

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications



ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Adaptive sliding mode based fault tolerant control of wheeled mobile robots

Mustafa Ayyıldız & Umut Tilki

To cite this article: Mustafa Ayyıldız & Umut Tilki (2023) Adaptive sliding mode based fault tolerant control of wheeled mobile robots, *Automatika*, 64:3, 467-483, DOI: 10.1080/00051144.2023.2190866

To link to this article: <https://doi.org/10.1080/00051144.2023.2190866>



© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 18 Mar 2023.



Submit your article to this journal [↗](#)



Article views: 1131



View related articles [↗](#)



View Crossmark data [↗](#)



Adaptive sliding mode based fault tolerant control of wheeled mobile robots

Mustafa Ayyıldız ^a and Umut Tilki ^b

^aElectrical and Electronics Engineering Department, Akdeniz University, Antalya, Turkey; ^bElectrical and Electronics Engineering Department, Suleyman Demirel University, Isparta, Turkey

ABSTRACT

In this paper, we propose an adaptive sliding mode-based fault tolerant control for mobile robots. While a mobile robot is tracking a given trajectory, several fault cases may occur, such as sensor model and controller faults, changes in the dynamic equation due to robot body shape or weight changes, and loss of actuator effectiveness. Disturbance signals are caused by the actuator faults and, for various reasons, can be considered the primary issue for the robots. In real-time applications, the Sliding Mode Controller (SMC) is insufficient if the robot parameters are unknown, the robot model is non-linear, and the overall system is subject to disturbances. An adaptive law is used to support the SMC to maintain the sliding surface and solve the problems of unknown system parameters, actuator faults, and disturbances. Besides SMC, the kinematic controller is also used, and its gain values are optimized using a neural network and a kinematic controller. The stability of the overall system is proven by using the Lyapunov theory. Besides actuator faults, the system is disturbed by defining a disturbance signal, which is added to the control signals. To show the effectiveness of the proposed controller, it is compared with traditional SMC and PID.

ARTICLE HISTORY

Received 19 September 2021
Accepted 8 March 2023

KEYWORDS

Adaptive fault tolerant control; neural network based adaptive backstepping control; adaptive sliding mode control; PID control; trajectory tracking

Introduction

Autonomous systems, including mobile robots, are widely used for many different tasks, such as search and rescue, monitoring, human–robot interaction tasks, etc. As the usage of these systems increases, the expected performance of these systems also increases. There are two major categories to consider for mobile robots: non-holonomic and holonomic. Wheeled Mobile Robots (WMR) are electromechanical systems that use activation torques to drive their wheels and are classified as non-holonomic. Autonomous systems, including mobile robots, are vulnerable to faults and external disturbances. Actuator, sensor, and controller (system) faults are the three major categories into which these faults can be divided. These faults can be amplified during the process of the system. It is crucial that the robot be able to behave tolerantly in the event of actuation and/or sensor faults. Therefore, Fault-Tolerant Control (FTC) systems are widely used today in a variety of fields, including the automotive and electronic industries, unmanned vehicle control, and even space research [1].

From its initial position and orientation, the WMR is expected to reach its target position and orientation. In order to accomplish this, one of the important aspects is the Trajectory Tracking (TT) control system. Thanks to the TT control, WMR is able to maintain track of its reference trajectory even if it changes over time [2].

In TT, the WMR must advance from the initial position point until it reaches the reference trajectory line at high torque. After obtaining the reference trajectory, WMR should reduce its torque and maintain a stable and precise trajectory track [3]. Such a TT controller was proposed by Kanayama et al. [4]. The stability of the proposed system and the control rule were proved using the Lyapunov approach. In addition, this study examined the effects of kinematic controller gains on the system state.

The Adaptive Sliding Mode Control (ASMC) was proposed as an FTC system for the TT problem, considering the uncertainties, nonlinear dynamic model, and artificial noises by Mevo et al. [5]. In that study, a nonlinear dynamic model structure was applied, and the designed controller overcame the uncertainty of robot parameters, as in Refs. [6] and [7]. ASMC reduced actuator faults in a Differential Drive Mobile Robot (DDMR) by using the sliding surface diagram for Loss of Effectiveness (LOE) actuator fault as suggested in Ref. [1]. Also, a similar FTC scheme was used in Jin et al. [8]. Actuator faults and unknown parameters degrade the driving performance of the mobile robot. To achieve asymptotic system stability for partial LOE and bias-actuator faults, novel adaptive fault-tolerant control strategies were developed. To address faults of partial loss of control effectiveness in DDMR, sliding mode control [9–11], model predictive control [12],

CONTACT Umut Tilki umuttilki@sdu.edu.tr Electrical and Electronics Engineering Department, Suleyman Demirel University, Isparta, 32260, Turkey

and a fault diagnosis-based approach [13] were used. The nonlinear model predictive based FTC scheme was proposed for an omnidirectional mobile robot with four mecanum wheels. The fault scenario was defined as the wheels not receiving commands and rotating freely [14]. Fault diagnosis was provided with a state transition scheme.

For trajectory tracking, neuro-fuzzy type artificial intelligence methods were employed. Mohareri et al. detailed the process of calculating the gain values using the direct Neural Network (NN) model while developing the kinematic controller [15]. In this work, a kinematic model-based backstepping controller was designed, as well as a non-linear kinematics-based controller and another controller based on the NN inverse model. On the other hand, a fuzzy rule-based fault-tolerant control was proposed for an omnidirectional mobile robot with four wheels [16]. The controller was responsible for not only the generation of an obstacle-free path but also providing an adaptive solution for a combination of one or more of the wheel's faults. The kinematic and dynamic controllers were used in conjunction with Internet of Things (IoT) and NN-based algorithms, and they demonstrated significant advantages in parameter uncertainty and roller skid [17]. Furthermore, learning-based algorithms such as iterative learning and distributed controllers were also considered as FTC to provide satisfactory transient and steady state performance [18,19].

The use of the adaptive law on the sliding mode controller is a correct method for time-varying trajectories and faults. However, if the gain values of the kinematic controller, which is another controller, are not adjusted properly, the results may not be satisfactory. Therefore, in this work, an artificial NN-based backstepping controller has been added to the system. The main objective of this study is to present an overview of recent developments in theory and methodology and develop a control strategy with improved transient and steady-state characteristics by reviewing the literature for FTC systems in DDMR. After evaluating the approaches in the literature on the simulation results, it is aimed to achieve the desired system behaviour faster by developing the controller structure within the scope of this study. The first controller system is the kinematic controller, which allows the use of reference velocity signals given by low-level feedback control to the DDMR wheels. Velocities are obtained by the kinematic controller by using the position errors and reference velocities of the DDMR. The Lyapunov stability method was used to prove the stability of the kinematic controller rule.

The second controller system is an artificial NN-based controller that optimizes the gain coefficients of the kinematic controller. Since the trajectory changes with time, constant gain values are insufficient. Besides, TT error by modelling a proper tracking system is

not a convergent function. The proposed NN-based control algorithm provides variable gains according to the instantaneous changing trajectory of the kinematic controller-based backstepping controller. Since the gains have an effect on the torque values in the wheels, the initial value of the gains should be suitable.

The third controller for this system is the ASMC which is an adaptive system that provides a predictive control scheme for DDMR with a nonlinear dynamic model, noise, uncertainties, and error. ASMC is a special case of variable structure control systems that have a complex structure that uses decision-making rules and feedback control laws together. The decision rule model chooses a specific feedback control structure based on conditions affecting the system's behaviour. This system is called the switching function. The ASMC structure is designed to be driven into the sliding surface in the system state space. Once the sliding surface is reached, ASMC tries to keep the states on the sliding surface boundary [20]. The Lyapunov stability approach was used to test the stability of the ASMC rule.

The last proposed controller system is the traditional PID and SMC, which has been added to the system to discuss the differences between ASMC, SMC, and PID. By using a PID controller or SMC instead of an ASMC in the same mobile robot system without modifying it, the stability of the PID controller in FTC is emphasized and compared [21].

The rest of this paper is organized as follows: The kinematic model and the nonlinear DDMR model are presented with kinematic constraints and the Lagrange approach in chapter 2. Chapter 3 gives kinematic, NN-based, adaptive sliding mode, and PID controllers. The asymptotic stability of this system is demonstrated by Lyapunov theory, which is discussed in Chapter 4. In chapter 5, numerical simulation results are given. Concluding remarks are given in chapter 6.

Mathematical model

This section presents the mathematical model of the DDMR structure. The designed DDMR structure has three wheels. The two wheels on the front of the robot body are the driven wheels. There are two actuators attached to these wheels. The wheel behind the robot body is non-driven and only exists for mobile robot balance.

The robot system described above, and the coordinate systems are shown in Figure 1. The midpoint "A" on the axis between the wheels is the origin of the robot frame. Point "C", which is the robot's centre of mass, is on the axis and is at a distance "d" from point "A". L is the distance between the wheel and the centre point. The wheel radius is shown as "R". Although the velocity of each wheel can change, the DDMR must turn around a point along the common wheel axes for the DDMR to fulfil its rolling movement [22]. The point

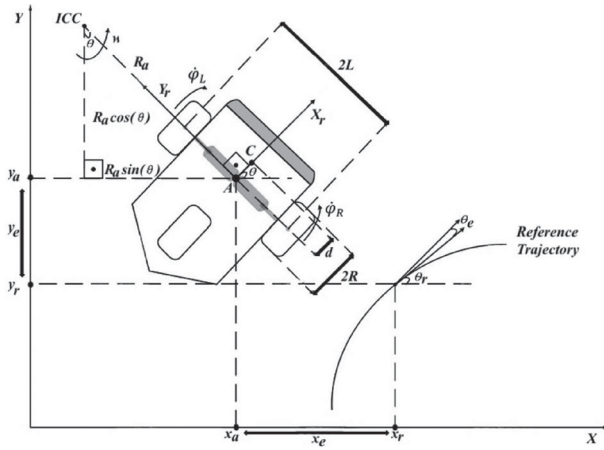


Figure 1. DDMR structure and coordinate systems.

around which the robot rotates, which is called the Instantaneous Center of Curvature (ICC), is shown in Figure 1.

