INTRODUCTION

1. Urgency of the Thesis

In the 4.0 industrial revolution, the demand for fully automated factories has made mobile robots an essential component. Mobile robots help optimize workflows, minimize human intervention, increase efficiency, and ensure accuracy. Among the common types of mobile robots today, omnidirectional mobile robots have the advantage of moving flexibly in all directions without being dependent on the robot's orientation, making them ideal for working in confined spaces.

Typically, omnidirectional mobile robots use either omni-wheels or mecanum wheels. Among them, mobile robots with mecanum wheels can carry heavy loads, work flexibly in complex, narrow, and crowded environments such as warehouses, hospitals, industrial production areas, or urban environments. However, these environments are often subject to continuously changing factors (obstacles, uneven floors, etc.), especially when the load varies or when situations arise that cause instability (for example, when the robot carries uneven materials). In such cases, the center of gravity of the robot changes, which can significantly affect the movement and trajectory tracking capabilities. Therefore, the research into adaptive control algorithms that enable robots to operate more accurately and stably in real working environments is crucial.

Trajectory tracking control for mecanum-wheeled omnidirectional mobile robots is a critical and urgent problem. It requires systems capable of controlling the robot to maintain its trajectory without deviation or instability, particularly when the robot model is nonlinear with uncertain components influenced by the working environment or when moving on uneven terrain. The development of trajectory tracking control algorithms helps minimize errors and stabilize the operation process. Key research directions include using conventional PID control algorithms, fuzzy PID control, or some studies using sliding mode controllers. Recently, advanced control algorithms such as MPC (Model Predictive Control), LQR (Linear Quadratic Regulator), and machine learning methods have been applied and proven to be effective.

A major challenge during the research is the development of control algorithms that can adapt to unstable environmental factors, such as the uncertain center of gravity of the robot when transporting heavy goods or when moving on uneven terrain. Machine learning, reinforcement learning, and deep learning algorithms based on artificial neural networks have also been researched for trajectory tracking control, achieving better control quality.

With the above trends in mind, the research topic chosen is: "**Research on the development of an adaptive algorithm and reinforcement learning based on Actor-Critic structure for trajectory tracking control of omnidirectional mecanum mobile robots".** The thesis focuses on developing adaptive and reinforcement learning algorithms to improve the quality of trajectory tracking control for four-wheeled mecanum omnidirectional mobile robots with changing center of gravity. The thesis applies traditional control algorithms and machine learning algorithms for comparison and verification of the proposed control algorithm's quality.

2. Objectives of the Thesis

The objective of this thesis is to research adaptive control and reinforcement learning algorithms to improve the quality of trajectory tracking for omnidirectional mecanum robots, under the influence of changing center of gravity and external disturbances. The thesis sets the following main research tasks:

- Study adaptive control algorithms to enhance the quality of trajectory tracking for omnidirectional mecanum robots with changing center of gravity.

- Study the Actor-Critic reinforcement learning algorithm structure for trajectory tracking control of omnidirectional mecanum robots with changing center of gravity.

3. Research Content

- Overview of omnidirectional robots, related research in the world, and the research direction of the thesis.
- Mathematical modeling for four-wheeled mecanum mobile robots with changing center of gravity.

- Apply algorithms for trajectory tracking control for mecanum robots such as PID, SMC, Backstepping-SMC, DSC.
- Propose an adaptive control algorithm based on fuzzy logic systems to improve trajectory tracking quality for omnidirectional mecanum robots with changing center of gravity and external disturbances.
- Propose an Actor-Critic reinforcement learning algorithm for trajectory tracking control of omnidirectional mecanum robots under the influence of external disturbances and model uncertainties.
- Simulate and experiment with the algorithms on a real robot model, then evaluate the quality and practical applicability of the proposed algorithm.

4. Scientific and Practical Significance of the Thesis

Currently, the trajectory tracking control problem for omnidirectional mecanum robots is urgent. This research not only serves industries and manufacturing but also contributes to the development of automation technology, enhancing the application of robots in rescue, security, and autonomous transportation fields.

The thesis proposes an adaptive control method, reinforcement learning based on fuzzy logic rules, and artificial neural networks as a new approach to the trajectory tracking problem for omnidirectional four-wheeled mecanum robots, adapting to the effects of changing center of gravity and uncertain model disturbances.

The research results serve as a scientific basis for practical application, along with building a robot prototype to verify the algorithm, opening up possibilities for practical deployment

5. Contributions of the Thesis

The contributions of the thesis include:

- Proposing a dynamic sliding mode adaptive fuzzy control algorithm to improve the trajectory tracking quality for four-wheeled mecanum autonomous mobile robot with changing center of gravity.
- Proposing an Actor-Critic reinforcement learning algorithm for trajectory tracking control of four-wheeled mecanum autonomous mobile robot with changing center of gravity.

CHAPTER 1. OVERVIEW OF OMNIDIRECTIONAL MOBILE ROBOTS

1.1. Overview of Four-Wheeled Mecanum Omnidirectional Mobile Robots

Mecanum-wheeled omnidirectional mobile robots (FMWR) are designed with four mecanum wheels, each independently driven by its own motor. The wheels are arranged symmetrically in pairs, creating a balanced posture for the robot and ensuring kinematic compatibility for each wheel. Each mecanum wheel is constructed with passive satellite rollers inclined at an angle of 45° to the main wheel axis, in Figure 1.

