International Journal of Technology 15(5) 1487-1501 (2024)
Received June 2023 / Revised March 2024 / Accepted May 2024

International Journal of Technology

http://ijtech.eng.ui.ac.id

Design of Adjustable Slider Controller in Combination with A* Algorithm in Motion Control for Mobile Robot

Vo Thu Ha[1*], Than Thi Thuong[1], Nguyen Thi Thanh[1]

[1]Control and automation engineering technology, Faculty of Electrical-Automation Engineering, University of Economics - Technology for Industries (UNETI), Hanoi, Vietnam

Abstract. This article presents an Adaptive Fuzzy Logic Dynamic Surface Controller (AFDSC) combined with the A* optimal path-finding algorithm for mobile robots' following trajectory tracking with the nonlinear system changes in robot parameters and is affected by wheel sliding friction disturbances when operating on different terrains. This algorithm is built based on the DSC dynamic sliding surface control algorithm, promoting the effective advantages of DSC and using fuzzy logic to adaptively adjust the coefficients of the virtual control signal and keep The system status signal located on the sliding surface to overcome the instability of DSC when encountering this state. The stability and convergence of the closed-loop system are guaranteed based on Lyapunov analysis. The robot's path planning trajectory is performed by the A* algorithm. At the same time, the content of the article mentions programming and experimental operation for mobile robots using the ROS2 Rolling with Focal (20.04) software operating system on the Jetson Nano 4G embedded computer. The correctness, the proposed controller's effectiveness, and the possibility of practical applications. Orbits are set as two periodic functions of period T as follows. Theoretical and experimental simulation results with position deviation-axis from 0.0038(m) to 0.0063(m), y-axis from 0.0029(m) to 0.0049(m), from 0.0021(rad) to 0.0035(rad). And experimental results with position error in the x-axis from 0.0062(m) to 0.0105(m), y-axis from 0.0042(m) to 0.0069(m), and 0.0031(rad) to 0.0053(rad)).

Keywords: Adaptive controller; Adaptive Fuzzy Dynamic Surface Controlzy (AFDSC); Dynamic Surface Control (DSC); Mobile Robot; Robot operating system (ROS)

1. Introduction

Nowadays, mobile robots have been integrated into various modern intelligent systems, including production systems, logistics, hospitals, smart factories and warehouses. For example, there are unmanned logistics transportation systems that incorporate receiving devices and automatic delivery. The problem of motion control for wheel-type mobile robots has attracted the attention of scientists around the world. Mobile robots are among the systems subject to nonholonomic constraints (Rusdinar *et al.,* 2021; Xin *et al.,* 2016). Furthermore, it is a nonlinear many-input-many-out system (Li, Wang, and Zhu, 2010).

Advancements in control theory have introduced various methods for designing

[*] Corresponding author's email: vothuha.robot.dien.uneti@gmail.com, Tel.: 0913024989

doi: 10.14716/ijtech.v15i5.6527

control laws for mobile robots, including sliding control, (Park *et al.,* 2009; Chwa, 2004) sustainable control (Jiang, 2000), adaptive control (Ye and Wang, 2020; Rubio, Valero, and Llopis-Albert, 2019; Nguyen *et al.,* 2018), backstepping control (Liu *et al.,* 2021; Feng and Wang, 2021), output feedback linearization (Aldo *et al.,* 2021; Rabbani and Memon, 2021). These control laws were designed with the assumption that "the wheel only rolls without slipping." However, in practical applications, the condition that the wheels only roll without slipping can often be violated. That is, wheel slippage has occurred (Zhang *et al.,* 2022; Liu, *et al.,* 2021).

Wheel slippage can occur due to factors like low friction on the floor or centrifugal forces during curved motion. Therefore, in such situations, if control performance is to be improved, it is necessary to design a controller capable of compensating for wheel slippage. The problem of wheel slip compensation control for mobile robots has had many scientists worldwide spend time researching and solving this problem. However, the majority of studies are carried out under the assumption that the slip angle (Hoang and Kang, 2016; Lenain *et al.,* 2010; Low and Wang, 2008) and the friction coefficient between the wheel and the road surface (Elsayed *et al.,* 2019; Chen *et al.,* 2018) are always accurately measured through sensors. Quantities, including translational acceleration, angular acceleration, translational velocity, and angular velocity, can all be easily measured directly through sensors, but the sliding angle and friction coefficient are very small. difficult to measure (Bayar *et al.,* 2016). To measure these signals accurately and reliably, the system must be integrated with complex and expensive sensors (Bayar *et al.,* 2016).

Several control methods have been proposed to address wheel slippage without requiring precise parameter measurements. Instead, the negative effect of wheel slippage on trajectory tracking performance will be compensated indirectly by the controllers. The control law in (Zhang *et al.,* 2020) is designed in the global coordinate system OXY, so it requires measuring velocities in this global system. This velocity measurement task was solved using the supper-twisting observer. The estimation results from this observer may contain errors accumulated during robot operation. So the ability to implement the control method in (Zhang *et al.,* 2020) is still limited. To overcome this drawback, the adaptive sustainable tracking control method is based on the backstepping technique (Xu, Yang, and Gadsden, 2020); creating a reverse impact from kinematics into dynamics) based on a Gaussian wave network for mobile robots to compensate for wheel slippage, model uncertainty, and external noise, show smaller position tracking errors compared to the control method (Zhang *et al.,* 2020) , which has asymptotically converged to zero. However, the disadvantage of this method is that it requires a very large input control signal (torque) at the initial time, the amount of calculation is large and complex, and it takes a lot of time to calculate due to having to calculate the derivative in each iteration step.

Sliding mode controller (SMC) has also been used (Attia and Suan, 2024; Edalati *et al.,* 2018) due to its superior properties when compared to Backstepping, particularly when the system is exposed to noise. Sliding control is preferred for its robustness, rapid response, straightforward control rules, and ease of implementation. Sliding controllers can be used for a wide class of nonlinear systems with uncertain parameters and interference effects. However, the limitation of the SMC algorithm is the chattering phenomenon, and reducing this phenomenon requires the object model to be accurate. This goes against the properties of the robot model, which is parameter uncertainty. To improve control quality in (Gore *et al.,* 2015), the structure and method of building a dynamic sliding surface controller (DSC) were presented. In particular, when the system contains uncertain components, research focuses on improving and developing the multi-sliding surface controller (MSSC). The design method also determines the control signal based on the

Lyapunov control function, so DSC ensures a stable closed system and can adapt to the uncertain composition of the system and deviations within certain limits. The design steps are similar to the Backstepping set design steps; however, to avoid having to take derivatives in the iteration steps for the virtual control signal, DSC has added a low-pass filter, both to get information about medium derivative to filter out high-frequency internal noises appearing in the control object (Qi *et al.,* 2018). To improve control quality, an adaptive controller based on the dynamic sliding surface control (DSC) technique combined with a fuzzy logic system (Wang, Wang, and Han, 2021) is studied because the fuzzy adaptive controller has a simple adjustment mechanism in design and installation (Rahman, Hassan, and Ihsan, 2022). This has opened up a new possible research direction for wheeled mobile robots (WMR) with horizontal sliding components; then integrating embedded programming on the Jetson embedded computer platform and the robot operating system (ROS); the STM32 microcontroller creates test autonomous vehicle system with the A* pathfinding algorithm. For fixed environments (warehouses, farms, factories, etc.), path planning algorithms such as Dijkstra, A*, and D* are commonly used. Because these algorithms have the advantage of finding the fastest path in a static environment, the computational methods of these algorithms are also quite fast (Liu *et al.,* 2023; Alshammrei Boubaker, and Kolsi, 2022; Hou *et al.,* 2022).