There is a coordinate system on point “A” of the robot body, as can be seen in Figure 1. This frame can be called a robot coordinate system or moving frame since this coordinate system is displaced according to the mobile robot’s movement. The other coordinate system is the inertial coordinate system, which is used when working on the position of the robot. These frames are represented as three-dimensional vectors as follows:

$$q^I = \begin{pmatrix} x^I \\ y^I \\ \theta^I \end{pmatrix}, q^r = \begin{pmatrix} x^r \\ y^r \\ \theta^r \end{pmatrix} \quad (1)$$

A transformation matrix obtained as follows to work on two frames by transforming robot coordinate frame (q^r) with respect to the inertial coordinate system (q^I) [23]:

$$\text{Rot}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Frame-to-frame conversion is done with the following equation by using the orthonormal rotation matrix ($\text{Rot}(\theta)$):

$$\dot{q}^I = \text{Rot}(\theta)^{-1} \dot{q}^r \quad (3)$$

Kinematic model

The forward kinematics of the robot is related to the relationship between the positions, velocities, accelerations of the robot and its physical structures. The equations given in Refs. [1,5] are used to develop the kinematic model structure. By changing the velocities of the two wheels, the trajectory of the robot can change. Assuming that both wheels must have the same angular velocity (ω) around the ICC, the following equations

can be derived using Figure 1 [24]:

$$\begin{aligned} V_R &= w(R_a + L) = R\dot{\phi}_R \\ V_L &= w(R_a - L) = R\dot{\phi}_L \end{aligned} \quad (4)$$

Let the position vector at point “A” be (x_a, y_a) . The position vector for the ICC is obtained as shown in the triangle in Figure 1:

$$\text{ICC} = (x_a - R_a \sin(\theta), y_a + R_a \cos(\theta)) \quad (5)$$

V_R and V_L are the right and left wheel linear velocities. R_a is the distance from the ICC point “A” between the wheels. At any time, v and w can be solved as follows:

$$\begin{aligned} v &= \frac{(V_R + V_L)}{2} = \frac{R(\dot{\phi}_R + \dot{\phi}_L)}{2} \\ w &= \frac{(V_R - V_L)}{2L} = \frac{R(\dot{\phi}_R - \dot{\phi}_L)}{2L} \end{aligned} \quad (6)$$

Due to the wheel structure, the robot body cannot move along the y^r axis in the moving frame. Thus, the velocity at point “A” is calculated as follows:

$$\begin{pmatrix} \dot{x}_A^r \\ \dot{y}_A^r \\ \dot{\theta}_A^r \end{pmatrix} = \begin{pmatrix} R/2 & R/2 \\ 0 & 0 \\ R/2L & -R/2L \end{pmatrix} \begin{pmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{pmatrix} \quad (7)$$

In Equation (7), the rotational velocities of the wheels can be shown as in Equation (8):

$$\eta = \begin{pmatrix} w_R \\ w_L \end{pmatrix} = \begin{pmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{pmatrix} \quad (8)$$

The velocity vector in the robot frame is transformed to the inertial frame using the orthogonal rotation matrix $\text{Rot}(\theta)$ as follows [25]:

$$\begin{aligned} \begin{pmatrix} \dot{x}_A^I \\ \dot{y}_A^I \\ \dot{\theta}_A^I \end{pmatrix} &= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_A^r \\ \dot{y}_A^r \\ \dot{\theta}_A^r \end{pmatrix} \\ &= \begin{pmatrix} \frac{R \cos\theta}{2} & \frac{R \cos\theta}{2} \\ \frac{R \sin\theta}{2} & \frac{R \sin\theta}{2} \\ \frac{R}{2L} & -\frac{R}{2L} \end{pmatrix} \begin{pmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{pmatrix} \end{aligned} \quad (9)$$

If the velocity terms for the DDMR body given in Equation (9) are replaced with the terms given in Equation (6), the forward kinematic model equation becomes as follows [26]:

$$\begin{pmatrix} \dot{x}_A^I \\ \dot{y}_A^I \\ \dot{\theta}_A^I \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (10)$$

The kinematic model is then derived by fault modelling. In this work, Loss of Effectiveness (LOE) is selected for fault modelling. The fault model is designed to account for the power loss in the motors. This is because there is no sensor model in the designed mobile robot. The LOE parameters are defined as $0 < k_L, k_R <$

1 are multiplied by $\dot{\varphi}_R$ and $\dot{\varphi}_L$, respectively, as follows [1]:

$$\begin{pmatrix} w_{Rf} \\ w_{Lf} \end{pmatrix} = \begin{pmatrix} k_R \dot{\varphi}_R \\ k_L \dot{\varphi}_L \end{pmatrix} = \begin{pmatrix} \dot{\varphi}_R + \Delta w_R \\ \dot{\varphi}_L + \Delta w_L \end{pmatrix} \quad (11)$$

$$\text{where } \begin{cases} \Delta w_R = (k_R - 1)\dot{\varphi}_R \\ \Delta w_L = (k_L - 1)\dot{\varphi}_L \end{cases}.$$

The fault values of right and left wheel angular velocities are obtained by w_{Rf} and w_{Lf} , respectively. The combination of the fault values with the kinematic model is given as follows:

$$\begin{pmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\theta}_A \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} + \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} \frac{R}{2}(\Delta w_R + \Delta w_L) \\ \frac{R}{2L}(\Delta w_R - \Delta w_L) \end{pmatrix} \quad (12)$$

The purpose of using the kinematic model of DDMR in the system is to express the current position of the robot body relative to the inertial coordinate system by using the velocities obtained from the nonlinear model structure.

Kinematic constraints

Kinematic constraints of the robot are caused by some assumptions, which are the wheel movement of the robot in the horizontal plane, the point of contact between the wheels and the ground, and the absence of friction for rotation around the wheel, etc. [15].

The velocity of the centre point "A" (in Figure 1) is zero along the lateral axis since the DDMR cannot move laterally [27,28]:

$$\dot{y}_a^f = 0 \quad (13)$$

Using the orthogonal rotation matrix $\text{Rot}(\theta)$, the velocity in the inertial frame is:

$$\dot{y}_a \cos(\theta) - \dot{x}_a \sin(\theta) = 0 \quad (14)$$

Ensure that the point of contact with the ground is on the left and right wheels respectively, labelled P . Thus, the velocities of the contact points in the robot coordinates are as follows:

$$\begin{aligned} v_{pR} &= R\dot{\varphi}_R \\ v_{pL} &= R\dot{\varphi}_L \end{aligned} \quad (15)$$

Velocities in the inertia frame can be calculated as a function of the velocities at point 'A'.

$$\begin{aligned} \dot{x}_{pR} &= \dot{x}_a + L\dot{\theta}\cos(\theta) \\ \dot{y}_{pR} &= \dot{y}_a + L\dot{\theta}\sin(\theta) \\ \dot{x}_{pL} &= \dot{x}_a + L\dot{\theta}\cos(\theta) \\ \dot{y}_{pL} &= \dot{y}_a + L\dot{\theta}\sin(\theta) \end{aligned} \quad (16)$$

The rolling constraint equations are obtained using Equation (16) as follows:

$$\begin{aligned} \dot{x}_{pR} \cos(\theta) + \dot{y}_{pR} \sin(\theta) - R\dot{\varphi}_R &= 0 \\ \dot{x}_{pL} \cos(\theta) + \dot{y}_{pL} \sin(\theta) - R\dot{\varphi}_L &= 0 \end{aligned} \quad (17)$$

Constraint terms Equations (14) and (17) can be written in a matrix form as follows:

$$A(q)\dot{q} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 \\ \cos(\theta) & \sin(\theta) & L & -R & 0 \\ \cos(\theta) & \sin(\theta) & -L & 0 & -R \end{pmatrix} \times \begin{pmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta}_a \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix} = 0 \quad (18)$$

Dynamic model

The dynamic analysis of the DDMR structure can be defined as the examination of the relationships between the torque or force magnitudes applied to the wheels by the actuator and the position, velocity, and acceleration of the DDMR with respect to time. The main difference between dynamic and kinematic modelling is that the kinematic model examines motion only with the geometric relations governing the system, without considering the forces affecting the motion [16,29].

The dynamic model of the system is derived from the Lagrangian dynamic approach [16,28]. The general form of the Lagrangian equation is obtained as follows:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = \sum_{j=1}^n \lambda_j a_{ji} + Q_i \quad (19)$$

Above, Q_i refers to nonconservative forces in the system, and the Lagrange multiplier is denoted by λ_j .

$$L = T - V \quad (20)$$

Equation (20) is the Lagrange function, which is the difference between the kinetic (T) and potential (V) energies of the system. In our case, the potential energy is zero because there is no height state in the system, so Lagrange is equal to the kinetic energy for this system (Figure 2).

The kinetic energy formula of the mobile robot body without wheels is as follows:

$$T_c = \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \quad (21)$$

In here, m_c refers to the mass of the DDMR without the driving wheels and actuators, I_c refers to the moment of inertia of the DDMR about the x_r axis through the centre of mass.

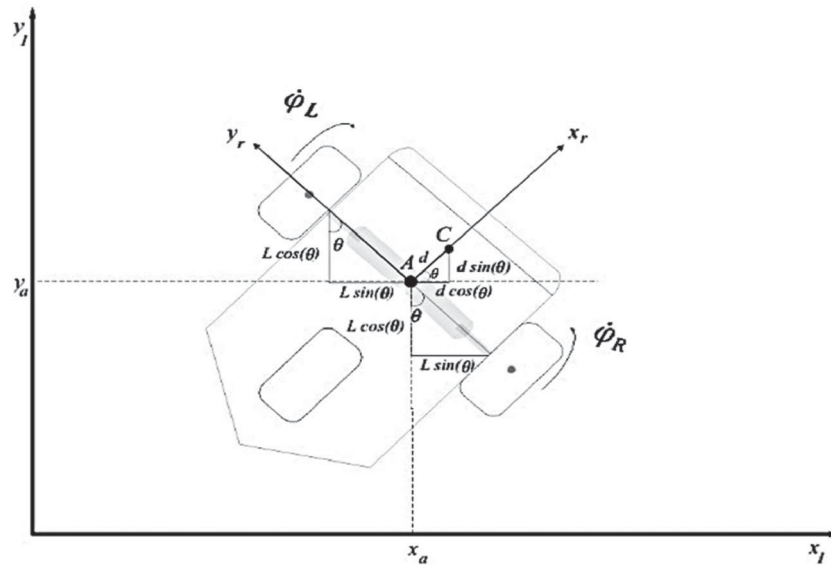


Figure 2. Coordinates of points on DDMR.

The kinetic energy of the right and left wheels and actuators of the robot is as follows:

$$\begin{aligned} T_{wR} &= \frac{1}{2}m_w v_{wR}^2 + \frac{1}{2}I_m \dot{\theta}^2 + \frac{1}{2}I_w \dot{\phi}_R^2 \\ T_{wL} &= \frac{1}{2}m_w v_{wL}^2 + \frac{1}{2}I_m \dot{\theta}^2 + \frac{1}{2}I_w \dot{\phi}_L^2 \end{aligned} \quad (22)$$

Here, m_w refers to the mass of each driving wheel, including the actuator. v_{wR} and v_{wL} are right and left linear velocities, respectively. I_w refers to the moment of inertia of each driving wheel with an actuator about the wheel axis, and I_m refers to the moment of inertia of each driving wheel with an actuator about the wheel radius [27].

The kinetic energy of the robot body in Equation (21) is obtained as follows:

$$\begin{aligned} T_c &= \frac{1}{2}m_c(\dot{x}_a^2 + \dot{y}_a^2) - m_c d \dot{\theta}(\dot{x}_a \sin(\theta) - \dot{y}_a \cos(\theta)) \\ &\quad + \frac{1}{2}m_c d^2 \dot{\theta}^2 + \frac{1}{2}I_c \dot{\theta}^2 \end{aligned} \quad (23)$$

Here, x_a , y_a , and θ_a refer to the position vector of the midpoint of the mobile robot frame.