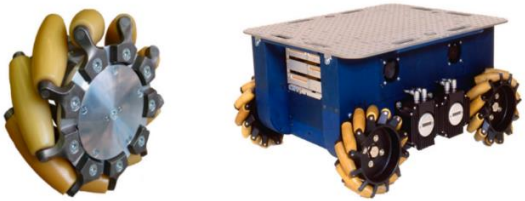


Figure 1.1 Four- Wheeled Mecanum Autonomous Robot Model

When the robot operates, the wheels are driven to rotate in directions perpendicular to the drive axis, and the passive rollers convert part of the longitudinal force into lateral slip force. This allows the robot to move sideways or in any direction independently of its orientation. Thanks to these advantages—flexible movement and high load capacity- FMWRs are widely used in industrial settings, such as warehouse lifting robots, production line transportation robots, and inspection robots in hazardous environments (e.g., radioactive, space, underwater).

In applications like warehouse transportation robots or collaborative arm-integrated robots, carrying additional loads changes the total weight and the center of gravity (CoG) of the robot. A shifted CoG (e.g., when rotating or moving diagonally) may lead to imbalance and affect stability. This highlights the need for control algorithms capable of adapting to such

changes and automatically adjusting parameters (like velocity, direction, and wheel force) to maintain robot stability and accurate trajectory tracking.

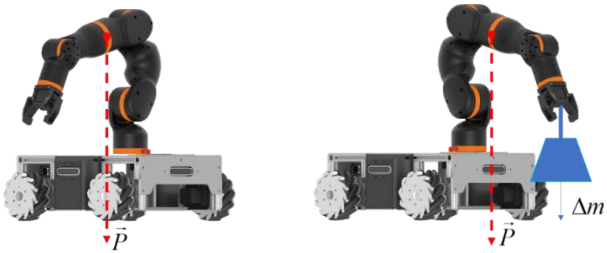


Figure 0.2 The Change in the Center of Mass of a Robot Integrated with a Collaborative Arm

1.2. Kinematic Equations of Mecanum-Wheeled Mobile Robots

The FMWR model is designed with four mecanum wheels arranged symmetrically in pairs. Each wheel is driven independently, allowing the robot to create both longitudinal and lateral forces to move forward, sideways, or in any direction without changing its heading Figure. 3.

The kinematic model in the global coordinate frame is expressed as follows:

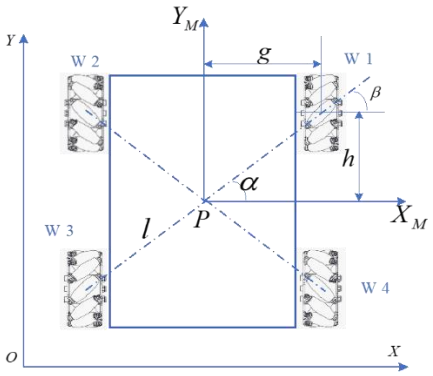


Figure 0.3 Mecanum Omnidirectional Mobile Robot Model

The kinematic equation of the FMWR in the global coordinate system is defined as follows:

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\phi}_R \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \tag{1.1}$$

The kinematic equation written in matrix form is as follows:

$$\dot{\boldsymbol{\eta}}_R = \mathbf{H}(\phi)\dot{\boldsymbol{\eta}} \tag{1.2}$$

Where: $\dot{\boldsymbol{\eta}}_R = \begin{bmatrix} \dot{x}_R & \dot{y}_R & \dot{\phi}_R \end{bmatrix}^T$ - it represents the velocity along the x and y axes and the orientation angle of the robot in the coordinate system attached to the robot. $\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^T$ - it represents the velocity along the x and y axes and the orientation angle of the robot relative to the global coordinate system.

1.3. Dynamic Model of Four-Wheeled Mecanum Mobile Robots

In this model, the actual center of gravity does not coincide with the geometric center of the robot. This deviation affects the robot's dynamics and control characteristics. The FMWR-ME model considers the CoG shift as a variable relative to the robot's frame. The CoG may be fixed or variable due to uncertain or distributed loads Fig.1.4. Where, the center of mass position $P = [x \quad y]^T$ is considered as the center of mass according to the robot's reference frame, and the position $P' = [x - d_1 \quad y - d_2]^T$ is the changing center of mass of the robot, which is considered relative and expressed according to the robot's coordinate system. The center of mass position can be fixed or vary in the case of carrying cargo with a center of mass that is difficult to determine (or uncertain). To ensure the FMWR remains balanced and can operate stably without tipping over during movement, the thesis

limits the offset of the robot's center of mass to not exceed: $d_1 < \frac{1}{2}g$ and $d_2 < \frac{1}{2}h$.

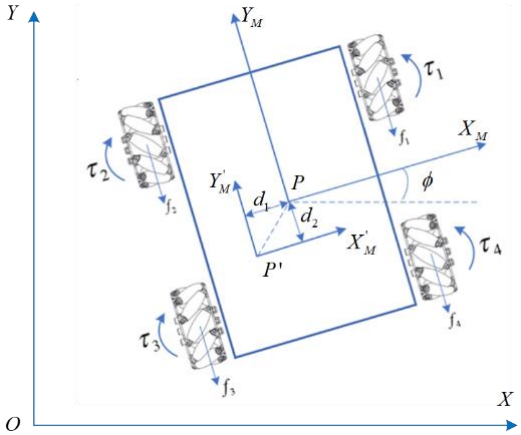


Figure 0.4 The FMWR model takes into account the uncertainty of the center of mass

The dynamic equations can be derived by differentiating the Lagrange equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\eta}}_i}\right) - \frac{\partial L}{\partial \boldsymbol{\eta}_i} = \mathbf{Q}_i, \quad i = 1,2,3 \tag{1.3}$$

where: $\boldsymbol{\eta}_i$ - is the generalized coordinate of the i;

