computational complexity associated with traditional sliding mode control techniques, as seen in typical studies [19–21]. Recent studies have applied dynamic surface control to autonomous robots, demonstrating significant improvements in trajectory tracking and disturbance rejection compared to traditional control methods [22–25]. The simulation results indicate that these algorithms provide good control quality and reduce oscillations. However, existing studies focus primarily on algorithms for omnidirectional robots with a fixed center of mass. In studies [26, 27], the authors developed a comprehensive dynamic model for a four-wheeled Mecanum robot, emphasizing the effects of mass eccentricity and friction uncertainty, and employed nonlinear adaptive control to enhance system performance. Mass eccentricity, which refers to the displacement of the robot's center of mass from its geometric center, poses significant challenges for precise control and trajectory tracking. Therefore, this study explores the application of fuzzy adaptive control combined with dynamic surface control for trajectory tracking in mobile robots with mass eccentricity.

Our primary contributions in this study are: (1) we conduct a comprehensive mathematical analysis of both the kinematic and dynamic models of four-Mecanum-wheeled mobile robots (FMWR), incorporating the effects of mass eccentricity using Lagrange equations; (2) we propose an adaptive fuzzy dynamic surface control (Fuzzy-DSC) algorithm that syn-

ergizes the benefits of dynamic surface control and fuzzy logic, providing robust trajectory tracking capabilities under mass eccentricity conditions; and (3) we perform extensive simulations to demonstrate the superior performance of the proposed control algorithm compared to conventional baseline controllers.

The remainder of this paper is organized as follows. Section 2 discusses the kinematic and dynamic modeling of the FMWR, taking into account the eccentricity of the mass. Section 3 explains the design and implementation of the Fuzzy-DSC algorithm. Section 4 presents simulation results and evaluates the performance of the proposed controller compared to the baseline methods. Finally, Section 5 concludes the paper with a summary of the findings and implications.

2. ROBOT MODEL WITH MASS ECCENTRICITY

The model of the four-Mecanum-wheeled mobile robot is illustrated in Figure 1. It consists of four Mecanum wheels driven by independent motors, arranged symmetrically. Each wheel has rollers angled at 45 degrees relative to the wheel's main axis. As the wheel moves, a portion of the translational force is converted into lateral sliding force, allowing the robot to move simultaneously forward and sideways without altering its heading angle [28].
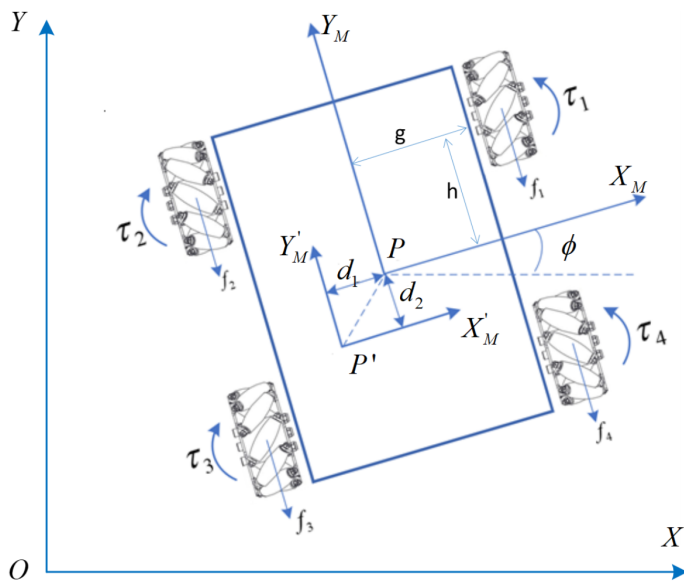

Figure 1: Robot model incorporating mass eccentricity

Here, the XOY coordinate system represents the inertial reference frame, while the $X_M O_M Y_M$ coordinate system is fixed to the robot's center, with the robot's heading angle ϕ rotating around the Z_M axis. The geometric center of the robot is $P = \begin{bmatrix} x & y \end{bmatrix}^T$ and the center of mass position of the robot is represented as $P' = \begin{bmatrix} x - d_1 & y - d_2 \end{bmatrix}^T$ with their respective offsets given by d_1 and d_2.

The kinematic equations in the inertial coordinate system are described as follows [26]

$$\dot{\eta}_R = H(\phi)\dot{\eta}, \tag{1}$$

where

- $\dot{\eta}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\phi}_R \end{bmatrix}$ is the velocity vector in the robot's frame,

- $\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}$ is the velocity vector in the inertial frame,

- $H(\phi)$ is the coordinate transformation matrix,

$$H(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The total kinematic energy of the entire robot can be computed as

$$T = \frac{1}{2}\left[m_b \dot{P}^T \dot{P} + J_b \dot{\phi}^2 + \sum_1^4 m_{wi}(r\dot{\psi}_i)^2 + \sum_1^4 J_{wi}\dot{\psi}_i^{\,2} \right], \tag{2}$$

where m_b and m_{wi} is the mass of the robot and the i-th wheel, J_b and J_{wi} is the inertial moment of the robot and the i-th wheel with respect to its axis, r is the wheel radius and ψ_i is angular velocity of the i-th wheel. Assuming the robot moves on a flat surface, the total potential energy $V = 0$. The Lagrangian function becomes $L = T - V = T$. According to the Euler-Lagrange equations, the complete dynamic equations are determined as follows [26]

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\eta}_i} \right) - \frac{\partial L}{\partial \eta_i} = Q_i, i = 1, 2, 3, \tag{3}$$

where η_i is the generalized coordinate for the i-th degree of freedom, Q_i represents the force or moment acting on the i-th degree of freedom.