The kinetic energy of the right and left wheels in Equation (22) of the robot:

$$\begin{aligned} T_{wR} &= \frac{1}{2}m_w(\dot{x}_a^2 + \dot{y}_a^2) + m_w L \dot{\theta}(\dot{x}_a \cos(\theta) + \dot{y}_a \sin(\theta)) \\ &\quad + \frac{1}{2}m_w L^2 \dot{\theta}^2 + \frac{1}{2}I_m \dot{\theta}^2 + \frac{1}{2}I_w \dot{\phi}_R^2 \\ T_{wL} &= \frac{1}{2}m_w(\dot{x}_a^2 + \dot{y}_a^2) - m_w L \dot{\theta}(\dot{x}_a \cos(\theta) + \dot{y}_a \sin(\theta)) \\ &\quad + \frac{1}{2}m_w L^2 \dot{\theta}^2 + \frac{1}{2}I_m \dot{\theta}^2 + \frac{1}{2}I_w \dot{\phi}_L^2 \end{aligned} \quad (24)$$

The total kinetic energy of the mobile robot is obtained using Equations (23) and (24), as follows:

$$T = T_c + T_{wR} + T_{wL} = \left(\frac{1}{2}m_c + m_w\right)(\dot{x}_a^2 + \dot{y}_a^2)$$

$$\begin{aligned} &- m_c d \dot{\theta}(\dot{x}_a \sin(\theta) - \dot{y}_a \cos(\theta)) \\ &+ \left(m_w L^2 + I_m + \frac{1}{2}m_c d^2 + \frac{1}{2}I_c\right) \dot{\theta}^2 \\ &+ \frac{1}{2}I_w(\dot{\phi}_R^2 + \dot{\phi}_L^2) \end{aligned} \quad (25)$$

The total mass and equivalent inertia of the mobile robot:

$$m = m_c + 2m_w \quad (26)$$

$$I = I_c + m_c d^2 + 2m_w L^2 + 2I_m \quad (27)$$

The Lagrange equation is given as follows using Equations (26) and (27):

$$\begin{aligned} L &= \frac{1}{2}m(\dot{x}_a^2 + \dot{y}_a^2) - m_c d \dot{\theta}(\dot{x}_a \sin(\theta) \\ &- \dot{y}_a \cos(\theta)) + \frac{1}{2}I \dot{\theta}^2 \\ &+ \frac{1}{2}I_w(\dot{\phi}_R^2 + \dot{\phi}_L^2) \end{aligned} \quad (28)$$

A step-by-step approach is used to find the dynamic model equations using generalized coordinates and the Lagrangian equation. As a result, the Lagrange equations are obtained as follows:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_a} \right) - \frac{\partial L}{\partial x_a} &= m \ddot{x}_a - m_c d \ddot{\theta} \sin(\theta) \\ &- m_c d \dot{\theta}^2 \cos(\theta) = C_1 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}_a} \right) - \frac{\partial L}{\partial y_a} &= m \ddot{y}_a + m_c d \ddot{\theta} \cos(\theta) \\ &- m_c d \dot{\theta}^2 \sin(\theta) = C_2 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} &= I \ddot{\theta} - m_c d \ddot{x}_a \sin(\theta) \\ &+ m_c d \dot{y}_a \cos(\theta) = C_3 \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_R} \right) - \frac{\partial L}{\partial \varphi_R} &= I_w \ddot{\varphi}_R = \tau_R + C_4 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}_L} \right) - \frac{\partial L}{\partial \varphi_L} &= I_w \ddot{\varphi}_L = \tau_L + C_5 \end{aligned} \quad (29)$$

Here, C_1, C_2, C_3, C_4 and C_5 refer to the coefficients related to kinematic constraints that can be written in terms λ Lagrange multiplier vectors.

Using the Lagrange equations above, general dynamic equation is given in Equation (30) where $M(q)$ is the symmetric positive definite inertia matrix, $V(q, \dot{q})$ is the centripetal and Coriolis matrix, $G(q)$ is the gravitational vector, τ_d denotes the bounded unknown disturbances or noises, including unstructured unmodelled dynamics, $B(q)$ is the input transformation matrix, τ is the input torque vector, $A^\tau(q)$ is the matrix associated with the constraints, and λ is the vector of the constraint forces.

$$\begin{aligned} M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) \\ + \tau_d = B(q)\tau - A^\tau(q)\lambda \end{aligned} \quad (30)$$

q , \dot{q} and \ddot{q} specify position, velocity, and acceleration vectors, respectively. Since the DDMR system moves in the horizontal plane and the system is only tested in simulation, the gravitational force $G(q)$ is neglected. Since gravity is neglected and wheel frictions are given to the system with the fault defined in the kinematic model, $F(\dot{q})$ is neglected. In addition, since the disturbance value is added to the ASMC signal, τ_d is neglected in the dynamic model. After these assumptions, the dynamic model is obtained as follows:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} = B(q)\tau - A^\tau(q)\lambda \quad (31)$$

By expanding the system in Equation (9), \dot{q} vector can be written as follows. The velocities (\dot{q}) and accelerations (\ddot{q}) in Equation (31) can be found using the in Equation (32).

$$\dot{q} = \begin{pmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix} = \begin{pmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta \\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta \\ \frac{R}{2L} & -\frac{R}{2L} \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix} \quad (32)$$

Here, to express Equation (31), the energy-based Lagrangian approach, which is the dynamic model of the robot, is used.

Using Equation (29), the structure in Equation (31) is obtained as follows:

$$\begin{aligned} M(q) &= \begin{pmatrix} m & 0 & -m_c d \sin\theta & 0 & 0 \\ 0 & m & m_c d \cos\theta & 0 & 0 \\ -m_c d \sin\theta & m_c d \cos\theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{pmatrix} \\ V(q, \dot{q}) &= \begin{pmatrix} 0 & 0 & -m_c d \dot{\theta} \cos\theta & 0 & 0 \\ 0 & 0 & -m_c d \dot{\theta} \sin\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\ B(q) &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \\ A^\tau(q)\lambda &= \begin{pmatrix} -\sin\theta & \cos\theta & \cos\theta \\ \cos\theta & \sin\theta & \sin\theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & R \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{pmatrix} \end{aligned} \quad (33)$$

Since the Lagrange multipliers λ are not known, the constraint term $A^\tau(q)\lambda$ in Equation (33) must be eliminated by defining the reduced vector.

The $S(q)$ matrix is used for the reduction. The $S(q)$ matrix is a modified forward kinematic matrix relating the distance between the robot's centre of gravity and the wheel axis, as seen in the forward kinematic equation [18].

$$\dot{q} = S(q)\eta \quad (34)$$

The acceleration matrix is obtained by taking the time derivative of Equation (34) as follows:

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \quad (35)$$

The $S(q)$ matrix is in the null space of the kinematic constraint matrix $A(q)$, which means $S^T(q)A^T(q) = 0$. Hence:

$$\begin{pmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\sin\theta & \frac{R}{2L} & 1 & 0 \\ \frac{R}{2}\cos\theta & \frac{R}{2}\sin\theta & -\frac{R}{2L} & 0 & 1 \end{pmatrix} \times \begin{pmatrix} -\sin\theta & \cos\theta & \cos\theta \\ \cos\theta & \sin\theta & \sin\theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & R \end{pmatrix} = 0 \quad (36)$$

As can be seen in Equation (36), the constraint term of the dynamic model can be eliminated.

The dynamic model should be revised accordingly. τ_L and τ_R are left and right torques respectively, used in the new dynamic equation. It can be expressed as follows:

$$\tau = u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \tau_R + \tau_L \\ \tau_R - \tau_L \end{pmatrix} \quad (37)$$

Using Equations (34) and (37), the dynamic model of DDMR is modified as follows:

$$\begin{aligned} M(q)[\dot{S}(q)\eta + S(q)\dot{\eta}] + V(q, \dot{q})[S(q)\eta] \\ = B(q)\tau - A^\tau(q)\lambda \end{aligned} \quad (38)$$

If the equation is reordered, the following equation is obtained:

$$(M(q)S(q))\dot{\eta} + (M(q)\dot{S}(q) + V(q, \dot{q})S(q))\eta = B(q)\tau - A^T(q)\lambda \quad (39)$$

Both sides of the Equation (39) are multiplied by transformation matrix $S^T(q)$:

$$(S^T(q)M(q)S(q))\dot{\eta} + (S^T(q)M(q)\dot{S}(q) + S^T(q)V(q, \dot{q})S(q))\eta = S^T(q)B(q)\tau - S^T(q)A^T(q)\lambda \quad (40)$$

The dynamic model terms in Equation (40) can be represented as follows:

$$\begin{aligned} \bar{M}(q) &= S^T(q)M(q)S(q) \\ \bar{V}(q, \dot{q}) &= S^T(q)M(q)\dot{S}(q) + S^T(q)V(q, \dot{q})S(q) \\ \bar{B}(q) &= S^T(q)B(q)\tau \end{aligned} \quad (41)$$

By reordering the Equations (40) and (41), the new dynamic model of DDMR becomes:

$$\bar{M}(q)\dot{\eta} + \bar{V}(q, \dot{q})\eta = \bar{B}(q)\tau \quad (42)$$

If the matrix multiplications shown in Equation (41) are done, the following matrix terms are obtained:

$$\begin{aligned} \bar{M}(q) &= \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix} \\ \bar{V}(q, \dot{q}) &= \begin{bmatrix} 0 & \frac{R^2}{2L}m_c d\dot{\theta} \\ -\frac{R^2}{2L}m_c d\dot{\theta} & 0 \end{bmatrix} \\ \bar{B}(q) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (43)$$

Using the velocity in Equation (6), Equation (42) can be converted to an alternative form. This structure in Equation (44) is the nonlinear model of DDMR.

$$\begin{aligned} \begin{bmatrix} m_0 & 0 \\ 0 & I_0 \end{bmatrix} \begin{pmatrix} \dot{v} \\ \dot{w} \end{pmatrix} + \begin{bmatrix} 0 & -m_c d w \\ m_c d w & 0 \end{bmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \\ = \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & \frac{1}{R} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \end{aligned} \quad (44)$$

where $\begin{cases} m_0 = \left(m + \frac{2I_w}{R^2}\right) \\ I_0 = \left(I + \frac{2L^2}{R^2}I_w\right) \end{cases}$.

Note that the Lagrange approximation considers the mass and inertia of the wheels and not the robot as a single rigid body. With the nonlinear model, linear velocity and angular velocities are obtained by using Equation (44).

Controller approaches

Kinematic controller

The kinematic-based backstepping controller was proposed in the literature for a non-holonomic DDMR

[4,15]. In that approach, a stable TT control rule for a non-holonomic mobile robot that neglects DDMR dynamics is based on the steering system. A mobile robot system has two postures, which are the reference position vector P_r and the current position vector P_c . They are three-dimensional vectors and include the x , y , and θ postures. The error is obtained by taking the difference between the reference and current position vectors.

The difference between P_r and P_c gives the error value in the base frame. This value is multiplied by the rotation matrix to obtain the error posture in the robot coordinate system.

$$e_p = \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{pmatrix} \quad (45)$$

The current velocities are obtained as follows:

$$\begin{pmatrix} v_c \\ w_c \end{pmatrix} = \begin{pmatrix} v_r \cos\theta_e + K_x x_e \\ w_r + v_r (K_y y_e + K_\theta \sin\theta_e) \end{pmatrix} \quad (46)$$

K_x , K_y and K_θ are positive gain values. These gains are very important for the system to work satisfactory. Since DDMR system needs slow and a non-oscillating response, the kinematic controller gains must be well tuned [15]. For this reason, Neural network-based algorithm is employed for determining the gains.

Neural network based adaptive backstepping controller

The proposed control algorithm provides the backstepping controller with parameters and adaptive gains that vary with reference trajectory [30,31]. The control structure adapts the kinematics-based controller gains to minimize the following cost function [15,32].

$$\mathcal{J} = \frac{1}{2} \sum g_x x_e^2 + g_y y_e^2 + g_\theta \theta_e^2 \quad (47)$$

In here, g_x , g_y and g_θ are neural network gains. Error values can be seen from Equation (45). K_x , K_y and K_θ are considered part of the cost function above and they are optimized and updated according to the gradient descent method. The kinematic controller gains are represented by the vector α . The partial derivative of the cost function with respect to α :

$$\frac{\partial \mathcal{J}}{\partial \alpha} = g_x x_e \frac{\partial x_e}{\partial \alpha} + g_y y_e \frac{\partial y_e}{\partial \alpha} + g_\theta \theta_e \frac{\partial \theta_e}{\partial \alpha} = e_p^T g \frac{\partial e_p}{\partial \alpha} \quad (48)$$

where $\alpha = [K_x \ K_y \ K_\theta]$.