The dynamic equations of the FMWR are written in matrix form as follows:

$$\mathbf{M}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \mathbf{D}\boldsymbol{\delta} = \mathbf{D}\boldsymbol{\tau} \tag{1.4}$$

whre: $\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{bmatrix}^{\mathrm{T}}, \zeta = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 & \zeta_4 \end{bmatrix}^{\mathrm{T}}$

$$\boldsymbol{\delta} = r.U.\zeta\ ;\ \mathbf{U} = \mathrm{diag}\left[\mathrm{sgn}\left(\dot{\psi}_1\right)\quad \mathrm{sgn}\left(\dot{\psi}_2\right)\quad \mathrm{sgn}\left(\dot{\psi}_3\right)\quad \mathrm{sgn}\left(\dot{\psi}_4\right)\right]$$

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

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

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

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

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

The dynamic equation (1.4) of the FMWR includes parameters d_1, d_2 that represent the deviation of the actual center of mass from the geometric center of the robot. These results are used to implement control algorithms to assess the impact of the center of mass deviation on the trajectory tracking control quality. Based on this, adaptive control algorithms are proposed to adjust to changes in model parameters and improve control performance.

CHAPTER 2. TRAJECTORY TRACKING CONTROL ALGORITHM FOR OMNIDIRECTIONAL MECANUM MOBILE ROBOT

After establishing the kinematic and dynamic models for the FMWR, the thesis applies several trajectory tracking control algorithms to evaluate the robot model and analyze the strengths and weaknesses of each method. Based on this, new control algorithms are proposed to enhance effectiveness.

2.1. Dynamic Surface Control Algorithm for FMWR

The DSC algorithm is developed using multi-sliding surface and Backstepping techniques to handle model uncertainties. It reduces chattering by incorporating a low-pass filter. The DSC consists of two main components: a multi-sliding surface (MSS) and a low-pass filter. The MSS processes current system states and filtered control signals to smooth the control actions and avoid chattering. Each sliding surface corresponds to a specific control state.

The control signal of the designed DSC algorithm, the control signal is determined:

$$\boldsymbol{\tau} = -\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}[\mathbf{M}(-\frac{\bar{\mathbf{x}}_2 - \mathbf{x}_{2d}}{\tau'} + K_2\mathbf{S}_2) - \mathbf{C}\dot{\boldsymbol{\eta}} - \mathbf{D}\boldsymbol{\delta}] \quad (2.1)$$

Choose Lyapunov function: $\dot{V} = \mathbf{S}_1^T\dot{\mathbf{S}}_1 + \mathbf{S}_2^T\dot{\mathbf{S}}_2$ (2.2)

Derivative (2.2) and using the inequality we get:

$$\dot{V} \le -(K_1 - \frac{1}{2}I)\mathbf{S}_1^T\mathbf{S}_1 - (K_2 - \frac{1}{2}I)\mathbf{S}_2^T\mathbf{S}_2 \quad (2.3)$$

where K_1 and K_2 are control parameters designed so that the function $\dot{V}$ is negative definite. Hence, if $K_1 > \frac{1}{2}I$ and $K_2 > \frac{1}{2}I$ then, $\dot{V} < 0$ for all $\mathbf{S}_1$

and $\mathbf{S}_2$. Hence, $\mathbf{S}_1 \to 0$ and $\mathbf{S}_2 \to 0$ when time $t \to \infty$, the asymptotic stability of the system follows the Lyapunov stability criterion. In particular, if the deviations are nonzero, the DSC control method will reduce the energy of the system ($\dot{V} < 0$), thereby correcting the deviations. The system if $\mathbf{x}_1$ and $\mathbf{x}_2$ follows the desired trajectory $\mathbf{x}_{1d}$ and $\mathbf{x}_{2d}$, respectively, when time $t \to \infty$.

The designed algorithms are simulated on Matlab/Simulink software to compare and evaluate the results. The simulation parameters of the FMWR model are selected as follows:

$$\begin{aligned} &m = 30\,(kg); m_w = 0.9\,(kg); J_b = 5\,(kg.m^2); \\ &J_w = 0.1\,(kg.m^2); g = 0.2\,(m), h = 0.3\,(m); r = 0.075\,(m); \\ &g = 9.8\,(m/s^2); f = \begin{bmatrix} 0.05 & 0.05 & 0.05 & 0.05 \end{bmatrix}^T (N) \end{aligned} \tag{2.4}$$

- PID algorithm: $K_I = [15,15,15]; K_P = [20,20,20]; K_D = 1$

- SMC algorithm: $\Delta = [5,5,5]; K_1 = [10,10,10]$

- DSC algorithm: $K_1 = [5,5,5]; K_2 = [10,10,10];$

Simulation with FMWR-ME model with eccentricity $d_1 = d_2 = 0.1m$, and variable mass $\Delta m = 5kg$

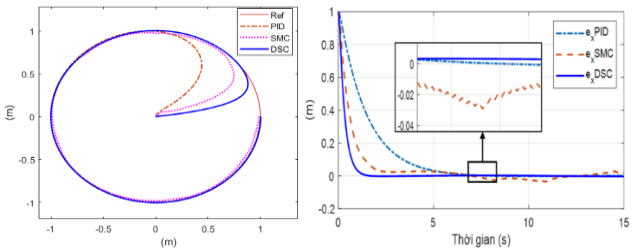


Figure 0Simulation results of circular trajectory tracking

The simulation results show that the DSC control algorithm has control parameters K_1 and K_2 affects the robot's trajectory tracking quality. Specifically, the parameters K_1 determine the speed of approaching the sliding surface, while K_2 affecting the stability of the system on the sliding surface. However, the selection of appropriate parameters to achieve optimal performance is often complicated and depends on many factors. In order to improve the adaptability and enhance the control performance, it is proposed to use a fuzzy tuner to optimize the control parameters. The fuzzy tuner can continuously adjust K_1 and K_2 follow the current state of the robot and the effects of the environment, thereby ensuring better trajectory tracking quality.