In this article, the authors propose to use A*, D*, or Dijkstra search algorithms to plan the path for autonomous robots in a static environment with fixed obstacles. In the article (Chen *et al.,* 2009), the authors also analyze the expansion and development algorithms of the A* algorithm to see the optimal usability of the algorithm in planning the path for the robot when we use it using the A* algorithm. In this paper, the authors will choose algorithm A* to plan the fastest path for an omnidirectional mobile robot in a static environment. This article proposes a new control structure with a kinematic and dynamic model of a mobile robot when sliding sideways using only one control loop, and designs a trajectory tracking controller for self-propelled vehicles based on the algorithm. Dynamic sliding surface control and adaptive control structure based on a fuzzy logic system (AFDSC) with an A* path-finding algorithm to ensure a stable closed system.

There are five main sections on this page. Target research and kinetic and dynamic models are introduced in Sections 1 and 2. Presenting the suggested controller is Section 3. The simulation results of the suggested controller are included in Part 4. The conclusion is the concluding section.

2. Kinetic Model of An Autonomous Vehicle

2.1. Kinetics of autonomous robots

Consider a self-propelled robot as depicted in Figure 1. Where: is the radius of the active wheel; *A* is the midway of two active wheels; *C* is the robot's center of gravity's coordinates; R_a is the distance between the center of gravity's coordinates and the wheel axle; and *2L* is the distance between the two wheels. robot's mass, *m*, and the distance between its center of gravity and its two wheels are *a* respectively.

The position of the robot is determined by the identity vector $q=[x,y,\theta]^T$ (x,y are the point coordinates C).

The state vector can be represented by the following five generic coordinates (1):

$$\xi=\begin{bmatrix} q^T & \varphi^T \end{bmatrix}^T=\begin{bmatrix} x & y & \theta & \varphi_r & \varphi_l \end{bmatrix}^T \tag{1}$$

With: ξ is fund the robot's set direction, $\dot{\varphi}_r,\dot{\varphi}_l$ are angular velocity

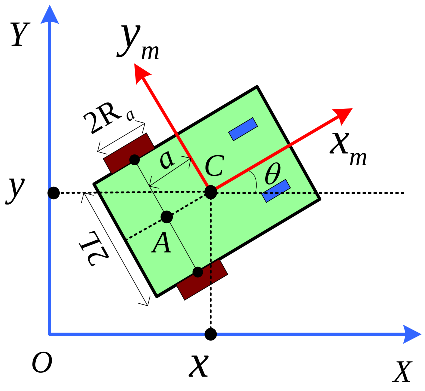


Figure 1 Kinetic relationship of self-propelled robot

According to the reference (Nardênio and Douglas, 2021; Mohareri, 2009) we have the kinematic equation of the robot, which will be written as equation (2) as follows:

$$\dot{q} = \begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix} R_a\dfrac{\dot{\varphi}_r+\dot{\varphi}_l}{2}\cos(\theta)\\ R_a\dfrac{\dot{\varphi}_r+\dot{\varphi}_l}{2}\sin(\theta)\\ \dfrac{R_a}{2L}(\dot{\varphi}_r-\dot{\varphi}_l)\end{bmatrix} \Leftrightarrow \dot{q} = \begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & -a\sin\theta\\ \sin\theta & a\cos\theta\\ 0 & 1\end{bmatrix}\begin{bmatrix}\upsilon\\ \omega\end{bmatrix} = H(q)\vartheta(t) \qquad (2)$$

Where υ the mobile robot's long velocity, angular velocity, and Jacobian matrix that translates its velocity $\vartheta(t)$ in base coordinates ($x_m C y_m$) to its velocity in the inertial reference coordinate system (XOY) are all.

Provide sufficient detail methods to allow the work to be reproduced. Methods already published should be indicated by a reference: only relevant modifications should be described.

2.2.Dynamics of autonomous robots

The dynamic equation of a mobile robot can be characterized as follows, according to documents (Nardênio and Douglas, 2021; Chen *et al.,* 2009)(3).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q,\dot{q})sng(\dot{q}) + \tau_d = -A^T(q)\lambda + B(q)\tau \qquad (3)$$

where (3): $M(q)$ is a positive definite inertial matrix, $C(q,\dot{q})$ is the Centripetal and Coriolis matrix, $G(q,\dot{q})sng(\dot{q})$ is the friction matrix, τ_d is the unknown noise component of the system, $B(q)$ is the input matrix, λ is the Lagrange multiplier, τ Motion control torque for mobile robots, $A(q)$ is the binding matrix, $\dot{q}$ and $\ddot{q}$ represent the generalized velocity and acceleration vectors, respectively.

with: $$M(q)=\begin{bmatrix} m_t & 0 & -m_c a\sin\theta & 0 & 0 & 0 \\ 0 & m_t & m_c a\cos\theta & 0 & 0 & 0 \\ -m_t a\sin\theta & m_c a\cos\theta & I_G & 0 & 0 & 0 \\ 0 & 0 & 0 & 2m_\omega & 0 & 0 \\ 0 & 0 & 0 & 0 & I_\omega & 0 \\ 0 & 0 & 0 & 0 & 0 & I\omega \end{bmatrix},\ B(q)=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tau=\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$C(q,\dot{q})=\begin{bmatrix} 0 & 0 & -m_t a\dot{\theta}\cos\theta & 0 & 0 & 0 \\ 0 & 0 & -m_c a\dot{\theta}\sin\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},\ A(q)^T\lambda=\begin{bmatrix} \cos(\theta) & \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \sin(\theta) & \cos(\theta) \\ -L & L & a \\ 0 & 0 & 1 \\ -r & 0 & 0 \\ 0 & -r & 0 \end{bmatrix}\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix},$$

$$G(q,\dot{q})sng(\dot{q})=\begin{bmatrix} F_1 & F_2 & F_3 & F_4 & 0 & 0 \end{bmatrix}^T$$

where: m_t is the mass of the robot, I is the inertia according to the center of gravity C of the robot, F_1, F_2, F_3, F_4 are the friction force in the direction of translational motion, rotational direction, and horizontal sliding friction, respectively, τ_l and τ_l are the left wheel torque and right wheel torque respectively.