Differentiating the Lagrangian function according to (3), we obtain

$$\begin{aligned} Q_1 &= \left[\tau_1 - r\,\mathrm{sgn}\left(\dot{\psi}_1\right)\zeta_1 \right]\left[\frac{1}{r}(\sin\phi - \cos\phi) \right] - \left[\tau_2 - r\,\mathrm{sgn}\left(\dot{\psi}_2\right)\zeta_2 \right]\left[\frac{1}{r}(\cos\phi + \sin\phi) \right] \\ &+ \left[\tau_3 - r\,\mathrm{sgn}\left(\dot{\psi}_3\right)\zeta_3 \right]\left[\frac{1}{r}(\cos\phi - \sin\phi) \right] + \left[\tau_4 - r\,\mathrm{sgn}\left(\dot{\psi}_4\right)\zeta_4 \right]\left[\frac{1}{r}(\cos\phi + \sin\phi) \right], \end{aligned} \tag{4}$$

$$\begin{aligned} Q_2 &= \left[\tau_2 - r\,\mathrm{sgn}\left(\dot{\psi}_2\right)\zeta_2 \right]\left[\frac{1}{r}(\cos\phi - \sin\phi) \right] - \left[\tau_1 - r\,\mathrm{sgn}\left(\dot{\psi}_1\right)\zeta_1 \right]\left[\frac{1}{r}(\sin\phi + \cos\phi) \right] \\ &+ \left[\tau_3 - r\,\mathrm{sgn}\left(\dot{\psi}_3\right)\zeta_3 \right]\left[\frac{1}{r}(\sin\phi + \cos\phi) \right] + \left[\tau_4 - r\,\mathrm{sgn}\left(\dot{\psi}_4\right)\zeta_4 \right]\left[\frac{1}{r}(\sin\phi - \cos\phi) \right], \end{aligned} \tag{5}$$

$$\begin{aligned} Q_3 &= (\tau_1 + \tau_2 + \tau_3 + \tau_4)\left[-\frac{\sqrt{2}}{r}l\sin\left(\frac{\pi}{4} - \alpha\right) \right] \\ &+ \left[\mathrm{sgn}\left(\dot{\psi}_1\right)\zeta_1 + \mathrm{sgn}\left(\dot{\psi}_2\right)\zeta_2 + \mathrm{sgn}\left(\dot{\psi}_3\right)\zeta_3 + \mathrm{sgn}\left(\dot{\psi}_4\right)\zeta_4 \right]\left[\sqrt{2}l\sin\left(\frac{\pi}{4} - \alpha\right) \right], \end{aligned} \tag{6}$$

where ζ_i is the frictional force of the i-th wheel with the floor.

From equation (4),(5) and (6), we can derive the complete dynamic equation in the presence of center of mass offset P',

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} + D\delta = D\tau + \tau_d, \tag{7}$$

where $\tau = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix}^{\mathrm{T}}$ is the moment applied to each wheel, τ_d is unknown general external disturbances, r is the radius of the wheel, $\dot{\eta}$ is the velocity vector of the robot, $\ddot{\eta}$ is the acceleration vector of the robot,

$$\zeta = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 & \zeta_4 \end{bmatrix}^{\mathrm{T}}, \, \delta = rU\zeta, \, U = \mathrm{diag}\begin{bmatrix} \mathrm{sgn}(\dot{\psi}_1) & \mathrm{sgn}(\dot{\psi}_2) & \mathrm{sgn}(\dot{\psi}_3) & \mathrm{sgn}(\dot{\psi}_4) \end{bmatrix},$$

$$M = \begin{bmatrix} m_1 & 0 & m_3 \\ 0 & m_1 & m_4 \\ m_3 & m_4 & m_2 \end{bmatrix}, \, m_1 = m_b + 4\left(m_w + \frac{J}{r^2}\right),$$

$$m_3 = m_b\left(d_1 \sin\phi + d_2 \cos\phi\right), \, m_4 = m_b\left(-d_1 \cos\phi + d_2 \sin\phi\right),$$

$$m_2 = m_b\left(d_1^2 + d_2^2\right) + J_b + 8\left(m_w + \frac{J}{r^2}\right)l^2 \sin^2\left(\frac{\pi}{4} - \alpha\right),$$

$$C = \begin{bmatrix} 0 & 0 & c_1 \\ 0 & 0 & c_2 \\ 0 & 0 & 0 \end{bmatrix}, \, c_1 = m_b\dot{\phi}\left(d_1 \cos\phi - d_2 \sin\phi\right), \, c_2 = m_b\dot{\phi}\left(d_1 \sin\phi + d_2 \cos\phi\right),$$

$$D = \frac{1}{r}\begin{bmatrix} -(c-s) & -(s+c) & -\sqrt{2}l \sin\left(\frac{\pi}{4} - \alpha\right) \\ -(c+s) & -(s-c) & -\sqrt{2}l \sin\left(\frac{\pi}{4} - \alpha\right) \\ c-s & s+c & -\sqrt{2}l \sin\left(\frac{\pi}{4} - \alpha\right) \\ c+s & s-c & -\sqrt{2}l \sin\left(\frac{\pi}{4} - \alpha\right) \end{bmatrix}^{\mathrm{T}}, \, s = \sin\phi, \, c = \cos\phi.$$

Within the scope of this paper, we assume that frictional forces are known and the primary source of uncertainty arises from the eccentricity of the mass, without external disturbances ($\tau_d = 0$). Control systems addressing general uncertainties and disturbances are discussed in [29].

3. CONTROL ALGORITHM DESIGN

3.1. Dynamic surface control algorithm

Dynamic Surface Control (DSC) is a nonlinear control technique designed to handle uncertainties in system dynamics while reducing the complexity associated with traditional back-stepping or sliding mode control. DSC involves defining a dynamic sliding surface within the state space of the system [19]. The goal is to design a control law that drives the system's state to converge to this surface, ensuring both stability and performance. The dynamic sliding surface is designed on the basis of the system's nonlinear characteristics and the desired trajectory. Subsequently, the control law is formulated to ensure that the system's trajectory remains on this surface.