2.2. Proposing a fuzzy adaptive tuning dynamic sliding surface control algorithm for omnidirectional mecanum mobile robot (Fuzzy-DSC-FMWR)

2.2.1. Thiết kế thuật toán Fuzzy-DSC-FMWR

A fuzzy logic controller is integrated into the system to automatically adjust the parameters K_1 and K_2 of the DSC control algorithm. During the trajectory movement, the input signals of the regulator include position error $\mathbf{e}$ and velocity error $\dot{\mathbf{e}}$. Based on these signals, the fuzzy logic controller will adjust the parameters to suit each stage of the control process. The fuzzy adaptive tuning dynamic sliding surface control (Fuzzy-DSC) algorithm has been designed to optimize the trajectory tracking control process. The structure diagram of the algorithm is shown in Figure 2.2

The input of the fuzzy controller is the tracking error of the robot trajectory $\mathbf{e}_1$ and the derivative of the error with respect to time $\dot{\mathbf{e}}_1$.

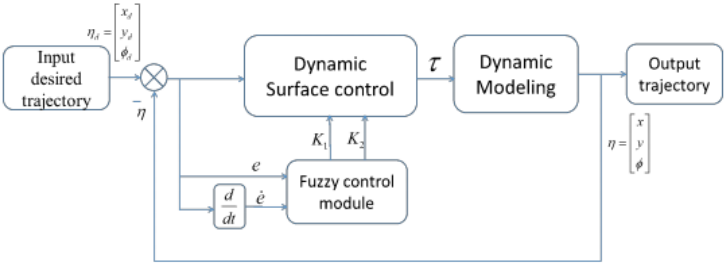


Figure 0.1 Fuzzy-DSC control algorithm structure

2.2.2. Simulation and evaluation of results

Simulation of Fuzzy-DSC and conventional DSC algorithms with a robot model with variable center of gravity and mass (FMWR-ME). Experiment with Gaussian noise: $\in_{Gaus} = normrnd(0, 50, size(t))$

$$\begin{aligned} d_1 &= 0.05sin(0.8t) + 0.025cos(1.5t) + 0.03sin(2.5t) \\ d_2 &= 0.03sin(0.3t) + 0.06cos(1.2t) + 0.05sin(2.8t) + 0.02cos(3.5t) \\ \Delta m &= 5kg \end{aligned} \tag{2.5}$$

The simulation results in Figures 2.3-2.4 show that when noise is introduced into the model, both algorithms control the robot to follow the set trajectory. However, there is still the influence of noise, leading to oscillation.

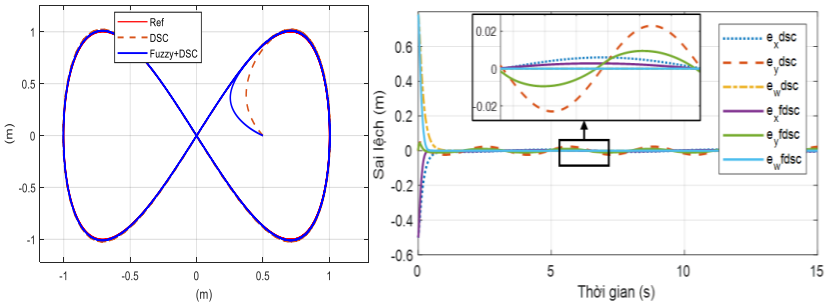


Figure 0.2 FMWR trajectory and deviations

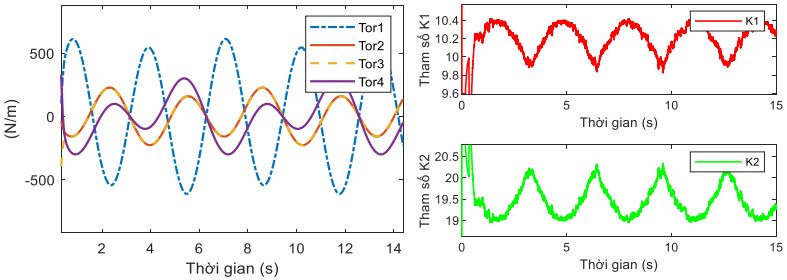


Figure 0.3 Control signals and calibration parameters

Table 0-1: Robot trajectory error evaluation table

Time (s)	**Error DSC (%)**		**Error Fuzzy-DSC (%)**	
	x-axis	y-axis	x-axis	y-axis
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%

Comment: The Fuzzy-DSC algorithm is superior to conventional DSC in handling noise and center of gravity deviation thanks to its ability to self-adjust parameters and flexibly adapt to noise signals. Specifically, the Fuzzy fuzzy controller can quickly change control parameters such as through established fuzzy rules, helping the control system adapt to unexpected changes in the environment or noise. This helps optimize the performance of the control system and maintain stability throughout the robot's movement.

CHAPTER 3. REINFORCEMENT LEARNING ALGORITHM FOR TRAJECTORY TRACKING CONTROL OF OMNIDIRECTIONAL MECANUM MOBILE ROBOT

In chapter 2, the proposed Fuzzy-DSC algorithm is a powerful solution in trajectory tracking control for omnidirectional mobile robots, especially when the system is nonlinear and has uncertain factors. The algorithm has a fast convergence speed and is stable with small disturbances when the parameters and fuzzy rules are designed appropriately. However, the algorithm has limited adaptability in changing environments or unpredictable uncertain disturbances, because the fuzzy rules are often designed in advance and are difficult to automatically adjust in real time. Therefore, the proposed reinforcement learning algorithm applies trajectory tracking control to FMWR, to improve the ability to adapt to changes in changing environments or to withstand uncertain disturbances of the robot model. The reinforcement learning algorithm uses the Actor-Critic structure to help the robot learn from real-life experiences and optimize actions to achieve the trajectory tracking goal without knowing exactly about the uncertain factors of the model.