From equation (2), we can prove that the transformation matrix $H(q)$ is the empty space of the constraint matrix, so we will have (4):

$$H(q)A(q)^T=0 \tag{4}$$

We differentiate the equation (2) we have (5):

$$\ddot{q}=\dot{H}(q)\upsilon(t)+H(q)\dot{\upsilon}(t) \tag{5}$$

From there, we have the system's dynamic equation as follows (6):

$$\bar{M}(q)\dot{\upsilon}(t)+\bar{C}(q,\dot{q})\upsilon(t)+\bar{G}(q,\dot{q})\operatorname{sgn}(\upsilon(t))+\bar{\tau}_d=\bar{B}(q)\tau \tag{6}$$

where: $\bar{M}(q)=H(q)M(q)H(q)$ $\bar{C}(q,\dot{q})=H(q)\left[M(q)\dot{H}(q)+C(q,\dot{q})H(q)\right]$, $\bar{B}(q)=H(q)B(q)$ $\bar{G}(q,\dot{q})\operatorname{sgn}(\upsilon(t))=H(q)G(q,\dot{q})sng(\upsilon(t))$ and $\bar{\tau}_d=H(q)\tau_d$.

3. Development of an A* pathfinding algorithm and an adaptive Sliding-fuzzy controller for mobile robots

3.1. Design in sliding mode

According to documentation [3], the recommended state variables are as follows:

$$\begin{cases} x_1=q=\begin{bmatrix} x & y & \theta \end{bmatrix}^T \\ x_2=\upsilon(t)=\begin{bmatrix} \upsilon & \omega \end{bmatrix}^T \end{cases} \tag{7}$$

Combining the kinematics equation (2) and equation (7), we get:

$$\dot{x}_1=H(q)x_2 \tag{8}$$

From equations (6), (7), and (8), we get the state model (9):

$$\begin{cases} \dot{x}_1 = H(q)x_2 \\ \mathrm{M}(q)\dot{x}_2 + C(q,\dot{q})x_2 + \bar{G}(q,\dot{q})\,\mathrm{sgn}(x_2) + \tau_d = B(q)\tau \end{cases} \tag{9}$$

First, the set $e_1 = x_1 - x_{1d}$ is the orbital deviation vector; there $x_{1d} = q_d = \begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$ is the set trajectory and velocity error (10):

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1d} = H(q)x_2 - \dot{x}_{1d} \tag{10}$$

Assume that the control signal is virtual in the design of the DSC controller. is the input to the first-order low-pass filter with the expression (11):

$$\alpha = -H(q)^{-1}\left(c_1 e_1 - \dot{x}_{1d}\right) \tag{11}$$

with $c_1 = \begin{pmatrix} c_{1x} & 0 & 0 \\ 0 & c_{1y} & 0 \\ 0 & 0 & c_{1\theta} \end{pmatrix}$ is the appropriate diagonal constant matrix value whose elements are positive values.

After calculating the virtual control law, α is passed through a first-order low-pass filter to calculate the derivative value for the virtual control signal (12):

$$T\dot{\alpha}_r + \alpha_r = \alpha \tag{12}$$

With T is chosen small enough not to increase the calculation time of DSC (13).

$$\alpha_f(s) = \frac{\alpha(s)}{Ts+1}, \dot{\alpha}_f = \frac{\alpha - \alpha_f}{T} \tag{13}$$

To demonstrate the availability of virtual control signals, choose the first Lyapunov function (14)

$$V_1 = \frac{1}{2} e_1^T e_1 \tag{14}$$

Consider the derivative of V_1 (15):

$$\dot{V}_1 = e_1^T \dot{e}_1 = e_1^T \left(H(q)x_2 - \dot{x}_{1d}\right) = -e_1^T c_1 e_1 + e_1^T \left(c_1 e_1 + H(q)x_2 - \dot{x}_{1d}\right) \tag{15}$$

If you watch, then $\dot{V}_1 = -e_1^T c_1 e_1 + e_1 \dot{c}_1 \left(e_1 - e_1\right)$

It can be seen from expression (15) with virtual control value from (12), $\dot{V}_1 = -e_1^T c_1 e_1 \le 0$ and condition $\dot{V}_1 = -e_1^T c_1 e_1 \le 0$ is satisfied.

Next, the sliding control technique is designed to obtain the control signal of the system and ensure that it achieves the ideal value. The system's virtual control signal bias (16):

$$e_2 = x_2 - \alpha_f \tag{16}$$

Choose slide (17): $$S = \lambda e_1 + H(q)e_2 \tag{17}$$

Where λ is the coefficient of the sliding surface.

The derivative of S is calculated (18):

$$\dot{S} = \lambda\dot{e}_1 + H(q)\dot{e}_2 + \dot{H}e_2 = \lambda\dot{e}_1 + \dot{H}(q)(q)e_2 + H(q)\left(\bar{M}(q)^{-1}\left(-\bar{C}(q,\dot{q})x_2 - \bar{G}(q,\dot{q})\,\mathrm{sgn}(x_2) + \bar{B}(q)\tau\right) - \dot{\alpha}_f\right) \tag{18}$$

As mentioned earlier, one advantage of a DSC controller is its ability to avoid the phenomenon of "term explosion" that occurs when the derivative of a virtual control signal is calculated repeatedly in each cycle. In order to ensure system stability and calculate the control signal, the value of Alpha is derived from the first-order filter (13). Additionally, the second Lyapunov function is selected (19):

$$V_2 = \frac{1}{2} S^T S \tag{19}$$

The system's control signal will be calculated in the form of a sliding controller to increase the system's robustness against noise. Therefore, the control signal will include two components TT, which is the control signal to keep the system state on the sliding surface TR obtained from the condition $\dot{S} = 0$ (20):

$$\tau_{eq} = -\bar{B}(q)^T \left(\bar{B}(q)\bar{B}(q)^T\right)^{-1} \left(\bar{M}(q)\left(H(q)^{-1}\left(\lambda\dot{e}_1 + \dot{H}(q)e_2\right) - \dot{x}_{2d}\right) - \bar{C}(q,\dot{q})x_2 - \bar{G}(q,\dot{q})\,\mathrm{sgn}(x_2)\right) \tag{20}$$