Consider a nonlinear dynamical system with the following form

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = u + f(x), \end{cases} \tag{8}$$

where f and $\frac{\partial f}{\partial x_1}$ are continuous functions. Our control objective is to use the control signal u to drive x_1 and x_2 to track the desired trajectory x_{1d} and x_{2d}.

Define the first sliding surface $S_1 = x_1 - x_{1d}$.

Define the variables $\bar{x}_2 = \dot{x}_{1d} - K_1 S_1$, with the virtual control signal x_{2d} tracked to $\bar{x}_2$ through a first-order inertia element acting as a filter with $x_{2d}(0) = \bar{x}_2(0)$ and τ' being a designed time constant

$$\tau' \dot{x}_{2d} + x_{2d} = \bar{x}_2. \tag{9}$$

Define the second sliding surface $S_2 = x_2 - x_{2d}$.

The derivative of the sliding surface S_2 is obtained as follows

$$\dot{S}_2 = -\dot{x}_{2d} + u + f(x). \tag{10}$$

Define the variables based on the full dynamic equations (7) and (8) with the control input τ

$$\begin{cases} u = M^{-1} D \tau, \\ f(x) = -M^{-1} \left(C \dot{\eta} - D \delta \right), \end{cases} \tag{11}$$

where we define $\eta = x_1$, the generalized coordinate vector of the robotic system.

Combined with equation (9), the control signal can be designed as follows

$$\tau = -D^T \left(D D^T \right)^{-1} \left[M \left(-\frac{\bar{x}_2 - x_{2d}}{\tau} + K_2 S_2 \right) - C \dot{\eta} - D \delta \right]. \tag{12}$$

This controller works on the dynamical level, where the pose and velocity are feed-backed. Using the first and second sliding surfaces, we determine

$$\dot{S}_1 = -K_1 S_1 + S_2, \tag{13}$$

$$\dot{S}_2 = -K_2 S_2. \tag{14}$$

We choose a Lyapunov candidate function to prove the stability of the system

$$V = \frac{S_1^T S_1 + S_2^T S_2}{2} \geq 0. \tag{15}$$

The derivative of the Lyapunov function

$$\dot{V} = S_1^T \dot{S}_1 + S_2^T \dot{S}_2. \tag{16}$$

Substituting (13) and (14) into (16), we obtain the following

$$\dot{V} = -K_1 S_1^T S_1 - K_2 S_2^T S_2 + S_1^T S_2. \tag{17}$$

We use the following inequality

$$\frac{S_1^T S_1 + S_2^T S_2}{2} \geq S_1^T S_2. \tag{18}$$

Substituting into equation (17), we obtain

$$\dot{V} \leq -(K_1 - \frac{1}{2}I)S_1^T S_1 - (K_2 - \frac{1}{2}I)S_2^T S_2. \tag{19}$$

Here, K_1 and K_2 are control parameters to be designed such that $\dot{V}$ is negative definite. Therefore, if $K_1 > \frac{1}{2}I$ and $K_2 > \frac{1}{2}I$ element-wise, $\dot{V} < 0$ for all nonzero S_1 and S_2. Thus, $S_1 \to 0$ and $S_2 \to 0$ as time $t \to \infty$, proving the asymptotic stability of the system according to the Lyapunov stability theorem. In particular, if the errors are not zero, the DSC method will damp the system's energy ($\dot{V} < 0$), thereby regulating the errors. The states of the system x_1 and x_2 track the desired trajectory x_{1d} and x_{2d}, respectively, over time $t \to \infty$. Therefore, we obtain the following theorem.

Theorem 1. *The nonlinear system* (7) *under the Dynamic Surface Control law* (12) *is guaranteed to track the desired trajectory asymptotically, ensuring that the tracking errors $S_1 = x_1 - x_{1d}$ and $S_2 = x_2 - x_{2d}$ converge to zero as $t \to \infty$, provided that the control gains K_1 and K_2 satisfy $K_1 > \frac{1}{2}I$ and $K_2 > \frac{1}{2}I$.*

The proposed DSC algorithm is summarized below.

Algorithm 1: Dynamic Surface Control (DSC)

    Data: Desired trajectory (x_{1d}, x_{2d}), current state (x_1, x_2), control parameters
        (K_1, K_2, τ')

    Result: Control signal τ

1 **Compute sliding surfaces:**

2 $S_1 \leftarrow x_1 - x_{1d}$;

3 $S_2 \leftarrow x_2 - x_{2d}$;

4 **Compute virtual control:**

5 $\bar{x}_2 \leftarrow \dot{x}_{1d} - K_1 S_1$;

6 $\dot{x}_{2d} \leftarrow \frac{\bar{x}_2 - x_{2d}}{\tau'}$;

7 **Compute system matrices:** D, M, C;

8 **Compute control input:**

9 $\dot{\eta} = x_2$;

10 $\tau \leftarrow -D^T(DD^T)^{-1}[M(-\bar{x}_2 - x_{2d}/\tau + K_2 S_2) - C\dot{\eta} - D\delta]$;

11 **return** τ;

3.2. A hybrid approach: fuzzy adaptive control and dynamic surface control

Dynamic surface control is recognized for its stability in managing uncertain parameters that vary within certain limits. This stability is most effectively achieved when the state of the system is on or near the sliding surface. To ensure the stability of the system, careful selection of controller parameters is essential: k_1 directly influences the quality of the robot's

trajectory tracking, while k_2 affects both the rate of convergence to the sliding surface and the maintenance of stability on it.

However, as the robot's center of mass shifts during operation, it can alter the system's behavior and potentially compromise stability. Therefore, to optimize trajectory tracking and maintain system stability, it is crucial to adjust the controller parameters based on different operational conditions.