3.1. Design of the Reinforcement Learning Algorithm for Trajectory Tracking Control of FMWR

Transform the dynamic model of FMWR into the following form:

$$\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u} \tag{3.1}$$

with: $\mathbf{F}(\mathbf{z}) = \begin{pmatrix} f(\mathbf{x}) - h_r(\mathbf{x}_r) + g(\mathbf{x})\boldsymbol{\tau}_r' \\ h_r(\mathbf{x}_r) \end{pmatrix}$, $\mathbf{G}(\mathbf{z}) = \begin{pmatrix} g(\mathbf{x}) \\ \mathbf{0}_{6\times 4} \end{pmatrix}$

To achieve the goal of orbit tracking, we consider the trajectory set for the robot as follows:

$$\dot{\mathbf{x}}_r = \mathbf{h}_r(\mathbf{x}_r) \tag{3.2}$$

where, $\mathbf{x}_r = (\boldsymbol{\eta}_r^T, \mathbf{v}_{qr}^T)^T$ and $\dot{\boldsymbol{\eta}}_r = \dot{\mathbf{v}}_r, h_r(\mathbf{x}_r)$ is a continuous function of the Lipschitz matrix, defined $\mathbf{e} \triangleq \mathbf{x} - \mathbf{x}_r$ as the tracking error, the trajectory tracking speed is described as follows:

$$\dot{\mathbf{e}} = f(\mathbf{x}) + g(\mathbf{x})\boldsymbol{\tau} - h_r(\mathbf{x}_r) = f(\mathbf{x}) - h_r(\mathbf{x}_r) + g(\mathbf{x})\boldsymbol{\tau} + g(\mathbf{x})\mathbf{u} \tag{3.3}$$

In there: $\mathbf{u} = \boldsymbol{\tau} - \boldsymbol{\tau}_r$, $\boldsymbol{\tau}_r(\mathbf{x}_r) = g^+(\mathbf{x}_r)(h_r(\mathbf{x}_r) - f(\mathbf{x}_r))$ (3.4)

Set $\mathbf{z} = (\mathbf{e}^T, \mathbf{x}_r^T)^T$ as new state variable

The control structure diagram is shown in Figure 3.1:

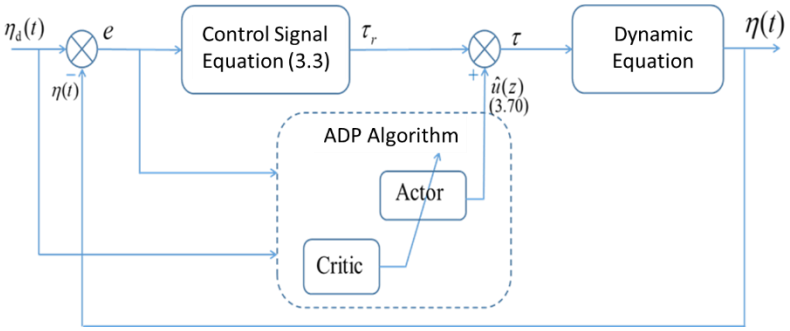


Figure 0.1 FMWR control scheme using actor-critic structure

To optimally control the system (3.1) we consider the following cost function:

$$V(\mathbf{z}(t)) = \int_t^\infty \mathbf{e}^{-\gamma(s-t)}\mathbf{U}(\mathbf{z}(s), \mathbf{u}(s))ds \tag{3.5}$$

The Hamiltonian equation is given by:

$$\mathbf{H}(\mathbf{z}, \mathbf{u}, \nabla\mathbf{V}) = \lambda_M^2(\mathbf{z}) + \mathbf{z}^T\bar{\mathbf{Q}}\mathbf{z} + \mathbf{u}^T(\mathbf{z})\mathbf{R}\mathbf{u}(\mathbf{z}) - \gamma\mathbf{V}(\mathbf{z}) + \nabla\mathbf{V}^T(\mathbf{z})(\mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u}(\mathbf{z})) \tag{3.6}$$

From there we can determine the optimal cost function as follows:

$$\mathbf{V}^*(\mathbf{z}) = \min_{\mathbf{u}\in\pi(\Omega)} \int_t^{\infty} \mathbf{e}^{-\gamma(s-t)} \left(\lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right) \tag{3.7}$$

From this, the optimal control signal is determined as follows:

$$\mathbf{u}^*(\mathbf{z}) = \arg\min_{\mathbf{u}\in\pi(\Omega)} \left[\mathbf{H}\left(\mathbf{z}, \mathbf{u}, \nabla \mathbf{V}^*(\mathbf{z})\right) \right] = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \nabla \mathbf{V}^*(\mathbf{z}) \tag{3.8}$$

To get the optimal control signal, (3.8)we need to solve the extremum problem in the HJB equation. Solving this problem is extremely difficult, so we apply the actor-critic algorithm to solve this equation.