However (20), τ_{eq} Only effective when the system is on a sliding surface. Therefore, the control signal τ_{sw} is used. It is capable of driving the state of the system towards the sliding surface. Expression of τ_{sw} is selected as follows:

$$\tau_{sw} = -\bar{B}(q)^T \left(\bar{B}(q)\bar{B}(q)^T\right)^{-1} \bar{M}(q)H(q)H(q)^{-1}\left(c_2\,\mathrm{sgn}(S) + c_2 S\right) \tag{21}$$

Where (21): $c_2 = \begin{pmatrix} c_{2x} & 0 & 0 \\ 0 & c_{2y} & 0 \\ 0 & 0 & c_{2\theta} \end{pmatrix}$ and $c_3 = \begin{pmatrix} c_{3x} & 0 & 0 \\ 0 & c_{3y} & 0 \\ 0 & 0 & c_{3\theta} \end{pmatrix}$ are positive definite coefficient matrices. Finally, the control signal of the system is the sum of τ_{eq}, τ_{sw} (20), (21):

$$\tau = \tau_{eq} + \tau_{sw} \tag{22}$$

Theorem 3.1: WMR described using the model (6) is controlled by (22) with τ_{eq} được xác định bởi (20) and τ_{sw} (21) ensure the closed system is stable and the tracking error approaches 0.

Proof: The derivative V_2 (19) can be determined (23):

$$\dot{V}_2 = S^T \dot{S} \tag{23}$$

Using (18), $\dot{V}_2$ (23) become

$$\dot{V}_2 = S^T \left(\lambda\dot{e}_1 + \dot{H}(q)e_2 + H(q)\left(\bar{M}(q)^{-1}\left(-\bar{C}(q)x_2 - \bar{G}(q,\dot{q})\,\mathrm{sgn}(x_2) + \bar{B}(q)\tau\right) - \dot{\alpha}_f\right)\right) \tag{24}$$

With control signal (20) và (16) then $x_{2d} = \dot{\alpha}_f$, The derivative of V_2 can be rewritten (24).

$$\dot{V}_2 = -S^T c_2\,\mathrm{sgn}(S) - S^T c_3 S \tag{25}$$

Noise value $\bar{\tau}_d$ is blocked $\left|\bar{\tau}_{\mathrm{d}}\right| \le \lambda$ is an uncertain noise value so it does not appear in the controller expression. By choosing c_2, c_3 (25) appropriately, we have (26)

$$\dot{V} = -S^T c_2\,\mathrm{sgn}(S) - S^T c_3 S \le 0 \tag{26}$$

The advantage of the DSC method is to increase the adaptability of the system and reduce the amount of controller computation. The control signal used contains a sliding component, hence the robust stability of the SMC. The low-pass filter used not only filters out endogenous high-frequency noise but also provides information about the derivative of the virtual control signal. Therefore, calculating the derivative of the virtual control signal becomes unnecessary.

3.2. Adaptive fuzzy-sliding mode Design

During the simulation to look for parameters suitable for slide controllers (Thi *et al.*, 2019), I found that the adaptive slide controller quality (13) largely depends on the selection of the control parameter values, especially c_3. From there, the idea of building a fuzzy logic system to adjust the parameters is c_3 to be formed and implemented in this article. The selected fuzzy model is the. model Sugeno fuzzy consists of two inputs, including bias e and religion function $\dot{e}$. Variable language of e : NB (large negative range), NS (small negative range), Z (zero), PS (small positive range), PB (large positive range; Language variables of: NB, NS, Z, PS, PB Relational $\dot{e}$ function the form belongs: trimf with parameter figure 2, and Fuzzy sets for input language variables e and $\dot{e}$ have the form of triangular and output functions c_3, with output values c_3 VS (very small)=20, S (small)=25, M (average) = 30, B (large) = 35, VB (very large) = 40 (Table 1)

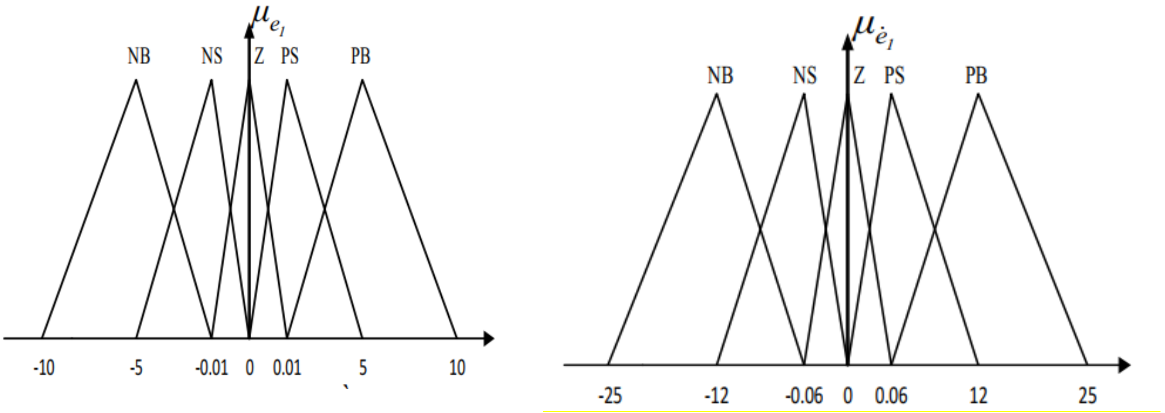


Figure 2 Input fuzzy sets e and $\dot{e}$

Table 1 Fuzzy inference coefficient for output

$\dot{e}_1$	e_1				
	NB	*NS*	*Z*	*PS*	*PB*
NB	M(M)	S(B)	VS(VB)	S(B)	M(M)
NS	B(S)	M(M)	S(B)	M(M)	B(S)
Z	VS(VB)	B(S)	M(M)	B(S)	VS(VB)
PS	B(S)	M(M)	S(B)	M(M)	B(S)
PB	M(M)	S(B)	VS(VB)	S(B)	M(M)

- Fuzzy controller using composition rule SumPROD
- Defuzzification by the formula $c_3 = \sum_{i=1}^{25} \mathrm{w}_i A_i / \sum_{i=1}^{25} \mathrm{w}_i$

From equation (13), we have the control structure diagram as shown in Figure 3 below:

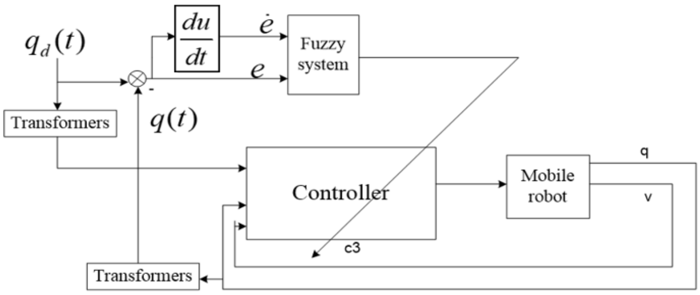
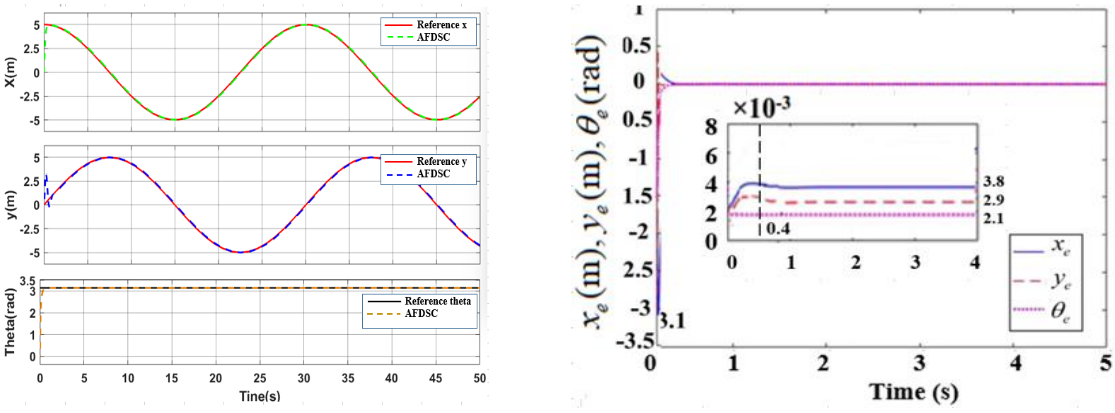


Figure 3 Structure diagram of an adaptive fuzzy sliding controller

4. Simulation of the adaptive fuzzy-sliding controller for autonomous vehicles

Perform the simulation of the proposed controller to control the motion of mobile robot with the following parameters and trajectory: $m = 10.0$ kg; J $= 0.56$kgm^2; $d = 0.3$ m; $r = 0.06$ m, set the trajectory for the robot to move $x_{1d} = \left[1+\cos(\frac{2*\pi t}{5}+\frac{\pi}{2});1+\sin(\frac{2*\pi t}{5}+\frac{\pi}{2});\frac{2*\pi t}{5}+\frac{\pi}{2}\right]$, $\lambda = \text{diag}(10,10,10)$ Perform the simulation using the proposed controller. Motion control for the robot to follow the trajectory in two cases: A) The robot runs normally without load (m=10kg); B) Robot runs with added load resistance (m=30kg). The trajectory tracking response is shown in two cases as shown in Figures 4 and 5 below:

- ✓ *Simulation results on Matlab case A: Simulate with the above parameters*

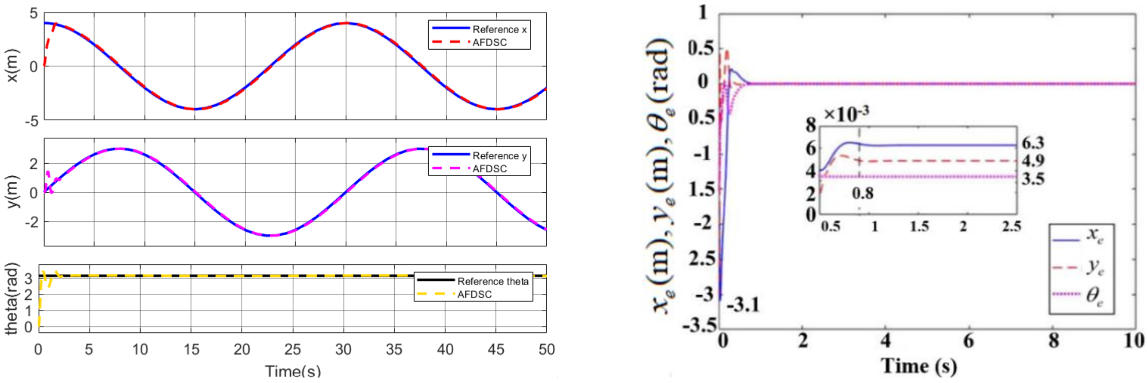


a) Trajectory response in the x-axis, and y-axis

b) Orbital deviation in x, y, and θ

Figure 4 Orbital response and orbital deviation in simulation with m=10k

- ✓ *Simulation results of case B: When the robot's load is increased to 50 kg and there is more impact noise*



a) Trajectory response in the x-axis and y-axis

b) Orbital deviation in x, y, and θ

Figure 5 Trajectory response and orbital deviation when simulating with m=30kg and with noise

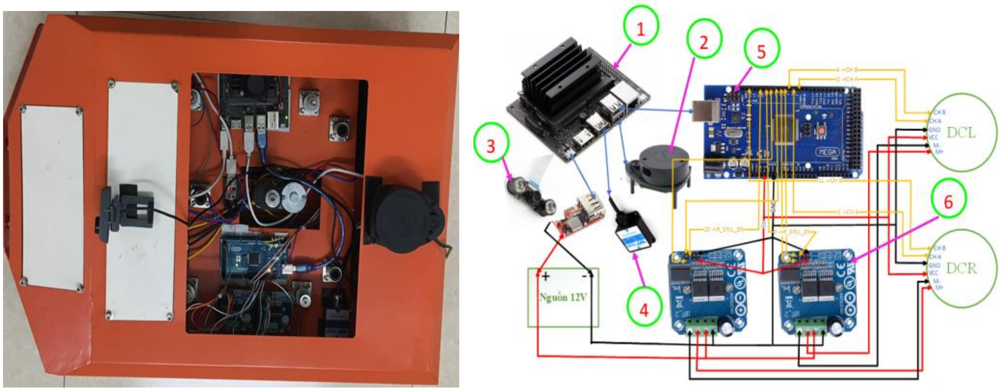
Table 2 Trajectory tracking simulation results of AFDSC controller

A(m=10kg)			B(m=50kg) with extra noise		
x_e (m)	y_e (m)	θ_e (rad)	x_e (m)	y_e (m)	θ_e (rad)
0.0038	0.0029	0.0021	0.0063	0.0049	0.0035

4.1. *Experimenting with operating a mobile robot using an adaptive sliding-fuzzy controller combined with real math A**

✓ *Hardware structure of mobile robot system*

Experiment with the robot system, complete the hardware design and program the control algorithms on ROS with the mobile robot model configured, as shown in Figure 6.