To address this issue, the paper proposes using fuzzy logic to tune the controller parameters. This method enables the robot to adapt to changes in its model, thereby enhancing the DSC controller's ability to maintain stability and performance despite variations in system parameters. The control structure diagram of the system is illustrated in Fig. 2
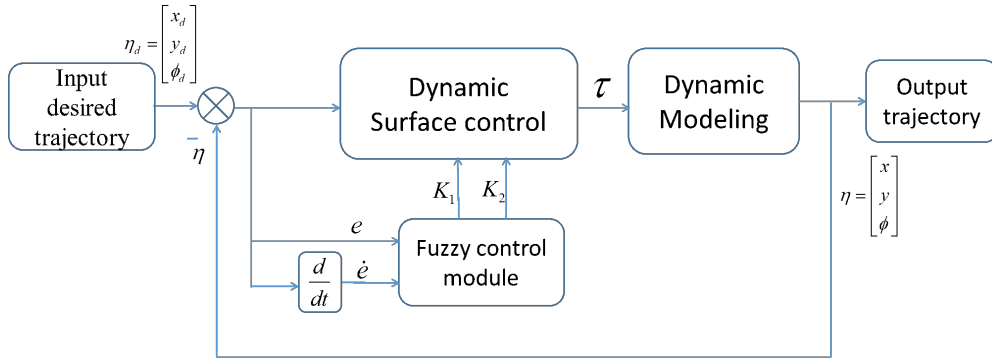

Figure 2: Schematic of the dynamic surface control system combined with fuzzy adaptation

The fuzzy tuning controller is constructed based on the Sugeno fuzzy model [30]. The input signals for this controller are the trajectory tracking error and its time derivative. The output signals, K_1 and K_2, are selected according to a rule-based inference system. The structure of the input and output signals for the fuzzy rule is shown in Figure 3 where N represents negative, P denotes positive, Z indicates zero, VS is very small, S stands for small, M signifies medium, B refers to big, and VB is very big. For example, the values of the outputs $K_1 = \mathrm{diag}(k_1, k_1, k_1)$ and $K_2 = \mathrm{diag}(k_2, k_2, k_2)$ $(k_1, k_2 > \frac{1}{2})$ can be designed as in Table 1.
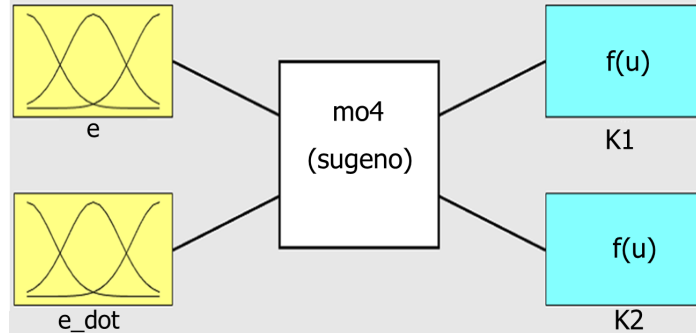

Figure 3: The input and output signals for the fuzzy rule

Building on the preceding discussions and the definitions of the input and output membership functions, the rule base can be established as follows.

Table 1: The values of the output variables

Values	Meaning	k_1	k_2
VS	Very small	2	5
S	Small	4	10
M	Medium	10	15
B	Big	13	20
VB	Very big	20	30

IF (e is NB) and ($\dot{e}$ is NB), THEN (k1 is M)(k2 is M)
IF (e is NS) and ($\dot{e}$ is NB), THEN (k1 is S)(k2 is B)
IF (e is Z) and ($\dot{e}$ is NB), THEN (k1 is VS)(k2 is VB)
IF (e is PS) and ($\dot{e}$ is NB), THEN (k1 is S)(k2 is B)
IF (e is PB) and ($\dot{e}$ is NB), THEN (k1 is M)(k2 is M)
IF (e is NB) and ($\dot{e}$ is NS), THEN (k1 is B)(k2 is S)
IF (e is NS) and ($\dot{e}$ is NS), THEN (k1 is M)(k2 is M)
IF (e is Z) and ($\dot{e}$ is NS), THEN (k1 is S)(k2 is B)
IF (e is PS) and ($\dot{e}$ is NS), THEN (k1 is M)(k2 is M)
IF (e is PB) and ($\dot{e}$ is NS), THEN (k1 is B)(k2 is S)
IF (e is NB) and ($\dot{e}$ is Z), THEN (k1 is VS)(K2 is VB)
IF (e is NS) and ($\dot{e}$ is Z), THEN (k1 is B)(k2 is S)
IF (e is Z) and ($\dot{e}$ is Z), THEN (k1 is M)(k2 is M)
IF (e is PS) and ($\dot{e}$ is Z), THEN (k1 is B)(k2 is S)
IF (e is PB) and ($\dot{e}$ is Z), THEN (k1 is VS)(k2 is VB)
IF (e is NB) and ($\dot{e}$ is PS), THEN (k1 is B)(k2 is S)
IF (e is NS) and ($\dot{e}$ is PS), THEN (k1 is M)(k2 is M)
IF (e is Z) and ($\dot{e}$ is PS), THEN (k1 is S)(k2 is B)
IF (e is PS) and ($\dot{e}$ is PS), THEN (k1 is M)(k2 is M)
IF (e is PB) and ($\dot{e}$ is PS), THEN (k1 is B)(k2 is S)
IF (e is NB) and ($\dot{e}$ is PB), THEN (k1 is M)(k2 is M)
IF (e is NS) and ($\dot{e}$ is PB), THEN (k1 is S)(k2 is B)
IF (e is Z) and ($\dot{e}$ is PB), THEN (k1 is VS)(k2 is VB)
IF (e is PB) and ($\dot{e}$ is PB), THEN (k1 is M)(k2 is M)
IF (e is PS) and ($\dot{e}$ is PB), THEN (k1 is S)(k2 is B)

Remark 1: In this paper, the fuzzy controller's parameters are carefully designed offline to ensure that the DSC loop achieves stability under all conditions. Specifically, our fuzzy rules guarantee that K_1 and K_2 are nonzero and large enough. On the other hand, there are several online real-time adaptive tuning mechanisms that could further improve the performance of the proposed method. This can be accomplished by incorporating self-tuning methods, such as Neuro-Fuzzy Systems (NFIS), Genetic Algorithms (GA), or Reinforcement Learning (RL), which allow for dynamic adjustment of fuzzy parameters based on real-time feedback. Our paper instead focuses on the stability analysis of the proposed Fuzzy-DSC framework.