The Actor-Critic algorithm combines with a real-time neural network to perform the approximate solution of the HJB equation. Based on the Weierstrass higher-order approximation theorem, the cost function $\mathbf{V}^*(\mathbf{z})$ can be approximated using a single-layer neural network as follows:

$$\mathbf{V}^*(\mathbf{z}) = \mathbf{W}^T \phi(\mathbf{z}) + \boldsymbol{\varepsilon}_v(\mathbf{z}) \tag{3.9}$$

The optimal control signal becomes:

$$\mathbf{u}^*(\mathbf{z}) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \left(\nabla \phi^T(\mathbf{z}) \mathbf{W} + \nabla \boldsymbol{\varepsilon}_v(\mathbf{z}) \right) \tag{3.10}$$

Based on the formula (3.9) and (3.10)the approximate critic NN for the optimal cost function and the approximate actor NN for the optimal policy are given as follows:

$$\hat{\mathbf{V}}\left(\mathbf{z}, \hat{\mathbf{W}}_c\right) = \hat{\mathbf{W}}_c^T \phi(\mathbf{z}) \tag{3.11}$$

$$\hat{\mathbf{u}}\left(\mathbf{z}, \hat{\mathbf{W}}_a\right) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \nabla \phi^T(\mathbf{z}) \hat{\mathbf{W}}_a \tag{3.12}$$

Where, $\hat{\mathbf{W}}_a, \hat{\mathbf{W}}_c$ is the estimate of the ideal weights $\mathbf{W}$. From which the trajectory tracking controller for FMWR can be obtained as follows:

$$\boldsymbol{\tau} = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{G}^{T}(\mathbf{z})\nabla\phi^{T}(\mathbf{z})\hat{\mathbf{W}}_{a} + g^{+}\left(x_{r}\right)\left(h_{r}\left(x_{r}\right) - f\left(x_{r}\right)\right) \quad (3.13)$$

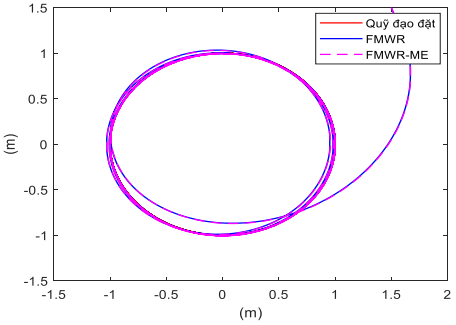
To solve the control optimization problem, we use the least squares method designed to train the ctitic neural network to minimize the integral error $E_{c} = \int_{0}^{t}\delta^{2}\left(s\right)ds$ as follows:

$$\hat{\mathbf{W}}_{c} = -\eta_{c}\boldsymbol{\Gamma}\left(t\right)\frac{\sigma\left(t\right)}{m_{\sigma}^{2}\left(t\right)}\delta \quad (3.14)$$

Determine the estimated bias of the critic weights as $\tilde{\mathbf{W}}_{c} = \mathbf{W} - \hat{\mathbf{W}}_{c}$ and the estimated bias of the actor weights as $\tilde{\mathbf{W}}_{a} = \mathbf{W} - \hat{\mathbf{W}}_{a}$ from which the Bellman bias can be determined:

$$\boldsymbol{\delta} = -\tilde{\mathbf{W}}_{c}^{T}\sigma + \frac{1}{4}\tilde{\mathbf{W}}_{a}^{T}\mathbf{D}_{1}\tilde{\mathbf{W}}_{a} - \boldsymbol{\varepsilon}_{H} \quad (3.15)$$

Simulation with robot model parameters as (2.5)



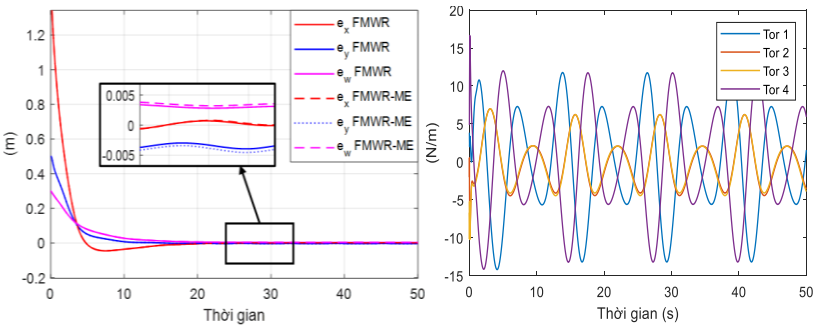


Figure 0.2 Comparison of FMWR and FMWR-ME orbits

During the first period, the robot moves closer to the set trajectory and automatically learns to adjust its actions, helping to follow the trajectory more accurately. During this process, the reinforcement learning algorithm allows the robot to adjust the control weights accordingly, gradually improving its ability to follow the desired trajectory, despite some small deviations, namely [0.00062, −0.00374, 0.00318] m along the x, y and rotation axes. The control signal form is shown in Figure 3.2.

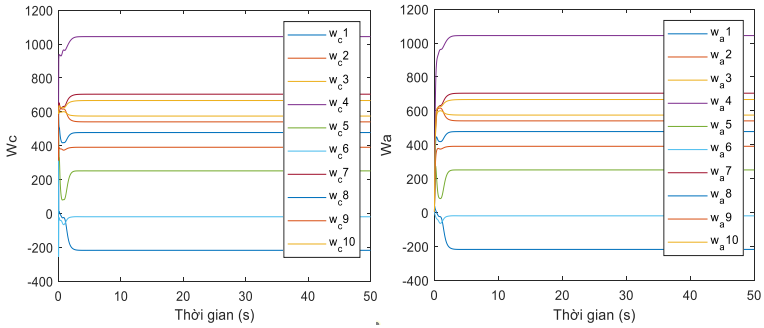


Figure 0.3 Convergence of Actor-Critic Neuron Weights

FMWR control scheme using actor-critic structure

To optimally control the system, (3.1)we consider the following cost function:

$$V(\mathbf{z}(t)) = \int_t^{\infty} \mathbf{e}^{-\gamma(s-t)} \mathbf{U}(\mathbf{z}(s), \mathbf{u}(s)) ds \tag{3.16}$$