1: Jetson Nano 4G
3:Camera AI
5: STM32 F407
2: Lidar RPLIDAR A1M8 360°
4: IMU HWT901B
6: Driver DC BTS7960 43A

Figure 6 Schematic of the hardware structure to control the self-propelled robot

The mobile robot's steel body is constructed to specifications and measures 70 cm x 50 cm x 38 cm. Its 10-centimeter-wide tires can handle almost any indoor surface. 12000-tick encoders are mounted on two motor shafts. The control circuit hardware structure for the robot is used. Jetson Nano 4G (master) high-performance processor, with the role of central processing, is a specialized high-performance processor for artificial intelligence processing (WHO). The microcontroller circuit (slave) (NCS uses ARM cortex M3 core STM32F407) is the part that receives control signals from Jetson Nano 4G (master). The H-bridge circuit (BTS7960 43A) uses MOSFETs as the power circuit to control the 4-wheel servo motors. The camera has a maximum RGB image resolution of up to 1280 x 720. RPLIDAR A1M8 360° Lidar is a laser scanner, which is combined with the Astra 3D Camera that will send position, direction, and obstacle signals to the Jetson Nano master processor and Sensor. The HWT901B IMU variable sends balanced signals to the Jetson master processor.

✓ *Conduct experiments*

Experiment for moving robots to avoid static obstacles in the robot club room and technology 810_HA10 with scenario Starting from the door to the middle of the room in front of the obstacle is a stack of cartons. Perform robot control in 2 cases: A) Robot runs normally without load (m = 10 kg); B) Robot runs with added load resistance (m=30kg).

To implement the global migration planning approach, ROS provides three interfaces Areas include carrot_planner, navfn, and global_planner. Including travel planning global transfer is the interface used to execute the algorithm $\boldsymbol{A*}$ (Chung, Ojeda, and Borenstein. 2001). Algorithm *A** will be executed by 2 nodes, a map (***Map***) and a location robot (***Robot Pose***), outputting data about the map and robot position Figure 7.

The navigation system designed in this paper combines information from encoder sensors, laser scanning sensors, and IMUs. The structure diagram of the system in ROS is developed using nodes, with each node having its own responsibility, services, and functions. different services, and functions. The information exchanged between each node is communicated through messages, the information transmitted and received according to predefined protocols and a common standard for communicating with external data packets. In addition, the tool supports visual inspection of algorithms and data monitoring and ensures that data is processed in real-time. The system structure diagram is designed on ROS including nodes and the data transmission process is depicted as shown in Figure 8. Running the experiment repeated 3 times for 2 cases (A and B) as suggested in the above scenario, we can calculate the deviation between the real trajectory and the planned trajectory as shown in Table 3.

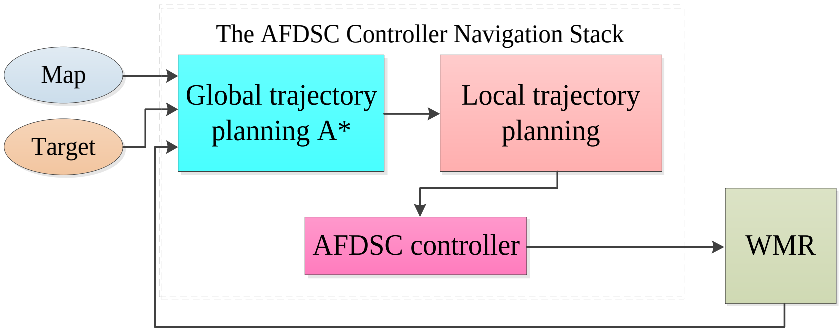

The AFDSC Controller Navigation Stack
Map
Target
Global trajectory planning A*
Local trajectory planning
AFDSC controller
WMR

Figure 7 WMR control structure diagram with AFDSC controller combined with A* algorithm

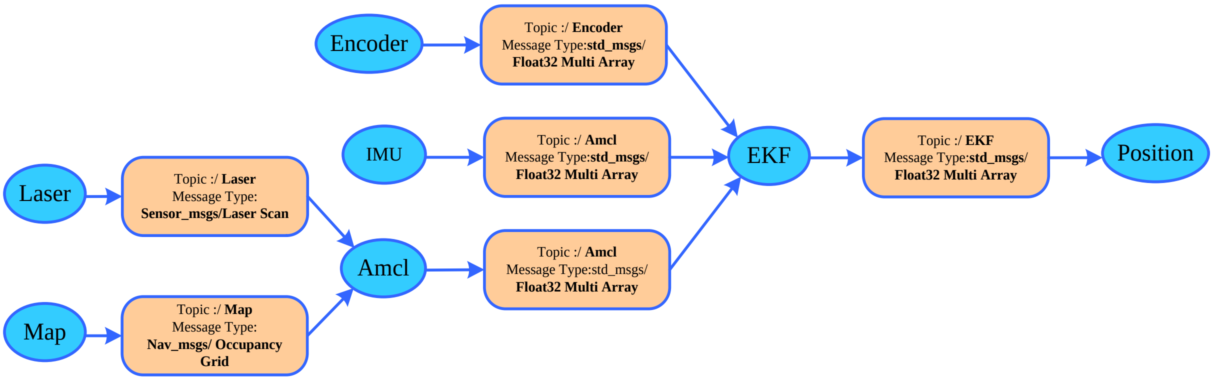
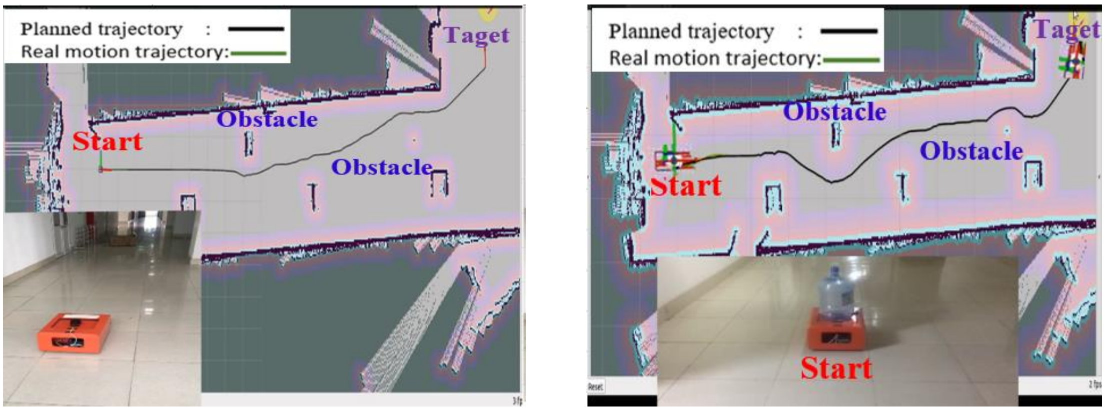