4.   SIMULATION RESULTS AND ANALYSIS

This section presents MATLAB/Simulink simulation results to illustrate the effectiveness of the proposed control scheme. We conducted three different simulation experiments: figure-eight trajectory tracking without mass eccentricity, figure-eight trajectory tracking with mass eccentricity, and circular trajectory tracking with mass eccentricity.

The parameters of the four-Mecanum-wheeled mobile robot are obtained from our real system and are as follows

$m = 30\,(\text{kg})$, $m_{\text{w}} = 0.9\,(\text{kg})$,
$J = 5\,(\text{kg.m}^2)$, $J_w = 0.1\,(\text{kg.m}^2)$, $g = 0.2\,(\text{m})$, $h = 0.3\,(\text{m})$, $l = 0.25\,(\text{m})$, $r = 0.075\,(\text{m})$,
$G = 9.8\,(\text{m/s}^2)$, $\mu = 0.01$, $\tau' = 0.02$, $\zeta = \begin{bmatrix} 0.05 & 0.05 & 0.05 & 0.05 \end{bmatrix}^T\,(\text{N})$.

We compare the performances of three different controllers:

1. Dynamic Surface Control (DSC) with fixed parameters

$$K_1 = \text{diag}\begin{pmatrix} 10 & 10 & 10 \end{pmatrix}, \; K_2 = \text{diag}\begin{pmatrix} 20 & 20 & 20 \end{pmatrix}.$$

2. Fuzzy and sliding mode control (Fuzzy-SMC) as formulated in [31] with our own fuzzy rule, later included in the last scenario.

3. Our Fuzzy-DSC as designed in Section 3.

The control algorithms are simulated using MATLAB/Simulink, with our architecture illustrated as a block diagram in Fig. 4.
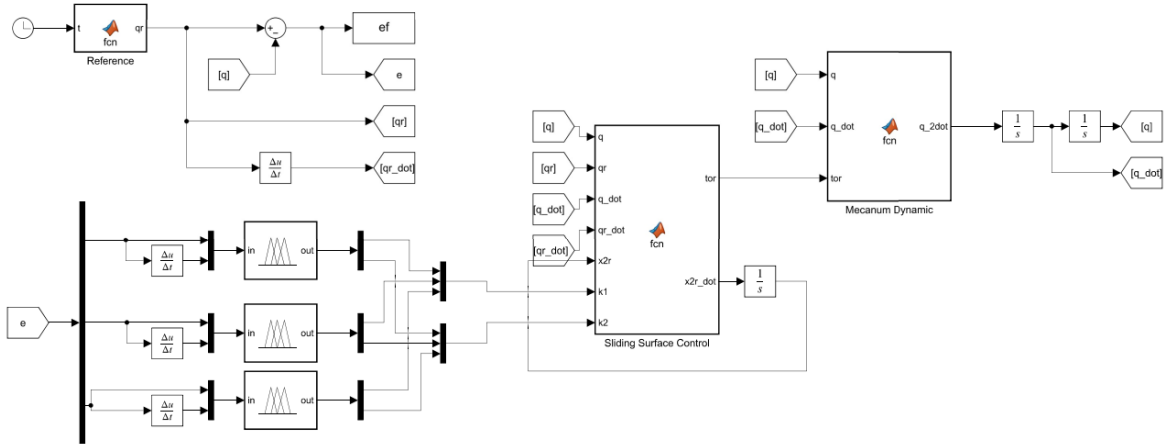

Figure 4: Block diagram of the simulation setup

4.1.   Figure-8 trajectory tracking without mass eccentricity

In the first simulation scenario, the mass of the robot and its center of gravity remain constant $d_1 = d_2 = 0$ and $\Delta m = 0$. The reference trajectory for the robot to follow is defined as follows

$$x = 10\sin(t), \; y = 10\sin(2t), \; \phi_r = \frac{\pi}{4}.$$

As shown in the simulation results in Fig. 5, both control algorithms allow the robot to closely and stably follow the reference trajectory. Furthermore, the Fuzzy-DSC algorithm achieves a faster tracking speed. Fig. 6 illustrates the pose tracking error (a), velocity
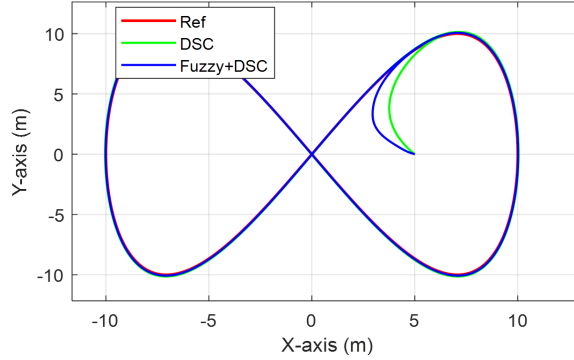