The matrix form of the g value here is shown as follows:

$$g = \begin{bmatrix} g_x & 0 & 0 \\ 0 & g_y & 0 \\ 0 & 0 & g_\theta \end{bmatrix} \quad (49)$$

In Equation (48), if e_p is replaced by Equation (45):

$$\frac{\partial \mathcal{J}}{\partial \alpha} = g e_p^T \frac{\partial (T_e(q_r - q_c))}{\partial \alpha} = -e_p^T \times g \times T_e \times \frac{\partial q_c}{\partial \alpha} \quad (50)$$

Since the vector q_r is the reference position consisting of fixed values, its derivative is zero. As the chain rule indicates, the following equation can be written:

$$\frac{\partial q_c}{\partial \alpha} = \frac{\partial q_c}{\partial v_c} \times \frac{\partial v_c}{\partial \alpha} \quad (51)$$

After the Equation (51) given above is obtained, the desired derivative expression in Equation (52) is written as follows:

$$\frac{\partial q_c}{\partial \alpha} = \begin{bmatrix} \mathcal{J}ac_{v11} & \mathcal{J}ac_{v12} \\ \mathcal{J}ac_{v21} & \mathcal{J}ac_{v22} \\ \mathcal{J}ac_{v31} & \mathcal{J}ac_{v32} \end{bmatrix} \times \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin \theta \end{bmatrix} \quad (52)$$

The derivative of the cost function with respect to the controller gains $\frac{\partial \mathcal{J}}{\partial \alpha}$ is rewritten as follows:

$$\frac{\partial \mathcal{J}}{\partial \alpha} = \begin{bmatrix} \frac{\partial \mathcal{J}}{\partial K_x} & \frac{\partial \mathcal{J}}{\partial K_y} & \frac{\partial \mathcal{J}}{\partial K_\theta} \end{bmatrix} = -e_p^T \times g \times T_e \times \mathcal{J}ac_v \times \frac{\partial v_c}{\partial \alpha} \quad (53)$$

The representation of the derivative of the cost function with respect to the controller gains in matrix form is written as follows:

$$\frac{\partial \mathcal{J}}{\partial \alpha} = \begin{bmatrix} \frac{\partial \mathcal{J}}{\partial K_x} & \frac{\partial \mathcal{J}}{\partial K_y} & \frac{\partial \mathcal{J}}{\partial K_\theta} \end{bmatrix} = - \begin{bmatrix} x_e & y_e & \theta_e \end{bmatrix} \times \begin{bmatrix} g_x & 0 & 0 \\ 0 & g_y & 0 \\ 0 & 0 & g_\theta \end{bmatrix} \times \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \mathcal{J}ac_{v11} & \mathcal{J}ac_{v12} \\ \mathcal{J}ac_{v21} & \mathcal{J}ac_{v22} \\ \mathcal{J}ac_{v31} & \mathcal{J}ac_{v32} \end{bmatrix} \times \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin \theta \end{bmatrix} \quad (54)$$

The algorithm for calculating the Jacobian matrix using the NN model is as follows: The initial weight values should be the most suitable for the model. Weights are tuned by training the model using the back propagation algorithm. It is essential to have incorrect values to use back propagation. Thus, the forward propagation method is used first. The forward propagation algorithm is used to obtain the prediction data generated by the NN. Since it is predicted data, there is a difference between the estimated value and the real value. The difference between these two values is used to determine the error value, which propagates from the output layer to the input layer on the NN, resulting

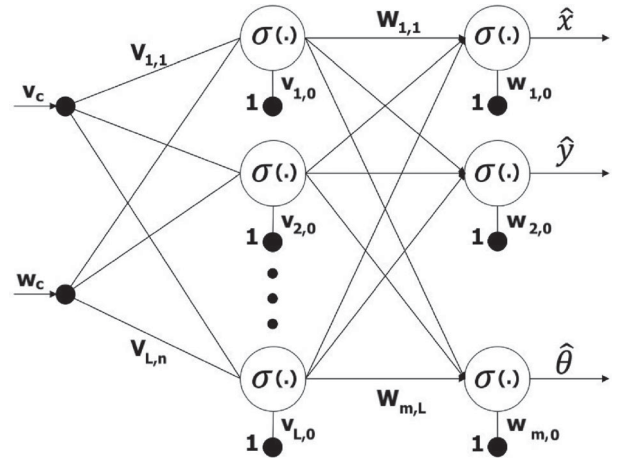


Figure 3. Two-layer NN model with two layers of neurons.

in “backward propagation” with various derivative processes [15]. The NN model with two layers of neurons is shown in Figure 3.

The first layer of the NN model feeds L neurons, and the second layer feeds m neurons. The following terminology is used to describe the different parameters of the neural networks shown in Figure 3: Kinematic controller velocities (v_c and w_c) are input layers for NN. \hat{x} , \hat{y} , $\hat{\theta}$ are the outputs of the NN. N_i and N_o are the number of inputs and outputs of the neural network, respectively. N_h is the number of neurons in the hidden layer. ζ and δ are the learning and momentum rate, respectively. σ is the activation function. Whereas Whi values are the weights from hidden layer to input layer, Woh values are the weights from the output layer to hidden layer.

According to the defined neural network model above, the relationship between the inputs and the outputs of the network is given as follows:

$$y_i = \sigma \left(\sum_{l=1}^L Woh_{il} \sigma \left(\sum_{j=1}^n Whi_{lj} x_j + v_{l0} \right) + w_{i0} \right) \quad (55)$$

where $i = 1, 2, \dots, m$ and $l = 1, 2, \dots, L$.

The activation function represented by σ is chosen as the sigmoid (logistic curve) function is given in Equation (56).

$$\sigma(\rho) = \frac{1}{1 + e^{-\rho}} \quad (56)$$

The back propagation algorithm needs the derivative of the activation function. Derivative of the sigmoid function with respect to ρ :

$$\frac{d}{d\rho} \sigma(\rho) = \frac{1}{(1 + e^{-\rho})^2} = \sigma(\rho)(1 - \sigma(\rho)) \quad (57)$$

The expression ρ in the Equation (56) above is the representation of the product of the weight and neuron values up to the desired layer.

The back propagation algorithm is a weight adjustment algorithm based on the gradient descent method.

$$\begin{aligned} W_{il}(k+1) &= W_{il}(k) - \zeta \frac{\partial E(k)}{\partial W_{il}} \\ V_{lj}(k+1) &= V_{lj}(k) - \zeta \frac{\partial E(k)}{\partial V_{lj}} \end{aligned} \quad (58)$$

An output signal is given as a reference to the neural network, and the difference between this output and the output (y_i) produced by NN is an error. The weights are changed in cycles until the error is minimized by the network.

$$e_i(k) = Y_i(k) - y_i(k) \quad (59)$$

$$\text{where } Y_i(k) = \begin{pmatrix} x_A \\ y_A \\ \theta_A \end{pmatrix}.$$

By using the error function, the cost function is found as follows:

$$E(k) = \frac{1}{2} e_i(k)^2 \quad (60)$$

The required gradients of the cost function $E(k)$ by weights can be easily determined using the chain rule.

$$\begin{aligned} \frac{\partial E}{\partial W_{il}} &= \left[e_i \times (-1) \times \sigma'(u_i^2) \right] \times z_l = -z_l \sigma'(u_i^2) e_i \\ \frac{\partial E}{\partial V_{lj}} &= \left[\sum_{i=1}^m \left(-\sigma'(u_i^2) e_i \right) \times W_{il} \times \sigma'(u_l^1) \right] \times x_j \\ &= -x_j \sigma'(u_l^1) \sum_{i=1}^m \left(\sigma'(u_i^2) e_i \right) \times W_{il} \end{aligned} \quad (61)$$

In here, z_l refer to the sigmoid function up to the hidden layer. The Jacobian matrix is obtained as follows:

$$\begin{aligned} \mathcal{J}ac_v &= \begin{bmatrix} \frac{\partial x}{\partial v_c} & \frac{\partial x}{\partial w_c} \\ \frac{\partial y}{\partial v_c} & \frac{\partial y}{\partial w_c} \\ \frac{\partial \theta}{\partial v_c} & \frac{\partial \theta}{\partial w_c} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\hat{x}(t) - \hat{x}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{x}(t) - \hat{x}(t-1)}{w_c(t) - w_c(t-1)} \\ \frac{\hat{y}(t) - \hat{y}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{y}(t) - \hat{y}(t-1)}{w_c(t) - w_c(t-1)} \\ \frac{\hat{\theta}(t) - \hat{\theta}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{\theta}(t) - \hat{\theta}(t-1)}{w_c(t) - w_c(t-1)} \end{bmatrix} \end{aligned} \quad (62)$$

With the above equations, the kinematic controller gains change and adapts to make the cost function zero according to the gradient descent method as follows:

$$\begin{aligned} K_{x(t)} &= K_{x(t-1)} + \Delta K_x \\ K_{y(t)} &= K_{y(t-1)} + \Delta K_y \\ K_{\theta(t)} &= K_{\theta(t-1)} + \Delta K_{\theta} \end{aligned} \quad (63)$$

Adaptive sliding mode controller

For the ASMC structure, traditional dynamic equations are used [5,6]. The sliding surface linear and angular

velocities, which come from the output of the kinematic controller and nonlinear model, are used [33]. The velocity error is found by taking the differences between these velocities [34].

$$e_c = \begin{pmatrix} e_v \\ e_w \end{pmatrix} = \begin{pmatrix} v_c \\ w_c \end{pmatrix} - \begin{pmatrix} v \\ w \end{pmatrix} \quad (64)$$

In this work, a PI type sliding surface is used for ASMC as follows:

$$s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = e_c(t) + \beta \int e_c(t) dt \quad (65)$$

In here, β must have a positive value ($\beta > 0$) satisfying the Hurwitz condition. If the system is on the sliding surface $s(t) = 0$ and the error $e_c(t = \infty) \rightarrow 0$, $\dot{s}(t) = 0$ is the necessary condition for the state to remain on the sliding surface during the trajectory tracking. Taking the derivative of Equation (65) for this case, it becomes as follows:

$$\begin{aligned} \dot{s}_1(t) &= \dot{v}_c(t) - \dot{v}(t) + \beta e_v(t) = 0 \\ \dot{s}_2(t) &= \dot{w}_c(t) - \dot{w}(t) + \beta e_w(t) = 0 \end{aligned} \quad (66)$$

The acceleration terms in Equation (66) can be expressed as:

$$\begin{aligned} \dot{v}(t) &= \dot{v}_c(t) + \beta e_v(t) \\ \dot{w}(t) &= \dot{w}_c(t) + \beta e_w(t) \end{aligned} \quad (67)$$

The linearization of the nonlinear dynamic model in Equation (44) is obtained as follows:

$$\begin{aligned} \dot{v}(t) &= \frac{1}{m_0 R} u_1 \\ \dot{w}(t) &= \frac{L}{I_0 R} u_2 \end{aligned} \quad (68)$$

If the acceleration expressions in Equation (67) are written into Equation (68), it becomes as follows:

$$u_{eq}(t) = \begin{cases} u_{eq1}(t) = \hat{\gamma}[\dot{v}_c(t) + \beta e_v(t)] \\ u_{eq2}(t) = \hat{\alpha}[\dot{w}_c(t) + \beta e_w(t)] \end{cases} \quad (69)$$

Here, γ and α are used as adaptive model parameters. The following torques force the system to approach the switching signal faster with gain K .

$$u_w(t) = \begin{cases} u_{w1}(t) = K_1 s_1 \\ u_{w2}(t) = K_2 s_2 \end{cases} \quad (70)$$

where $K_1, K_2 > 0$.