Encoder
Topic :/ Encoder Message Type:std_msgs/ Float32 Multi Array
IMU
Topic :/ Amcl Message Type:std_msgs/ Float32 Multi Array
EKF
Topic :/ EKF Message Type:std_msgs/ Float32 Multi Array
Position
Laser
Topic :/ Laser Message Type: Sensor_msgs/Laser Scan
Amcl
Topic :/ Amcl Message Type:std_msgs/ Float32 Multi Array
Map
Topic :/ Map Message Type: Nav_msgs/ Occupancy Grid

Figure 8 Map button and location used in ROS



(a) (b)

Figure 9 (a) Experimental image of mobile robot motion without load (m = 10 kg) and (b) Experimental image of mobile robot movement when there is an additional load (m = 30 kg)

Table 3 Trajectory deviation results when running experiments

Run times	A			Obstacle avoidance time (s)	B			Obstacle avoidance time(s)
	x_e (m)	y_e (m)	θ_e (rad)		x_e (m)	y_e (m)	θ_e (rad)	
1	0.0062	0.0045	0.0037	0.8102	0.0102	0.0075	0.0053	1.0032
2	0.0065	0.0051	0.0031	0.7834	0.0095	0.0081	0.0049	1.2143
3	0.0071	0.0042	0.0040	0.8621	0.0105	0.0069	0.0051	1.7302

5. Evaluate simulation results and experimental results

The results show in Figure 9(a), 9(b), and Table 2 that in both cases, the controller is guaranteed to follow the set trajectory, but the setting time and the error value are slightly different. Specifically, with the same controller as well as controller parameters, in the case of no external interference, the orbital response approaches the set orbit after about 0.4 seconds with a very small deviation of 0.0038 (m). Meanwhile, with changing model parameters and adding uncertainty, the system is only stable after 0.8 seconds and the determination error is 0.0063 (m). Whenever there is a change in the bias and the bias derivative, the fuzzy rules are specifically designed to update the parameters of DSC online. The newly proposed AFDSC set guarantees improved quality in tracking the system's preset orbit. It enables faster attainment of the set orbit and reduces the tracking error as well. However, the model deviation is large and contains many factors uncertainty and noise. These factors, which affect the robot with large amplitudes, mean that AFDSC no longer ensures quality. This will be the next research direction of the author group.

The results are shown in Table 2. The goal of the experiment was to successfully verify the proposed algorithms through experimentation. To achieve this, the robot is equipped with high-performance hardware and processing control circuits, as well as programming support software based on the ROS robot operating system. Synchronously design and manufacture hardware and peripheral devices, manufacture electronic circuits, control, and peripheral communication to be fast and strong enough to be expandable and upgradeable. Design structure, program, install researched algorithms for robots, run tests, and evaluate results. The experimental results show the correctness of the theoretical analysis, the effectiveness of the proposed controller, and the possibility of practical application. Experimental results show that when the robot encounters an obstacle, it takes about 0.7834 to 1.7302 s to change the local trajectory to avoid the obstacle. With the obstacle fixed in the corridor, it shows the robot's responsiveness when moving flexibly to avoid obstructions, calculate the optimal local trajectory, and navigate along the local trajectory new with the allowed distance and distance and the movement follows the specified trajectory with the position error in the x-axis from 0.0062 to 0.0105 m, the y-axis from 0.0042 to 0.0069 m, and from 0.0031 to 0.0053 rad.

6. Conclusions

This article successfully achieves the goal of researching and proposing an algorithm based on the DSC (Dynamic Sliding Surface Control) algorithm for differential self-propelled robots navigating uncertain routes in a non-linear system. The algorithm focuses on improving tracking quality and adapting to changes in robot parameters, as the robot interacts with various objects and environments, and operates under noise on different terrains. This algorithm takes advantage of efficiency DSC uses fuzzy logic to adaptively adjust the coefficients of the virtual control signal and keeps the system's status signal on the sliding surface to overcome the disadvantages. DSC instability when encountering this state. This algorithm has not been installed on any robot before at home or abroad, has high flexibility, a simple structure, and is easy to program and install on the microprocessor. It can adapt and Maximize efficiency by using the DSC platform it is especially suitable for differential self-propelled robot models based on simulation results and experimental runs. The next research direction of the authors is to continue researching and improving adaptive algorithms when robots move on sloping terrain (3D), as well as have movement patterns suitable for different types of terrain with different complexities when operating in actual applications. Optimize processing time, apply the ROS robot programming operating system for intelligent programming, improving the system's tracking speed.

References

Aldo, J., Vicente, P., Anand, S., Juan, D., 2021. Adaptive Fuzzy Velocity Field Control for Navigation of Nonholonomic Mobile Robots. *Journal of Intelligent & Robotic Systems,* Volume 101, p. 8

Alshammrei, S., Boubaker, S., Kolsi, L., 2022. Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance. *Computers, Materials & Continua*, Volume 72(3), pp. 5939–5954

Attia, H., Suan, S.T.K., 2024. Robust Sliding Mode Controller Design for Boost Converter Applications. *International Journal of Technology,* Volume 15(3), pp. 481–491

Bayar, G., Bergerman, M., Konukseven, E., Koku, A.B., 2016. Improving The Trajectory Tracking Performance of Autonomous Orchard Vehicles Using Wheel Slip Compensation. *Biosystems Engineering,* Volume 146, p. 149–164

Chen, C., Gao, H., Ding, L., Li, W., Yu, H., Deng, Z., 2018. Trajectory Tracking Control of WMRs with Lateral And Longitudinal Slippage Based on Active Disturbance Rejection Control. *Robotics and Autonomous Systems,* Volume 107, pp. 236–245

Chen, C., Li, T., Yeh, Y., Chang, C., 2009. Design and Implementation of An Adaptive Sliding-Mode Dynamic Controller For Wheeled Mobile Robots. *Mechatronics,* Volume19(2), pp. 156–166

Chung, H., Ojeda, L., Borenstein, J., 2001. Sensor Fusion for Mobile Robot Dead-Reckoning With a Precision-Calibrated Fiber Optic Gyroscope. *In:* Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, Volume 4, pp. 3588–3593

Chwa, D.K., 2004. Sliding-Mode Tracking Control Of Nonholonomic Wheeled Mobile Robots In Polar Coordinates. *IEEE Transactions on Control Systems Technology*, Volume 12(4), pp. 637–644

Edalati, L., Sedigh, A.K., Shooredeli, M.A., Moarefianpour, A., 2018. Adaptive Fuzzy Dynamic Surface Control of Nonlinear Systems With Input Saturation and Time-Varying Output Constraints. *Mechanical Systems and Signal Process,* Volume 100, p. 311–329

Elsayed, M., Soe, M.T., Kit, W.W., Abdalla, H., 2019. An Innovative Approach to Developing a 3D Virtual Map Creator using an Ultrasonic Sensor Array*. International Journal of Technology,* Volume 10(7), pp. 1344–1354