Figure 5: Visualization of figure-8 trajectory without mass eccentricity

tracking error (b), and adaptive parameters (c). Table 2 presents the deviations observed in the calculated trajectory of the robot. The DSC algorithm exhibits larger trajectory tracking errors, particularly at the orbital curves, with a y-axis error is $0.22\,(\text{m})$, the tracking error of the Fuzzy-DSC algorithm is $0.09\,(\text{m})$.
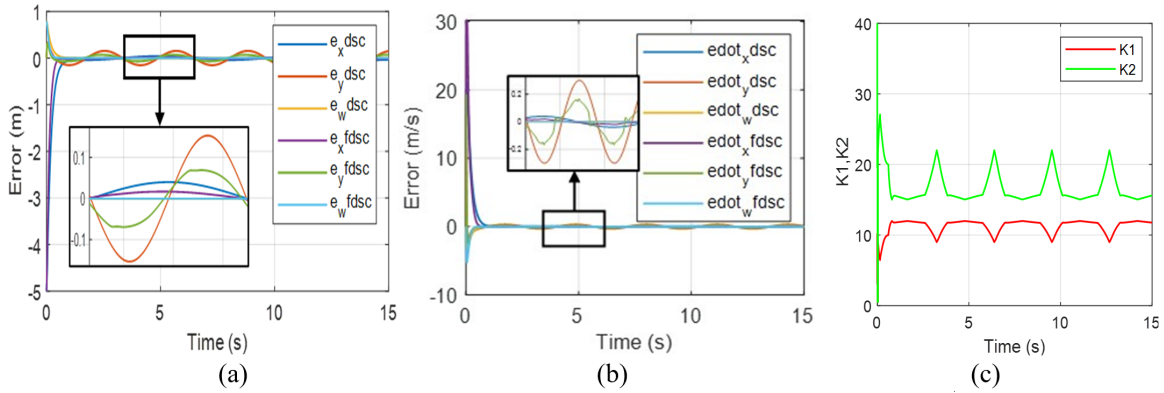

Figure 6: Figure-8 tracking performance without mass eccentricity

Table 2: Deviations in robot trajectory during movement with $d_1 = d_2 = 0$ and $\Delta m = 0$

Times (s)	DSC Error (%)	Fuzzy-DSC Error (%)	x-axis	y-axis
0	100.00%	0.00%	100.00%	0.00%
1	0.97%	3.03%	0.27%	1.34%
2	0.78%	1.40%	0.32%	0.99%
2.5	0.60%	3.05%	0.23%	1.37%
3	0.27%	1.90%	0.09%	0.73%
4	0.49%	2.98%	0.24%	1.39%
5	0.80%	0.60%	0.33%	0.59%

4.2.   Figure-8 trajectory tracking with mass eccentricity

In the second simulation scenario, we change the total mass of the robot and its center of gravity $d_1 = d_2 = 0.1\,(\mathrm{m})$ and $\Delta m = 5\,(\mathrm{kg})$. Fig. 7 shows that when the robot's mass
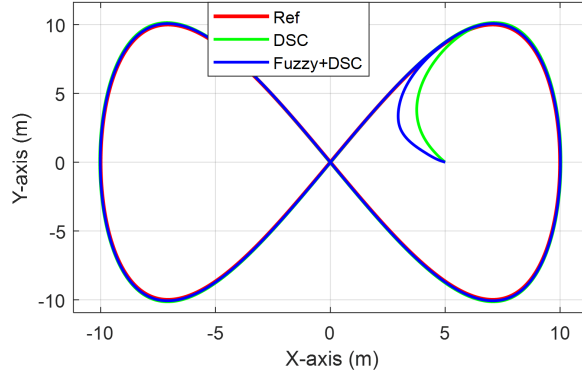

Figure 7: Visualization of figure-8 trajectory with mass eccentricity

and center of gravity change, both algorithms can still control the robot to move along the reference trajectory while maintaining stable operation. However, there is a noticeable deviation, as illustrated in Fig. 8(a), where the DSC algorithm exhibits a larger trajectory tracking error of $0.33\,(\mathrm{m})$. Fig. 8(b) demonstrates that the Fuzzy-DSC algorithm achieves better velocity tracking performance. Furthermore, the Fuzzy-DSC algorithm adjusts the controller parameters K_1, K_2, as shown in Fig. 8(c), to adapt to the model changes, resulting in improved control quality with a smaller tracking error is $0.11\,(\mathrm{m})$.
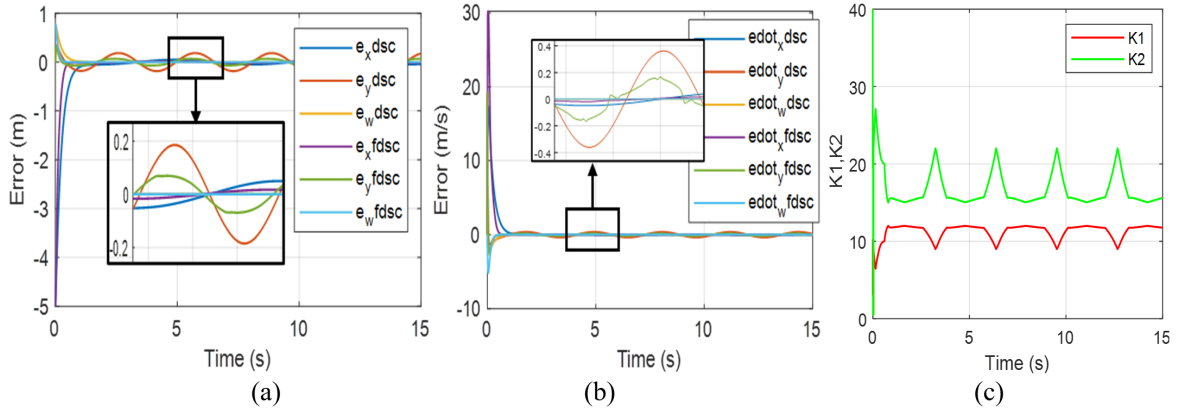

Figure 8: Figure-8 tracking performance with mass eccentricity

Table 3 presents the deviations observed in the calculated trajectory of the robot, considering the eccentricity of the mass. The results indicate an improvement in control quality using the Fuzzy-DSC algorithm.