The discontinuity of the sign function in the ASMC law and the large η value can cause chattering phenomena. To avoid chattering, the sign function has been replaced by the continuous \tanh (Hyperbolic tangent) function. In other words, the \tanh function is used as an estimator of the sign function. The steepness of the \tanh function determines how it can approach the sign function [34].

$$u_d(t) = \begin{cases} u_{d1}(t) = \eta_1 \tanh(s_1/\varepsilon) \\ u_{d2}(t) = \eta_2 \tanh(s_2/\varepsilon) \end{cases} \quad (71)$$

where $\varepsilon > 0$.

The control signal is obtained by summing the above Equations (69), (70), and (71), as follows:

$$u(t) = u_{eq}(t) + u_w(t) + u_d(t) \quad (72)$$

u_{eq} is the adaptive compensation item, u_w is the feedback item, and u_d is the robustness item. Disturbances are added externally to the control signals. In this work, the disturbances are limited and represented with Δ_1 and Δ_2 : $|\Delta| \leq D < |\eta|$.

The adaptive law is obtained for this system as proposed in Refs. [34–37]. Using a reasonable priori knowledge of physical systems, an adaptive robust control design based on a discontinuous projection map is obtained to solve the TT control problem. The estimated parameters γ and α are revised with the adaptive law parameters are obtained as follows:

$$\begin{aligned} \dot{\hat{\gamma}} &= \text{Proj}_{\hat{\gamma}}[-as_1(\dot{v}_c + \beta e_v)] \\ \dot{\hat{\alpha}} &= \text{Proj}_{\hat{\alpha}}[-bs_2(\dot{w}_c + \beta e_w)] \end{aligned} \quad (73)$$

$$\begin{aligned} \text{Proj}_{\hat{\gamma}}(\Delta) &= \begin{cases} 0, & \text{if } \hat{\gamma} = \gamma_{\max} \text{ and } \cdot > 0 \\ 0, & \text{if } \hat{\gamma} = \gamma_{\min} \text{ and } \cdot < 0 \\ \Delta, & \text{otherwise} \end{cases} \\ \text{Proj}_{\hat{\alpha}}(\Delta) &= \begin{cases} 0, & \text{if } \hat{\alpha} = \alpha_{\max} \text{ and } \cdot > 0 \\ 0, & \text{if } \hat{\alpha} = \alpha_{\min} \text{ and } \cdot < 0 \\ \Delta, & \text{otherwise} \end{cases} \end{aligned} \quad (74)$$

Maximum or minimum values of γ and α can be calculated as follows:

$$\begin{aligned} \gamma_{\max, \min} &= m_0 R. \\ \alpha_{\max, \min} &= I_0 R/L \end{aligned} \quad (75)$$

Using the above equation, ASMC law is obtained as:

$$u(t) = \begin{cases} u_1(t) = \hat{\gamma}[\dot{v}_c(t) + \beta e_v(t) \\ \quad + K_1 s_1 + \eta_1 \tanh(s_1/\varepsilon)] \\ u_2(t) = \hat{\alpha}[\dot{w}_c(t) + \beta e_w(t) \\ \quad + K_2 s_2 + \eta_2 \tanh(s_2/\varepsilon)] \end{cases} \quad (76)$$

PID controller

In this section, a conventional PID controller, which is widely used in the literature, is added to the system to compare the two controllers. The proportional term (P) is used to calculate instantaneous errors, the integral term (I) provides information about the amount of previous errors, and the derivative term (D) provides an estimate of future errors based on the instantaneous magnitude fluctuation rate [38].

Two PID controllers are used in the system. The first one produces a control signal with linear velocity errors. The second one generates a control signal with angular velocity errors. The following function is used for these

two PID controllers:

$$u(t) = \begin{cases} u_1(t) = K_p e_v(t) + K_i \int_0^t e_v(t) dt + K_d \frac{d}{dt} e_v(t) \\ u_2(t) = K_p e_w(t) + K_i \int_0^t e_w(t) dt + K_d \frac{d}{dt} e_w(t) \end{cases} \quad (77)$$

Lyapunov stability analysis

Theorem 1: *The stability analysis of the kinematic controller is proved by the Lyapunov stability approach in this section [5,15]. The mobile robot tracking error system dynamics is given in Equation (45). The time derivative of the TT position error is:*

$$\begin{aligned} \dot{e}_p &= \begin{pmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{pmatrix} \\ &= \begin{pmatrix} w(e_p, q^r) y_e - v(e_p, q^r) + v_r \cos(\theta_e) \\ -w(e_p, q^r) x_e + v_r \sin(\theta_e) \\ w_r - w(e_p, q^r) \end{pmatrix} \end{aligned} \quad (78)$$

The position error dynamics can be represented in the matrix form as follows:

$$\begin{aligned} \begin{pmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{pmatrix} &= v \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} + w \begin{pmatrix} y_e \\ -x_e \\ -1 \end{pmatrix} \\ &+ \begin{pmatrix} v_r \cos(\theta_e) \\ v_r \sin(\theta_e) \\ w_r \end{pmatrix} \end{aligned} \quad (79)$$

For any initial conditions, all the functions for the mobile tracking error dynamic system are bounded and go to zero with time; so $\lim_{t \rightarrow \infty} (x_e, y_e, \theta_e) = 0$.

Proof: Except for the Lyapunov theorem, the LOE chosen as the fault model effect on the DDMR model can be seen from the equations as follows:

$$\begin{aligned} \dot{x}_e &= -\frac{\dot{x}}{\Delta v \cos^2 \theta} + \frac{\dot{\theta}}{\Delta w} y_e + v_r \cos \frac{\dot{\theta}}{\Delta w} \\ \dot{y}_e &= -x_e \frac{\dot{\theta}}{\Delta w} + v_r \sin \frac{\dot{\theta}}{\Delta w} \\ \dot{\theta}_e &= -\frac{\dot{\theta}}{\Delta w} + w_r \end{aligned} \quad (80)$$

For the above Equation (78), the scalar function (V_0), which is the positive definite Lyapunov function for the kinematic controller analysis, is suggested as follows:

$$V_0 = \frac{(x_e^2 + y_e^2)}{2} + \frac{(1 - \cos(y_e))}{K_y} \quad (81)$$

$V_0 \geq 0$ and if $e_p = 0$, $V_0 = 0$. Also, if $e_p \neq 0$, $V_0 > 0$. The time derivative of V_0 is:

$$\dot{V}_0 = \dot{x}_e x_e + \dot{y}_e y_e + \frac{\dot{y}_e \sin(y_e)}{K_y} \quad (82)$$

Substituting (45) and (78) in (82), it is obtained as follows:

$$\dot{V}_0 = -K_x x_e^2 - \frac{v_r K_\theta \sin^2(\theta_e)}{K_y} \leq 0 \quad (83)$$

For $t > 0$, \dot{V}_0 is a negative definite function for any $x_e, y_e, \theta_e \neq 0$ that indicates the asymptotic stability of the overall system. Thus, e_p goes to zero under the following conditions: reference velocities are continuous and v_r, w_r, K_x, K_y and K_θ are bounded [15]. ■

Theorem 2: *Considering the values of the disturbances and faults of the mobile robot, an adaptive law is proposed in Equation (76). The stability of the ASMC is proved by the Lyapunov stability approach in this section, which is proposed in [5,34]. The positive definite Lyapunov function is selected as:*

$$\begin{aligned} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} &= \frac{1}{2} \begin{pmatrix} \gamma & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} s_1^2 \\ s_2^2 \end{pmatrix} \\ &+ \frac{1}{2} \begin{pmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{b} \end{pmatrix} \begin{pmatrix} \tilde{\gamma}^2 \\ \tilde{\alpha}^2 \end{pmatrix} \quad (84) \\ \text{where } \begin{cases} \tilde{\gamma} &= \hat{\gamma} - \gamma (a > 0) \\ \tilde{\alpha} &= \hat{\alpha} - \alpha (b > 0) \end{cases} \end{aligned}$$

Proof: The derivative of Lyapunov function in Equation (84) is as follows:

$$\begin{aligned} \begin{pmatrix} \dot{V}_1 \\ \dot{V}_2 \end{pmatrix} &= \begin{pmatrix} \gamma & 0 \\ 0 & \alpha \end{pmatrix} \begin{pmatrix} \dot{s}_1 s_1 \\ \dot{s}_2 s_2 \end{pmatrix} \\ &+ \begin{pmatrix} \frac{1}{a} & 0 \\ 0 & \frac{1}{b} \end{pmatrix} \begin{pmatrix} \dot{\tilde{\gamma}} \tilde{\gamma} \\ \dot{\tilde{\alpha}} \tilde{\alpha} \end{pmatrix} \quad (85) \end{aligned}$$

If external artificial noises are added to the Equation (66), derivative of sliding surface is obtained as follows:

$$\begin{aligned} \dot{s}_1(t) &= \dot{v}_c(t) - \left(\frac{1}{\gamma} u_1(t) + \Delta_1 \right) + \beta e_v(t) = 0 \\ \dot{s}_2(t) &= \dot{w}_c(t) - \left(\frac{1}{\alpha} u_2(t) + \Delta_2 \right) + \beta e_w(t) = 0 \end{aligned} \quad (86)$$

where $\dot{v}(t) = \left(\frac{1}{\gamma} u_1(t) + \Delta_1 \right)$, $\dot{w}(t) = \left(\frac{1}{\alpha} u_2(t) + \Delta_2 \right)$

By substituting Equation (86) into Equation (85), the accelerations become:

$$\begin{aligned} \dot{V}_1 &= \gamma s_1 \left[(\dot{v}_c(t) + \beta e_v(t)) - \left(\frac{1}{\gamma} u_1(t) + \Delta_1 \right) \right] + \frac{1}{a} \tilde{\gamma} \dot{\tilde{\gamma}} \\ \dot{V}_2 &= \alpha s_2 \left[(\dot{w}_c(t) + \beta e_w(t)) - \left(\frac{1}{\alpha} u_2(t) + \Delta_2 \right) \right] + \frac{1}{b} \tilde{\alpha} \dot{\tilde{\alpha}} \end{aligned} \quad (87)$$

Writing the control torques given in Equation (76) into Equation (87):

$$\begin{aligned} \dot{V}_1 &= -s_1 \tilde{\gamma} (\dot{v}_c(t) + \beta e_v(t)) - K_1 s_1^2 - \eta_1 s_1 \tanh\left(\frac{s_1}{\varepsilon}\right) \\ &\quad - \Delta_1 s_1 + \frac{1}{a} \tilde{\gamma} \dot{\tilde{\gamma}} \\ \dot{V}_2 &= -s_2 \tilde{\alpha} (\dot{w}_c(t) + \beta e_w(t)) - K_2 s_2^2 - \eta_2 s_2 \tanh\left(\frac{s_2}{\varepsilon}\right) \\ &\quad - \Delta_2 s_2 + \frac{1}{b} \tilde{\alpha} \dot{\tilde{\alpha}} \end{aligned} \quad (88)$$

The inequality can be obtained for every scalar value of

s ($s \in R$) and positive scalar ε :

$$\begin{aligned} -\eta_1 s_1 \tanh\left(\frac{s_1}{\varepsilon}\right) &\leq -\eta_1 |s_1| + \eta_1 \mu \varepsilon \\ -\eta_2 s_2 \tanh\left(\frac{s_2}{\varepsilon}\right) &\leq -\eta_2 |s_2| + \eta_2 \mu \varepsilon \end{aligned} \quad (89)$$

where μ is the nominal control inputs.