The Hamiltonian equation is given by :

$$\mathbf{H}(\mathbf{z}, \mathbf{u}, \nabla \mathbf{V}) = \lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \mathbf{u}^T(\mathbf{z}) \mathbf{R} \mathbf{u}(\mathbf{z}) - \gamma \mathbf{V}(\mathbf{z}) + \nabla \mathbf{V}^T(\mathbf{z})(\mathbf{F}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u}(\mathbf{z}))$$
(3.17)

From there we can determine the optimal cost function as follows:

$$\mathbf{V}^*(\mathbf{z}) = \min_{\mathbf{u} \in \pi(\Omega)} \int_t^{\infty} \mathbf{e}^{-\gamma(s-t)} \left(\lambda_M^2(\mathbf{z}) + \mathbf{z}^T \bar{\mathbf{Q}} \mathbf{z} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right) \tag{3.18}$$

From this, the optimal control signal is determined as follows:

$$\mathbf{u}^*(\mathbf{z}) = \arg \min_{\mathbf{u} \in \pi(\Omega)} \left[\mathbf{H}\left(\mathbf{z}, \mathbf{u}, \nabla \mathbf{V}^*(\mathbf{z})\right) \right] = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \nabla \mathbf{V}^*(\mathbf{z}) \tag{3.19}$$

To get the optimal control signal, (3.8)we need to solve the extremum problem in the HJB equation. Solving this problem is extremely difficult, so we apply the actor-critic algorithm to solve this equation.

The Actor-Critic algorithm combines with a real-time neural network to perform the approximate solution of the HJB equation. Based on the Weierstrass higher-order approximation theorem, the cost function $\mathbf{V}^*(\mathbf{z})$ can be approximated using a single-layer neural network as follows:

$$\mathbf{V}^*(\mathbf{z}) = \mathbf{W}^T \phi(\mathbf{z}) + \boldsymbol{\varepsilon}_v(\mathbf{z}) \tag{3.20}$$

The optimal control signal becomes:

$$\mathbf{u}^*(\mathbf{z}) = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{G}^T(\mathbf{z}) \left(\nabla \phi^T(\mathbf{z}) \mathbf{W} + \nabla \boldsymbol{\varepsilon}_v(\mathbf{z}) \right) \tag{3.21}$$

Based on the formula (3.9)and (3.10)the approximate critic NN for the optimal cost function and the approximate actor NN for the optimal policy are given as follows:

$$\hat{\mathbf{V}}\left(\mathbf{z},\hat{\mathbf{W}}_c\right)=\hat{\mathbf{W}}_c^T\phi(\mathbf{z}) \tag{3.22}$$

$$\hat{\mathbf{u}}\left(\mathbf{z},\hat{\mathbf{W}}_a\right)=-\frac{1}{2}\mathbf{R}^{-1}\mathbf{G}^T(\mathbf{z})\nabla\phi^T(\mathbf{z})\hat{\mathbf{W}}_a \tag{3.23}$$

Where, $\hat{\mathbf{W}}_a,\hat{\mathbf{W}}_c$ is the estimate of the ideal weights $\mathbf{W}$. From which the trajectory tracking controller for FMWR can be obtained as follows:

$$\boldsymbol{\tau}=-\frac{1}{2}\mathbf{R}^{-1}\mathbf{G}^T(\mathbf{z})\nabla\phi^T(\mathbf{z})\hat{\mathbf{W}}_a+g^+\left(x_r\right)\left(h_r\left(x_r\right)-f\left(x_r\right)\right) \tag{3.24}$$

To solve the control optimization problem, we use the least squares method designed to train the ctitic neural network to minimize the integral error $E_c=\int_0^t\delta^2\left(s\right)ds$ as follows:

$$\hat{\mathbf{W}}_c=-\eta_c\boldsymbol{\Gamma}\left(t\right)\frac{\sigma\left(t\right)}{m_\sigma^2\left(t\right)}\delta \tag{3.25}$$

Determine the estimated bias of the critic weights as $\tilde{\mathbf{W}}_c=\mathbf{W}-\hat{\mathbf{W}}_c$ and the estimated bias of the actor weights as $\tilde{\mathbf{W}}_a=\mathbf{W}-\hat{\mathbf{W}}_a$ from which the Bellman bias can be determined: $\boldsymbol{\delta}=-\tilde{\mathbf{W}}_c^T\sigma+\frac{1}{4}\tilde{\mathbf{W}}_a^T\mathbf{D}_1\tilde{\mathbf{W}}_a-\boldsymbol{\varepsilon}_H$ (3.26)

Simulation with robot model parameters as(2.5)