Feng, X., Wang, C., 2021. Robust Adaptive Terminal Sliding Mode Control of an Omnidirectional Mobile Robot for Aircraft Skin Inspection. *International Journal of Control, Automation and Systems,* Volume 19, pp. 1078–1088

Gore, R., Reynolds Jr, P.F., Kamensky, D., Diallo, S., Padilla, J., 2015. Statistical Debugging For Simulations. *ACM Transactions on Modeling and Computer Simulation (TOMACS),* Volume 25(3), pp. 1–26

Hoang, N., Kang, H., 2016. Neural Network-Based Adaptive Tracking Control of Mobile Robots in The Presence of Wheel Slip and External Disturbance Force. *Neurocomputing,* Volume 188, pp. 12–22

Hou, Y., Gao, H., Wang, Z., Du, C., 2022. Path Planning for Mobile Robots Based on Improved A* Algorithm. *In:* International Conference on Neural Computing for Advanced Applications, Volume 1637, pp. 169–183

Jiang, Z.P., 2000. Robust Exponential Regulation of Nonholonomic Systems With Uncertainties. *Automatic,* Volume 36(2), pp. 189–209

Lenain, R., Thuilot, B., Cariou, C., Martinet, P., 2010. Mixed Kinematic and Dynamic Sideslip Angle Observer for Accurate Control of Fast Off-Road Mobile Robots. *Journal of Field Robotics,* Volume 27, pp. 181–196

Li, Y., Wang, Z., Zhu, L., 2010. Adaptive Neural Network PID Sliding Mode Dynamic Control of Nonholonomic Mobile Robot. *In:* Proceedings of the 2010 IEEE International Conference on Information and Automation, Harbin, China, pp. 753–757

Liu, C., Xie, S., Sui, X., Huang, Y., Ma, X., Guo, N., Yang, F., 2023. PRM-D* Method for Mobile Robot Path Planning. *Sensors,* Volume 23(7), p. 3512

Liu, J., Wang, Z., Zhang, L., Walker, P., 2021. Sideslip Angle Estimation of Ground Vehicles: A Comparative Study. *IET Control Theory Application,* Volume 14, pp. 3490–3505

Low, C.B., Wang, D., 2008. GPS-Based Tracking Control For A Car-Like Wheeled Mobile Robot With Skidding and Slipping. *IEEE/ASME Transactions on Mechatronics,* Volume 13(4), p. 480–484.

Mohareri, O., 2009. *Mobile Robot Trajectory Tracking Using Neural Networks.* Doctoral Dissertation, American University of Rajah

Nardênio, AM., Douglas, WB., 2021. *Wheeled Mobile Robot Control: Theory, Simulation, and Experimentation.* Springer

Nguyen, T., Nguyentien, K., Do, T., Pham, T., 2018. Neural Network-based Adaptive Sliding Mode Control Method for Tracking of a Nonholonomic Wheeled Mobile Robot with Unknown Wheel Slips, Model Uncertainties, and Unknown Bounded External Disturbances. *Acta Polytechnica Hungarica,* Volume 15, pp. 103–123

Park, B.S., Yoo, S.J., Park, J.B., Choi, Y.H., 2009. Adaptive Neural Sliding Mode Control Of Nonholonomic Wheeled Mobile Robots With Model Uncertainty. *IEEE Transactions on Control Systems Technology*, Volume 17(1), pp. 207–214

Qi, S., Zhang, D., Guo, L., Wu, L., 2018. Adaptive Dynamic Surface Control of Nonlinear Switched Systems with Prescribed Performance. *Journal of Dynamical and Control Systems,* Volume 24, pp. 269–286

Rabbani, M.J., Memon, A.Y., 2021. Trajectory Tracking and Stabilization of Nonholonomic Wheeled Mobile Robot Using Recursive Integral Backstepping Control. *Electronics,* Volume 10(16), pp. 1–22

Rahman, A., Hassan, N., Ihsan, S.I., 2022. Fuzzy Logic Controlled Two Speed Electromagnetic Gearbox for Electric Vehicle. International Journal of Technology, Volume 13(2), pp. 297–309

Rubio, F., Valero, F., Llopis-Albert, C., 2019. A Review of Mobile Robots: Concepts, Methods, Theoretical Framework, and Applications. *Sage Journal,* Volume 16(2), p. 1729881419839596

Rusdinar, A., Purnama, I., Fuadi, A.Z., Adiluhung, H., Wicaksono, M., Risnanda, Ningrum, R.A., 2021. Automated Ultraviolet C Light Mobile Robot for Room Sterilization and Disinfection. *International Journal of Technology,* Volume 12(4), pp. 854–864

Thi, K.D.H., Nguyen, M.C., Vo, H.T., Nguyen, D.D., Bui, A.D., 2019. Trajectory Tracking Control For Four-Wheeled Omnidirectional Mobile Robot Using Backstepping Technique Aggregated With Sliding Mode Control. *In*: 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), pp. 131–134

Wang, C., Wang, D., Han, Y., 2021. Neural Network Based Adaptive Dynamic Surface Control for Omnidirectional Mobile Robots Tracking Control with Full-State Constraints and Input Saturation. *International Journal of Control, Automation and Systems,* Volume 19(12), pp. s4067–s4077

Xin, L., Wang, Q., She, J., Li, Y., 2016. Robust Adaptive Tracking Control Of Wheeled Mobile Robot. *Robotics and Autonomous Systems,* Volume 78, pp. 36–48

Xu, Z., Yang, S.X., Gadsden, S.A., 2020. Enhanced Bioinspired Backstepping Control for a Mobile Robot with Unscented Kalman Filter. *IEEE Access*, Volume 8, pp. 125899-125908

Ye, H., Wang, S., 2020. Trajectory Tracking Control for Nonholonomic Wheeled Mobile Robots with External Disturbances and Parameter Uncertainties. *International Journal of Control, Automation and Systems,* Volume 18, pp. 3015–3022

Zhang, J.J., Fang, Z.L., Zhang, Z.Q., Gao, R.Z., Zhang, S.B., 2022. Trajectory Tracking Control of Nonholonomic Wheeled Mobile Robots Using Model Predictive Control Subjected to Lyapunov-Based Input Constraints. *International Journal of Control, Automation and Systems,* Volume 20(5), pp. 1640–1651

Zhang, L., Chen, H., Huang, Y., Guo, H., Sun, H., Ding, H., Wang, N., 2020. Model Predictive Control For Integrated Longitudinal and Lateral Stability of Electric Vehicles With In-Wheel Motors. Emerg. *IET Control Theory & Applications,* Volume 14(18), pp. 2741–2751