The suitable adaptive law for ASMC is selected from [5,34]:

$$\begin{aligned} \dot{\hat{\gamma}} &= -a s_1 (\dot{v}_c + \beta e_v) \\ \dot{\hat{\alpha}} &= -b s_2 (\dot{w}_c + \beta e_w) \end{aligned} \quad (90)$$

Therefore,

$$\begin{aligned} \dot{V}_1 &= -K_1 s_1^2 - \eta_1 |s_1| + \eta_1 \mu \varepsilon + \Delta_1 s_1 \\ &\leq -K_1 s_1^2 + \eta_1 \mu \varepsilon \leq 0 \\ \dot{V}_2 &= -K_2 s_2^2 - \eta_2 |s_2| + \eta_2 \mu \varepsilon + \Delta_2 s_2 \\ &\leq -K_2 s_2^2 + \eta_2 \mu \varepsilon \leq 0 \end{aligned} \quad (91)$$

Since $|\Delta| \leq D$, ASMC asymptotic stability is verified. ■

Simulations

The proposed controller algorithms are tested in the MATLAB/Simulink environment with numerical simulation. The block diagram containing the whole system is given in Figure 4. The reference block shown at left produces the reference position vector. The error is obtained by taking the difference between the current position vector obtained in the kinematic model and the reference position vector. Since the error vector is in the inertial frame, it is necessary to transform the error vector into the robot frame. The kinematic controller obtains the current linear and angular velocities by using the position errors in the robot frame and reference velocities. The DDMR system needs the variables K_x, K_y and K_θ with respect to its position in the state space. Kinematic controller gain values in adaptive structures are obtained by the neural network block. ASMC generates a control signal using current velocities from the kinematic controller, velocities from the nonlinear model, and wheel angular velocities from the kinematic model. As illustrated in Figure 4, the transition between controller strategies, namely PID, SMC, and ASMC, can be provided by a switch. Unlike ASMC, the PID controller generates the control signal without using wheel angular velocities. Disturbances are added to the output of the control signals for all controller approaches that we used in this work. The linear and angular velocities of the robot are obtained with the nonlinear dynamic model. The fault structure is modelled in the kinematic model. The system position vectors are obtained in the block diagram shown on the far right in Figure 4, and the LOE factors (k_L and k_R) are taken 50%. With these faults, the wheels of the mobile robot are slowed down. The maximum and minimum values of m_0 and I_0 are taken as $17.51 \text{ kg} \leq m_0 \leq 37.51 \text{ kg}$ and $0.57 \text{ kgm}^2 \leq$

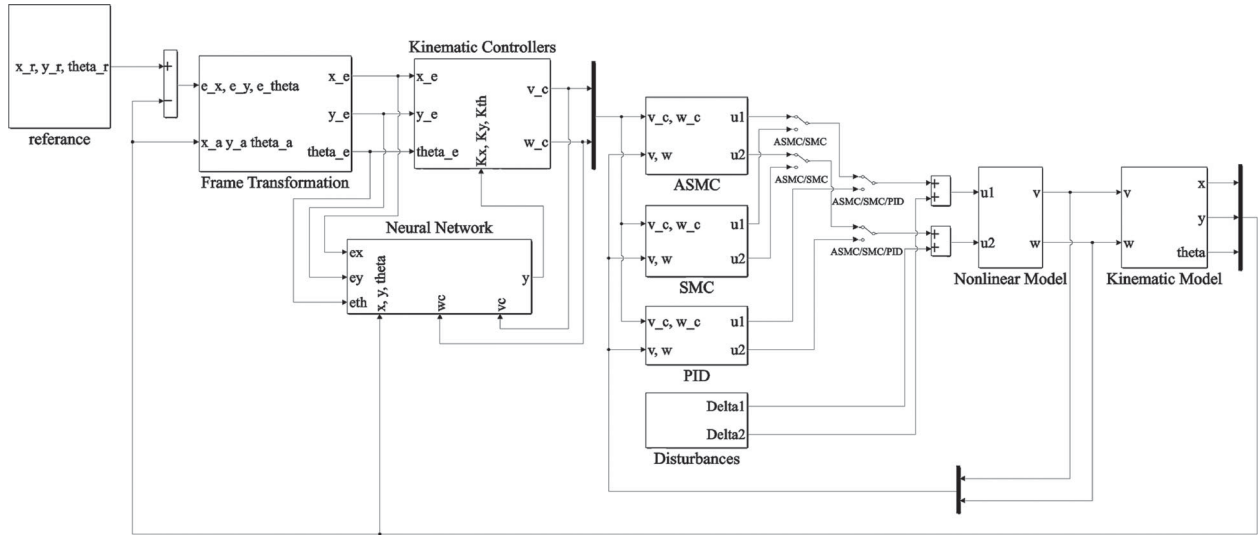


Figure 4. DDMR controller and modelling.

Table 1. Controller gains and parameters.

Parameter	Value	Parameter	Value
x_r	t	m_w	0.5kg
y_r	$\sin(t)$	m_c	17kg
θ_r	0	R	0.095 m
v_r	0.06m/s	L	0.24 m
w_r	0rd s ⁻¹	d	0.05
x_0	0	I_c	0.537kgm ²
y_0	0	I_m	0.0011kgm ²
θ_0	0	I_w	0.0023kgm ²
β	5	ε	0.05
a	0.5	b	0.5
K_1	0.171	K_2	0.171
K_x	2	K_y	150
K_θ	5		

$I_0 \leq 4.57\text{kgm}^2$. The mobile robot, which is subject to both faults and disturbances, tries to follow its reference trajectory. Ode45 is used to solve non-stiff differential equations in systems with ASMC, SMC, and PID. In each case, same disturbances which are $\Delta_1 = 0.25 * \sin(3 * t)$ and $\Delta_2 = 0.75 * \cos(2 * t)$ is used for fair comparison. Unstable signals, which are disturbances, are added to the control signals. The noise signals are more unstable at the beginning of the simulation. For this reason, the mobile robot has difficulty settling into the reference trajectory.

Reference and robot parameters are shown in Table 1.

The initial values of K_x , K_y and K_θ values are given in Table 1. For the PID control approach, two different PID controllers are used for linear and angular velocities. PID gain values have different values because the two systems use two different error values. PID1 and PID2 generate control signals using linear velocity error and angular velocity, respectively. To determine PID controller gains, PID tuner application (transfer function based) is used in the MATLAB/Simulink environment. The PID Tuner app automatically tunes the gains of a PID controller for a SISO plant, which is a

Table 2. PID parameters.

PID Parameter	PID1	PID2
	Value	Value
K_P	77.18	5.32
K_I	35.64	21.86
K_D	-0.01	-0.06
N	54.8	77.51

mobile robot system, to achieve a balance between performance and robustness. Here, the balance between settling time and robustness is provided with the PID controller gains, which are given in Table 2.

Although the system is subjected to faults and disturbances, as shown in Figure 5, it shows a satisfactory result in TT with the proposed controller structures. In the sixth second, a fault occurs in the system, and the fault continues until the end of the simulation. Therefore, a deterioration in the condition of the trajectory tracking occurs after sixth second. If SMC is compared with the PID system, there are small differences. However, the ASMC system gives much better results than the PID and SMC systems in the curved parts of the reference trajectory, and the ASMC system tracks closer to the reference trajectory in the linear parts. The system with a PID controller settles into the reference trajectory later than the ASMC and SMC systems. The three-controller system gives satisfactory results in TT compared with [6,39–41].

As can be seen in Figure 6, the linear velocity increases until the DDMR catches the reference trajectory. Afterwards, it progresses stably. When the fault occurs in the sixth second, linear velocity values increase in DDMR systems. Because the fault model in the actuators tries to reduce the wheel velocities of the DDMR. The controller system does not allow the robot to stop or slow down by increasing the velocity. If the linear velocity values in the ASMC system are compared with those in the PID and SMC systems, it is seen

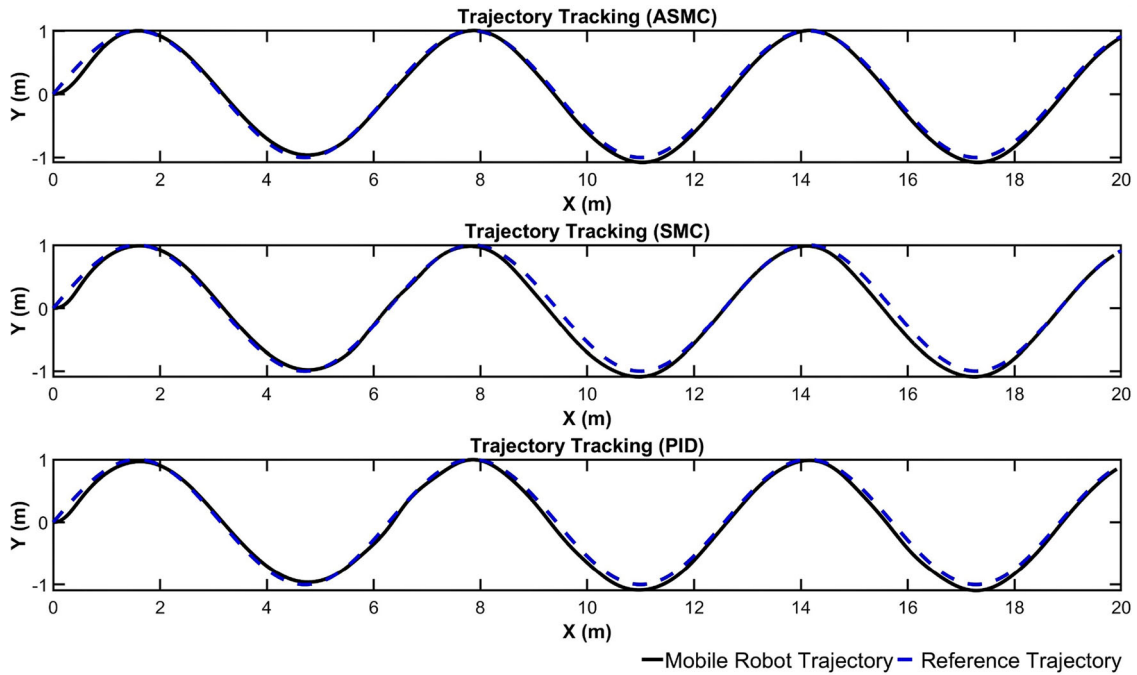


Figure 5. Trajectory tracking.

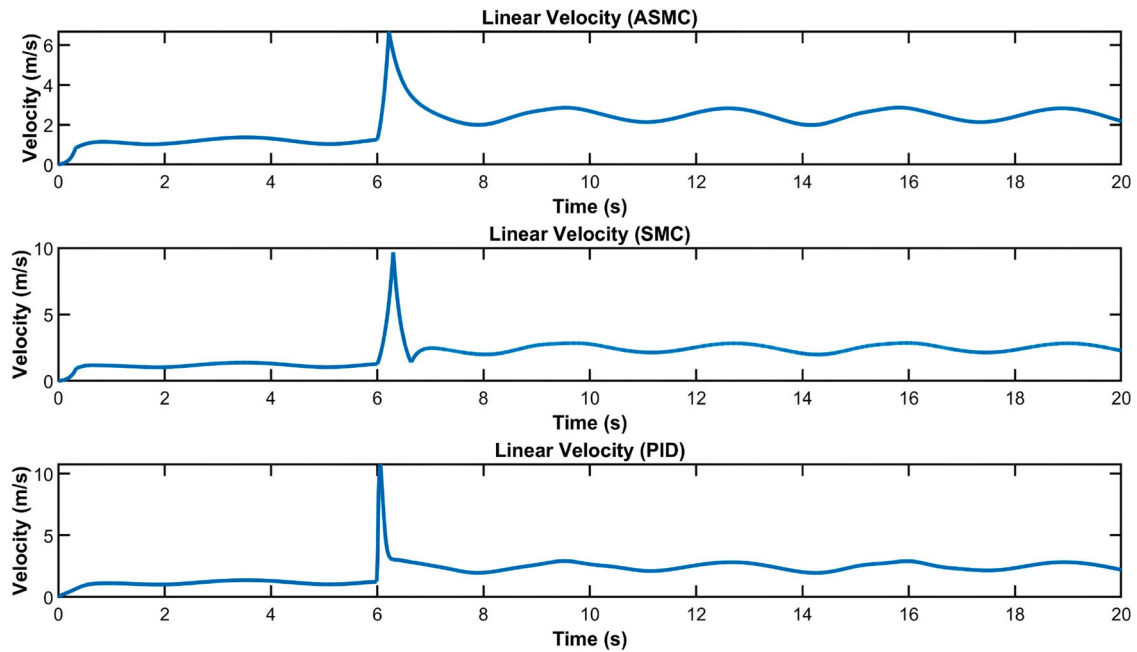


Figure 6. Linear velocity.

that the two figures are very similar. The mobile robot with the PID system initially accelerates to catch the trajectory more slowly than the ASMC and SMC systems. After the eighth second, it progresses stably, similar to a sine wave.