4.3.   Circular trajectory tracking with mass eccentricity

In the third simulation scenario, we attempt to track a circular trajectory defined as $x_r = 10\sin(\frac{2\pi}{15}t), y_r = 10\cos(\frac{2\pi}{15}t), \phi_r = \frac{\pi}{4}$ to evaluate the Fuzzy-DSC algorithm. Moreover, we

Table 3: Deviations in robot trajectory during movement with $d_1 = d_2 = 0.1\,(\text{m})$ and $\Delta m = 5\,(\text{kg})$

Times (s)	DSC Error (%)	Fuzzy-DSC Error (%)	x-axis	y-axis
0	100.00%	0.00%	100.00%	0.00%
1	1.35%	3.30%	0.28%	1.42%
2	0.87%	1.41%	0.33%	1.00%
2.5	0.67%	3.33%	0.25%	1.39%
3	0.31%	2.18%	0.08%	0.74%
4	0.50%	3.23%	0.25%	1.44%
5	0.89%	0.62%	0.34%	0.61%

compare it with a Fuzzy-SMC baseline, inspired by adaptive sliding mode control algorithms in [10–12]. The results demonstrate that the Fuzzy-DSC algorithm effectively tracks the circular trajectory, demonstrating its robustness and versatility in handling diverse trajectory profiles (Fig. 9). The trajectory errors recorded at various time points in Table 4 indicate that the Fuzzy-DSC algorithm offers better control quality compared to both conventional DSC and Fuzzy-SMC. Meanwhile, in Fig. 9(c), Fuzzy-SMC exhibits significant chattering in velocity tracking. By analyzing the errors between the desired and actual trajectories over time, it becomes evident that the Fuzzy-DSC approach achieves more accurate and consistent path tracking, highlighting its effectiveness in enhancing control performance.
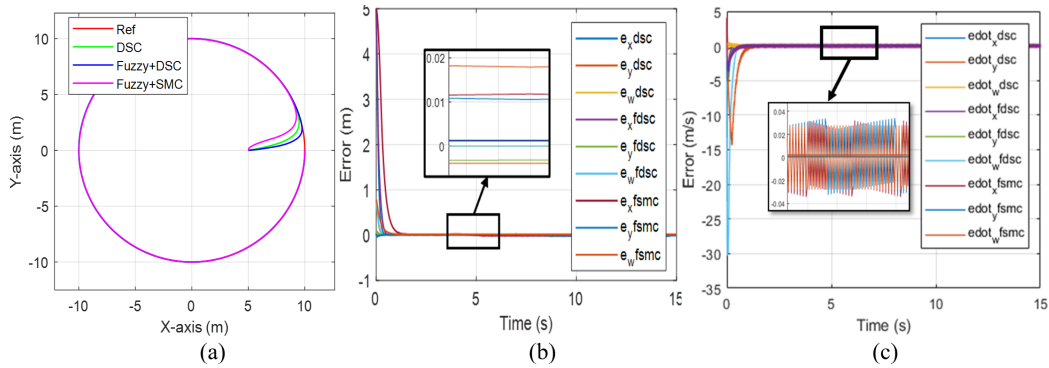

Figure 9: Circular trajectory tracking performance with mass eccentricity

Table 4: Movement error of the robot along a circular trajectory with $d_1 = d_2 = 0.1\,(\text{m})$, $\Delta m = 5\,(\text{kg})$

Times (s)	DSC Error (%)		Fuzzy-DSC Error (%)		Fuzzy-SMC Error (%)	
	x-axis	y-axis	x-axis	y-axis	x-axis	y-axis
0	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
1	0.61%	0.04%	0.06%	0.03%	0.58%	0.05%
2	0.11%	0.11%	0.05%	0.05%	0.13%	0.11%
2.5	0.09%	0.13%	0.04%	0.06%	0.09%	0.11%
3	0.06%	0.15%	0.02%	0.06%	0.07%	0.16%
4	0.04%	0.16%	0.02%	0.07%	0.05%	0.17%
5	0.07%	0.14%	0.03%	0.06%	0.06%	0.16%

Remark 2: Similarly to other controller design methods [10–12], it is essential to tune parameters, such as fuzzy rules in our case, to achieve optimal performance. If the system dynamics becomes more complex (e.g., adding more states or inputs), the fuzzy controller might require a significantly larger rule base, complicating the design and tuning process. A possible direction for further research is the development of self-tuning adaptive mechanisms that could potentially guarantee the stability of the entire system.

Remark 3: In this paper, we do not consider skidding or sliding effects on the movement of four-Mecanum-wheeled robots [32]. These effects are often treated as lumped disturbances and handled using an observer [28]. Instead, we focus on analyzing the stability of the entire dynamical system in the context of challenges such as mass eccentricity. The principles that enable a fuzzy controller to adjust in real-time based on sensor inputs (e.g., wheel speed, surface roughness) and dynamically modify the control inputs are of great interest.

5. CONCLUSION

This paper demonstrates the effectiveness of combining fuzzy adaptive control with dynamic surface control in addressing the challenges of trajectory tracking for four-Mecanum-wheeled mobile robots with mass eccentricity. The proposed approach provides a robust solution for managing the complexities introduced by mass eccentricity, resulting in provable improved stability and accuracy in trajectory tracking, as evidenced by extensive simulations. Future work will focus on refining the proposed control methodology and conducting experiments in real-world robotic systems.

ACKNOWLEDGMENT

REFERENCES

[1] M. Abd Mutalib and N. Azlan, "Prototype development of mecanum wheels mobile robot: A review," *Applied Research and Smart Technology (ARSTech)*, vol. 1, no. 2, pp. 71–82, 2020.

[2] H. Taheri, B. Qiao, and N. Ghaeminezhad, "Kinematic model of a four mecanum wheeled mobile robot," *International Journal of Computer Applications*, vol. 113, no. 3, pp. 6–9, 2015.

[3] M. Abdelrahman *et al.*, "A description of the dynamics of a four wheel mecanum mobile system as a basis for a platform concept for special purpose vehicles for disabled persons," in *58th Ilmenau Scientific Colloquium*, 2014, pp. 1–10.

[4] Y. Li *et al.*, "Modeling and kinematics simulation of a mecanum wheel platform in recurdyn," *Journal of Robotics*, vol. 2018, no. 1, p. 9373580, 2018.