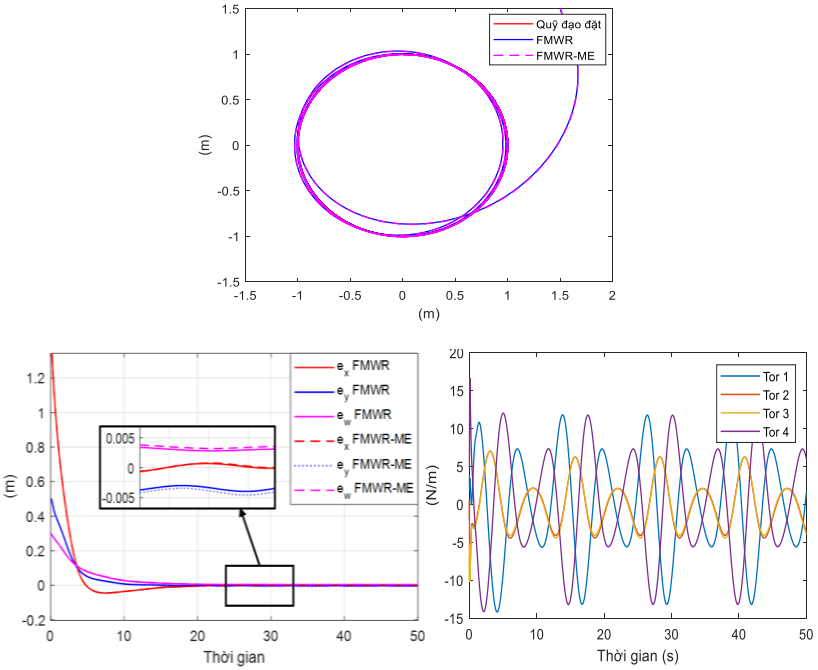


Figure 0. 4 Comparison of FMWR and FMWR-ME orbits

During the first period, the robot moves closer to the set trajectory and automatically learns to adjust its actions, helping to follow the trajectory more accurately. During this process, the reinforcement learning algorithm allows the robot to adjust the control weights accordingly, gradually improving its ability to follow the desired trajectory, despite some small deviations, namely [0.00062, −0.00374, 0.00318] m along the x, y and rotation axes. The control signal form is shown in Figure 3.2.

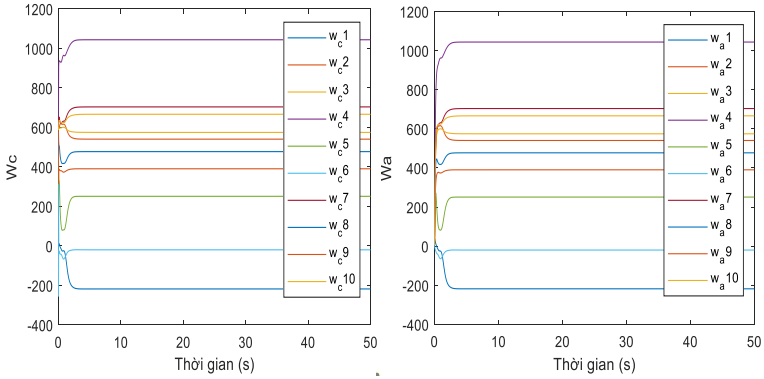


Figure 0. 5 Convergence of Actor-Critic Neuron Weights

After about 6 seconds, the process of self-learning and adapting to the noise signal is completed, then the weights in the NNs network have converged in Figure 3.3 and the robot has started to move stably to follow the set trajectory.

Comments: The results show that the algorithm not only ensures effective trajectory tracking but also has high stability, even when there is the impact of uncertain noise from the model. Furthermore, this algorithm is capable of adapting to changes in the robot's mass and center of gravity.

Comparing the Fuzzy-DSC algorithm and actor-critic reinforcement learning, both algorithms have their own advantages and disadvantages, and are suitable for different applications. If the robot is required to have long-term optimization capabilities and the ability to learn from the environment, the RL-AC algorithm is the right choice, although the learning process can be long and requires a large computer configuration. On the other hand, if immediate stability and strong adaptability to disturbances are required, the Fuzzy-DSC algorithm is a good choice, especially in applications that require high stability and less changing environments.

CHAPTER 4. EXPERIMENTS WITH THE OMNIDIRECTIONAL MECANUM MOBILE ROBOT MODEL

The FMWR model is designed and fabricated for use in the laboratory.

4.1. Design and manufacture of FMWR model

The robot model is designed with total robot weight: 30kg, moving speed: 0.5m/s, load capacity 20kg, operating time 30 minutes.

Control circuit uses Jetson_Tx2, running ROS operating system, STM32

Figure 0.1 FMWR Experimental Model

4.2. Experimental results

The proposed Fuzzy-DSC algorithm is applied to the robot model to run experiments with good results, showing its practical applicability

Figure 0.2 Experimental Results of the FMWR Model

CONCLUSION AND RECOMMENDATIONS

The thesis presents research and proposes algorithms to improve the quality of trajectory tracking control for four-wheeled omnidirectional mobile robots with variable center of gravity. The main contributions of the thesis:

- **Proposed sliding mode control algorithm with fuzzy adaptive tuning to improve trajectory tracking control quality for four-wheeled mecanum autonomous mobile robot with an uncertain center of mass.** The algorithm is designed with the input of the fuzzy tuning unit being the position and velocity errors through the sugeno fuzzy model, to online tune the control parameters of the dynamic sliding surface control algorithm.
- **Proposed reinforcement learning algorithm with Actor-Critic structure applied to trajectory tracking control for four-wheeled mecanum autonomous mobile robot with an uncertain center of mass.** The algorithm uses a neural network with an Actor-Critic structure to approximate the HJB equation. In which, the Actor NNs network is used to approximate the optimal control law, and the Critic NNs are used to approximate the Bellman function. The algorithm is proven to be stable according to the Lyapunov function criterion. The simulation results of the algorithm applied to control trajectory tracking for FMWR show that the robot's trajectory tracking ability is stable even with the robot's center of gravity changing, can withstand the influence of external disturbances and operates stably with small deviations**.**
- **Thesis development direction:** In the future, the research on algorithms can be developed as follows: Developing algorithms with ADP algorithm using multi-layer artificial neural networks. Researching the application of reinforcement learning algorithms to real models and evaluating the results for practical application.