In Figure 7, the angular velocity changes abruptly until it settles onto the reference trajectory. There is more fluctuation after the sixth second, as does the linear velocity. The difference between the angular velocity and the linear velocity is that it can take negative values according to the direction of the mobile robot. If the angular velocity values in the ASMC system are compared with the PID and SMC systems, it is seen that the

angular velocity graphs are almost the same due to the reference trajectory being important for angular velocity change. The difference here is that the fluctuation is greater in the PID system due to the structure of the PID.

In Figure 8, control torques which are generated for both right and left wheels by each controller can be seen. Here there are high torque values when the DDMR starts to move from its initial state and the fault occurs in the sixth second, so the DDMR moves with acceleration. This is because the controllers are trying to push the mobile robot to follow the reference trajectory. After DDMR catches the reference line, left and right

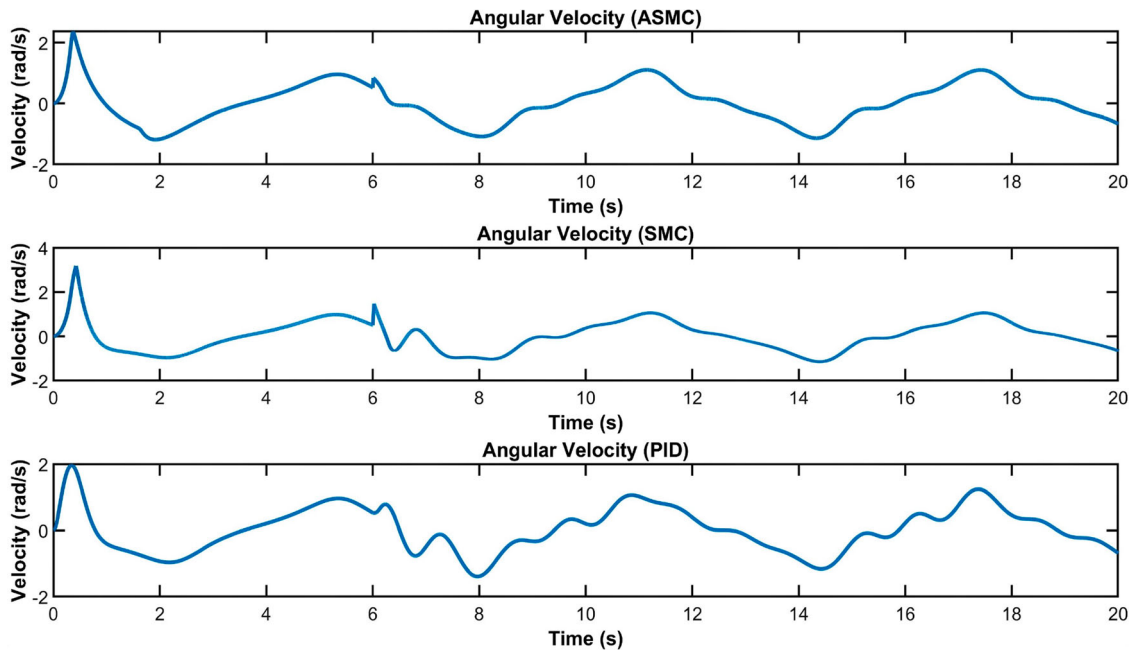


Figure 7. Angular velocity.

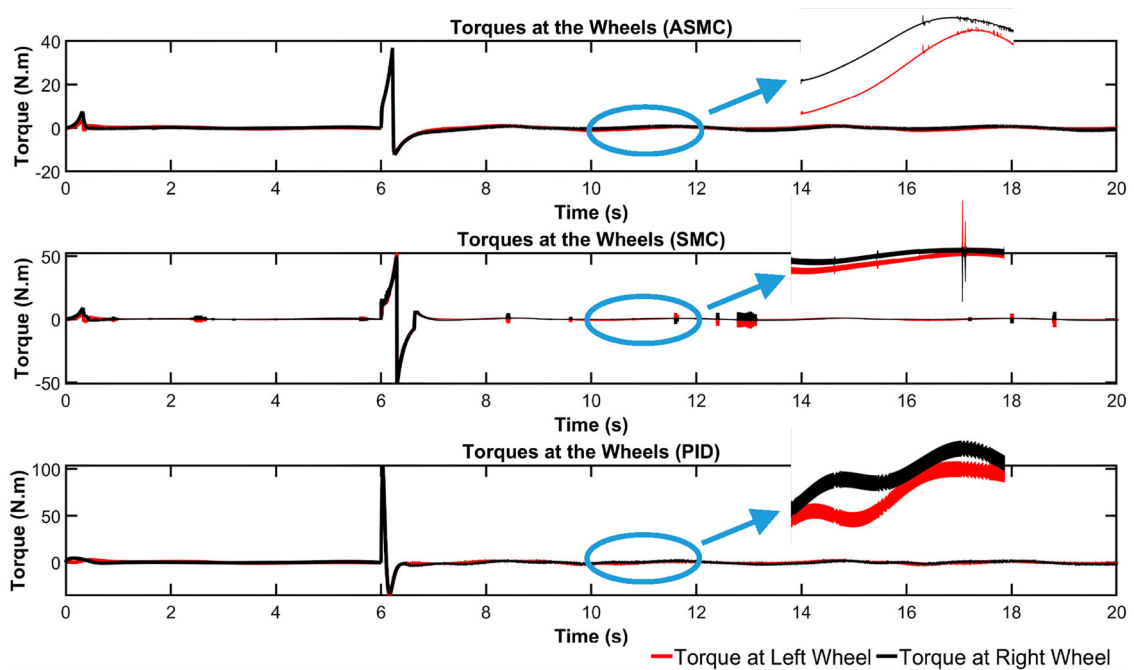


Figure 8. Controller torque.

wheel torques decrease, and the control signals for all three ASMC, SMC, and PID tend to go to zero. All three systems perform satisfactorily in TT, but SMC and PID controller have more chattering than ASMC, so ASMC produces more smooth control inputs.

Figure 9 shows that the error graphs are similar for all three control strategies. The DDMR system has a higher error value until the mobile robot catches the reference line. Afterwards, the error value decreases, and the process proceeds stably. When the fault occurs in the sixth second, the error values increase, and then the error value decreases, and the process proceeds

stably again. In DDMRs with fault models, the system error does not disappear because the actuators are constantly given error values over time and disturbances are always present when the system is operating.

To compare the results more accurately, Root Mean Square (RMS) values of the errors for all three systems are given in Table 3.

As can be seen from Table 3, the ASMC gives better results compared to the PID controller and SMC, but the difference is small since the kinematic and NN-based adaptive backstepping controllers have effective operation.

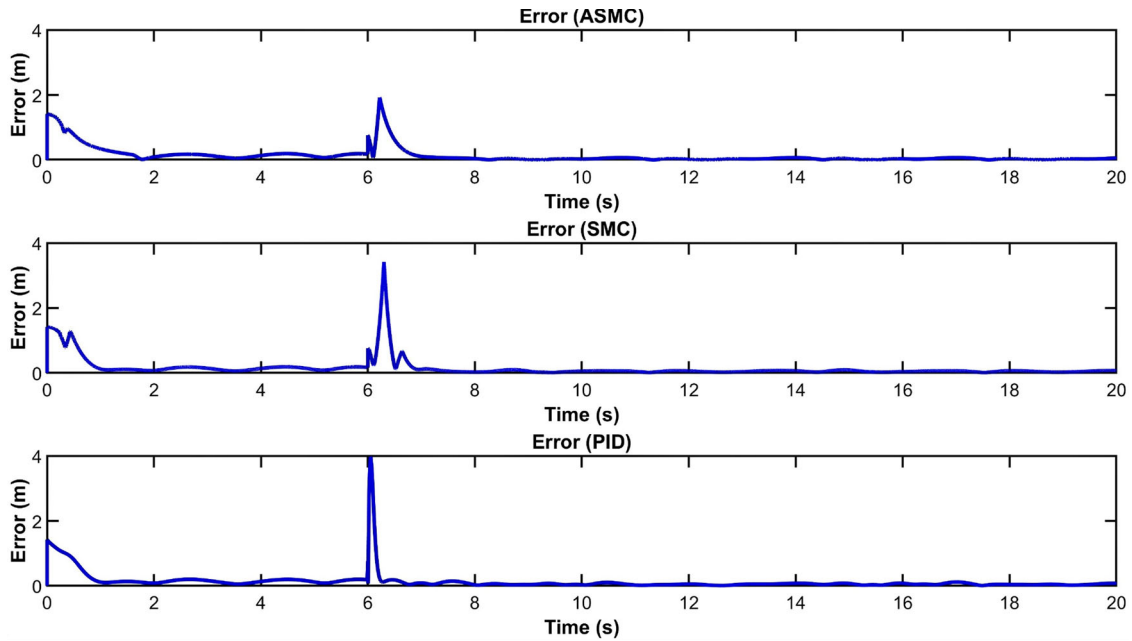


Figure 9. Trajectory tracking errors.

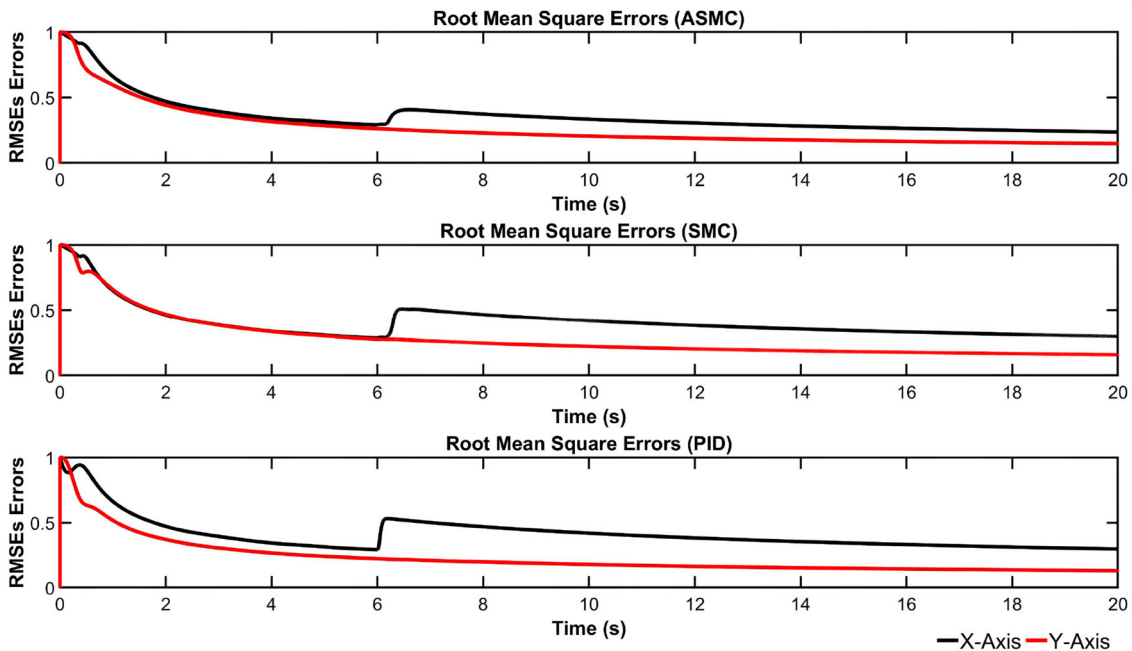


Figure 10. RMS error

Table 3. RMSEs average values.