[5] K.-L. Han *et al.*, "Design and control of mobile robot with mecanum wheel," in *ICCAS-SICE: IEEE*, 2009, pp. 2932–2937.

[6] I. Zeidis and K. Zimmermann, "Dynamics of a four-wheeled mobile robot with mecanum wheels," *Journal of Mechanics*, vol. 99, no. 12, p. e201900173, 2019.

[7] N. Tlale and M. de Villiers, "Kinematics and dynamics modelling of a mecanum wheeled mobile platform," in *15th International Conference on Mechatronics and Machine Vision in Practice: IEEE*, 2008, pp. 657–662.

[8] Z. Hendzel, "Modelling of dynamics of a wheeled mobile robot with mecanum wheels with the use of lagrange equations of the second kind," *International Journal of Applied Mechanics and Engineering*, vol. 22, no. 1, pp. 81–99, 2017.

[9] z. Hendzel, "A description of the motion of a mobile robot with mecanum wheels–dynamics," in *Progress in Automation, Robotics and Measurement Techniques*. Springer, 2020, pp. 337–345.

[10] X. Feng and C. Wang, "Robust adaptive terminal sliding mode control of an omnidirectional mobile robot for aircraft skin inspection," *International Journal of Control, Automation and Systems*, vol. 19, no. 2, pp. 1078–1088, 2021.

[11] H.-M. Wu and M. Karkoub, "Frictional forces and torques compensation based cascaded sliding-mode tracking control for an uncertain omnidirectional mobile robot," *Measurement and Control*, vol. 55, no. 3-4, pp. 178–188, 2022.

[12] Z. Song *et al.*, "A trajectory tracking control based on a terminal sliding mode for a compliant robot with nonlinear stiffness joints," *Micromachines*, vol. 13, no. 3, p. 409, 2022.

[13] G. Zhong, C. Wang, and W. Dou, "Fuzzy adaptive pid fast terminal sliding mode controller for a redundant manipulator," *Mechanical Systems and Signal Processing*, vol. 159, p. 107577, 2021.

[14] S. Jeong and D. Chwa, "Sliding-mode-disturbance-observer-based robust tracking control for omnidirectional mobile robots with kinematic and dynamic uncertainties," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 741–752, 2020.

[15] S. Chang, Y. Wang, and Z. Zuo, "Fixed-time formation-containment control for uncertain multi-agent systems with varying gain extended state observer," *Information Sciences*, vol. 612, pp. 759–779, 2022.

[16] C.-H. Kuo, "Trajectory and heading tracking of a mecanum wheeled robot using fuzzy logic control," in *International Conference on Instrumentation, Control and Automation (ICA)*. IEEE, 2016, pp. 54–59.

[17] C. Kumile and N. Tlale, "Intelligent distributed fuzzy logic control system (idflcs) of a mecanum wheeled autonomous guided vehicle," in *International Conference Mechatronics and Automation*. IEEE, 2005, pp. 131–137.

[18] P. Jamali *et al.*, "Software based modeling, simulation and fuzzy control of a mecanum wheeled mobile robot," in *First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2013, pp. 200–204.

[19] D. Swaroop *et al.*, "Dynamic surface control for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1893–1899, 2000.

[20] B. Song, A. Howell, and K. Hedrick, "Dynamic surface control design for a class of nonlinear systems," in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*.   IEEE, 2001, pp. 2797–2802.

[21] D. Tobergte and S. Curtis, *Dynamic Surface Control of Uncertain Nonlinear Systems*, 2013, vol. 53.

[22] S.-H. Yu, C.-H. Hyun, and H. Kang, "Robust dynamic surface tracking control for uncertain wheeled mobile robot with skidding and slipping," in *International Conference on Control and Robotics Engineering (ICCRE)*.   IEEE, 2016, pp. 1–4.

[23] X. Zhang *et al.*, "Acceleration-level pseudo-dynamic visual servoing of mobile robots with backstepping and dynamic surface control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 10, pp. 2071–2081, 2017.

[24] S. Luo *et al.*, "Wheeled mobile robot rbfnn dynamic surface control based on disturbance observer," *International Scholarly Research Notices*, vol. 2014, no. 1, p. 634936, 2014.

[25] K. Shojaei and A. Shahri, "Output feedback tracking control of uncertain non-holonomic wheeled mobile robots: a dynamic surface control approach," *IET Control Theory and Applications*, vol. 6, no. 2, pp. 216–228, 2012.

[26] L.-C. Lin and H.-Y. Shih, "Modeling and adaptive control of an omni-mecanum-wheeled robot," *Intelligent Control and Automation*, vol. 4, no. 2, pp. 166–179, 2013.

[27] F. Becker *et al.*, "An approach to the kinematics and dynamics of a four-wheel mecanum vehicle," *Scientific Journal IFToMM Problems of Mechanics*, vol. 2, pp. 27–37, 2014.

[28] A. Rauniyar *et al.*, "Mewbots: Mecanum-wheeled robots for collaborative manipulation in an obstacle-clustered environment without communication," *Journal of Intelligent and Robotic Systems*, vol. 102, no. 1, p. 3, 2021.

[29] T. Zhao, X. Zou, and S. Dian, "Fixed-time observer-based adaptive fuzzy tracking control for mecanum-wheel mobile robots with guaranteed transient performance," *Nonlinear Dynamics*, pp. 1–17, 2022.

[30] H. Nguyen and M. Sugeno, *Fuzzy Systems: Modeling and Control*.   Springer Science and Business Media, 2012, vol. 2.

[31] D. Nguyen Minh *et al.*, "Research dynamic surface control for four mecanum wheeled mobile robot," *Journal of Military Science and Technology, FEE*, pp. 41–49, 2022.

[32] L. Tagliavini *et al.*, "Wheeled mobile robots: state of the art overview and kinematic comparison among three omnidirectional locomotion strategies," *Journal of Intelligent and Robotic Systems*, vol. 106, no. 3, p. 57, 2022.