Axis	DDMR with ASMC Average error	DDMR with SMC Average error	DDMR with PID Average error
X	0.3293	0.3852	0.4084
Y	0.2097	0.2397	0.2266

Figure 10 shows that the RMS error graphs are similar for all three control strategies. It starts at zero and suddenly reaches a value of one. The RMS error value decreases and progresses stably until the fault occurs in the sixth second, and then the error value decreases and progresses stably again. If the angular velocity values in the ASMC system are compared with the PID and SMC systems, it is clear that the RMS error in the

x-axis is less than the PID and SMC systems in the sixth second.

In Figure 11, the controller gains are compared with each other with respect to the adaptive controller point of view, K_x , K_y and K_θ values are not kept constant for all controllers. The change in the gain values is small because a big change affects the system operation and increases the torque value. Looking at the other changes, it is seen that the gain values do not take very different values from the initial conditions until the fault occurs in the sixth second, and then the gain value decreases and progresses stably again. Here, there are high torque values when the fault occurs. K_y and K_θ values change in very similar ways in ASMC and

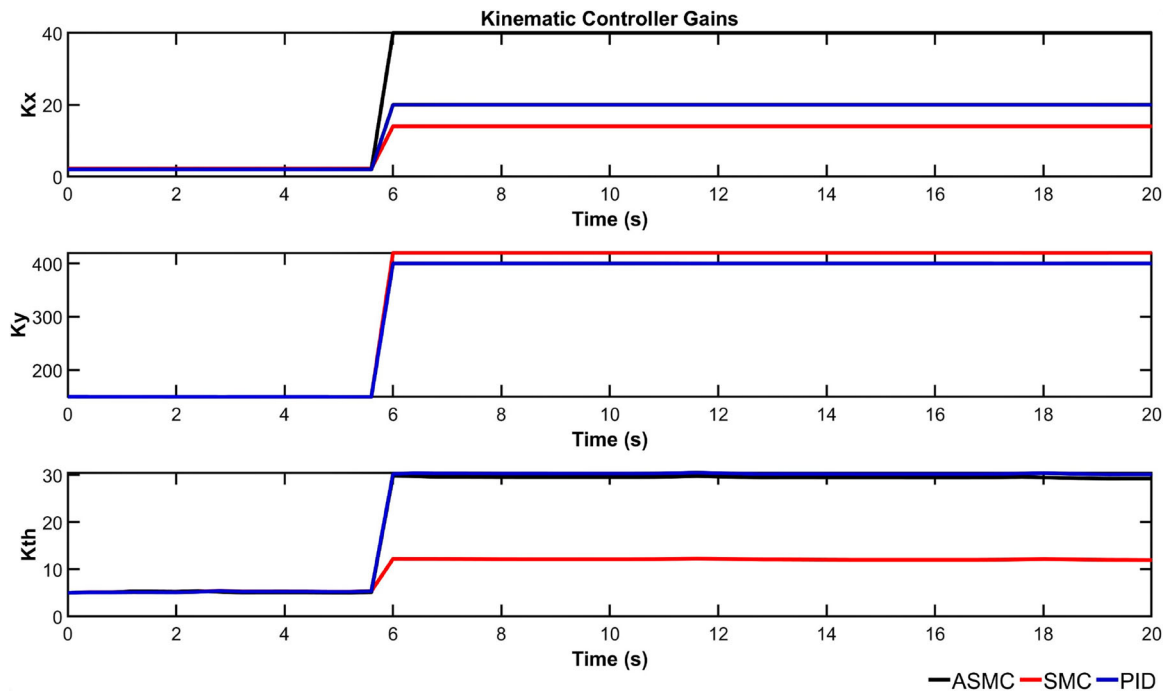


Figure 11. Kinematic controller gain.

PID. K_x values change more in ASMC than in SMC and PID as the sliding mode controller structure has an adaptive law. Due to having a fault in the system, the kinematic controller gain values are high compared to those without a fault modelling system [5,15,42], so that the DDMR has high velocity values.

Conclusion

In this paper, first the dynamic and kinematic models are obtained by using kinematic constraints and the Lagrangian approach. The proposed control methods for TT of the DDMR system are ASMC, SMC, and PID controllers. The effects of disturbances and faults in the system's actuators are eliminated, and the DDMR system follows the reference trajectory by adding kinematic controllers and NN-based controllers to the system. In the real-time application, the robot model is nonlinear since the robot parameters are unknown. For this reason, in this work, an adaptive law is proposed to maintain the sliding state, solving the problem of unknown parameters and disturbances. A comparison is made between the PID controller, which is mostly used in linear systems, the SMC, and the adaptive SMC used in nonlinear systems, and responses under faults and disturbances are examined.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Mustafa Ayyıldız  <http://orcid.org/0000-0003-0294-3822>
Umut Tilki  <http://orcid.org/0000-0002-8988-787X>

References

- [1] Parisa Y, Meskin N. Design and real-time implementation of actuator fault-tolerant control for differential-drive mobile robots based on multiple-model approach. *Proc Inst Mech Eng.* 2018;232(6):652–661.
- [2] Li D, Ye J. Adaptive robust control of wheeled mobile robot with uncertainties. In *Advanced Motion Control (AMC), 2014 IEEE 13th International Workshop on*, pp. 518–523, 2014.
- [3] Laumond J-P. *Robot motion planning and control*, Vol. 229. New York: Springer; 1998.
- [4] Kanayama Y, Kimura Y, Miyazaki F, et al. A stable tracking control method for an autonomous mobile robot. In *Robotics and Automation, 1990. Proceedings. 1990 IEEE International Conference on*, pp. 384–389, 1990.
- [5] Mevo BB, Saad MR, Fareh R. Adaptive sliding mode control of wheeled mobile robot with nonlinear model and uncertainties. In *Electrical and Computer Engineering, IEEE Canadian Conference on*, pp. 1–5, 2018.
- [6] Koubaa Y, Boukattaya M, Dammak T. Adaptive sliding-mode dynamic control for path tracking of non-holonomic wheeled mobile robot. *J Autom Syst Eng.* 2015;9:119–131.
- [7] Baek J, Kwon W. Practical adaptive sliding-mode control approach for precise tracking of robot manipulators. *Appl Sci.* 2020;10(8):2909.
- [8] Jin XZ, Zhao YX, Wang H, et al. Adaptive fault-tolerant control of mobile robots with actuator faults and unknown parameters. *IET Control Theory Applic.* 2019;13(11):1665–1672.
- [9] Ma Y, Cocquempot V, El Najjar MEB, et al. Fault-tolerant control for physically linked two 2WD mobile robots with actuator faults. *IFAC-PapersOnLine.* 2017; 50(1):13563–13568.
- [10] Moradi M, Fekih A. A stability guaranteed robust fault tolerant control design for vehicle suspension systems subject to actuator faults and disturbances. *IEEE Trans Control Syst Technol.* 2014;23(3):1164–1171.
- [11] Zhang D, Liu G, Zhou H, et al. Adaptive sliding mode fault-tolerant coordination control for four-wheel

- independently driven electric vehicles. *IEEE Trans Ind Electron.* **2018**;65(11):9090–9100.
- [12] Kamel MA, Zhang Y, Yu X. Fault-tolerant cooperative control of multiple wheeled mobile robots under actuator faults. *IFAC-PapersOnLine.* **2015**;48(21):1152–1157.
- [13] Zhang X, Cocquempot V. Fault tolerant control for an electric 4WD vehicle's path tracking with active fault diagnosis. *IFAC Proc Vol.* **2014**;47(3):6728–6734.
- [14] Karras GC, Fourlas GK. Model predictive fault tolerant control for omni-directional mobile robots. *J Intell Robot Syst.* **2020**;97(3):635–655.
- [15] Mohareri O. Mobile robot trajectory tracking using neural networks. Degree of Master of Science in Mechatronics Engineering. Sharjah: American University of Sharjah; **2009**.
- [16] Alshorman AM, Alshorman O, Irfan M, et al. Fuzzy-based fault-tolerant control for omnidirectional mobile robot. *Machines.* **2020**;8(3):55.
- [17] Liu Q, Cong Q. Kinematic and dynamic control model of wheeled mobile robot under internet of things and neural network. *J Supercomput.* **2022**;78(6):8678–8707.
- [18] Jin X. Fault-tolerant iterative learning control for mobile robots non-repetitive trajectory tracking with output constraints. *Automatica (Oxf).* **2018**;94:63–71.
- [19] Yoo SJ, Kim TH. Predesigned fault-tolerant formation tracking quality for networked uncertain nonholonomic mobile robots in the presence of multiple faults. *Automatica (Oxf).* **2017**;77:380–387.
- [20] Lewis FL, Abdallah CT, Dawson DM. Control of robot manipulators. Editorial Maxwell McMillan, Canada, pp. 25–36, 1993.
- [21] Demirbaş F, Kalyoncu M. Differential drive mobile robot trajectory tracking with using PID and kinematic based backstepping controller. *Selcuk Univ J Eng Sci Technol.* **2017**;5(1):1–15.
- [22] Ayyıldız M, Tilki U. "Adaptive sliding mode control of mobile robot and comparison with PID controller," International Symposium on Applied Sciences and Engineering (ISASE), 7–9 April, pp. 494–498, 2021.
- [23] Raj L, Czmerk A. Modelling and simulation of the drivetrain of an omnidirectional mobile robot. *Automatika: Časopis za Automatiku, Mjerenje, Elektroniku, Računarstvo i Komunikacije.* **2017**;58(2):232–243.
- [24] Hanna DK, Joukhadar A. A novel control-navigation system-based adaptive optimal controller and EKF localization of DDMR. *Int J Adv Res Artif Intell.* **2015**;4(5):21–29.
- [25] Contreras JCM, Herrera D, Toibero JM, et al. Controllers design for differential drive mobile robots based on extended kinematic modeling. European Conference on Mobile Robots (ECMR). IEEE, 2017.
- [26] Lamraoui HC, Qidan Z, Benrabah A. Dynamic velocity tracking control of differential-drive mobile robot based on LADRC. In International Conference on Real-time Computing and Robotics (RCAR), pp. 633–638, IEEE, 2017.
- [27] Dhaouadi R, Hatab AA. Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: a unified framework. *Adv Robot Automat.* **2013**;2:1–7.
- [28] Bouzoualegh S, Guechi E-H, Kelaiaia R. Model predictive control of a differential-drive mobile robot. *Electr Mech Eng.* **2018**;10:20–41.
- [29] Fierro R, Lewis FL. Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. *J Robot Syst.* **1997**;14(3):149–163.
- [30] Ye J. Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique. *Neurocomputing.* **2008**;71(16-18):3373–3378.
- [31] Fierro R, Lewis FL. Control of a nonholonomic mobile robot using neural networks. *IEEE Trans Neural Netw.* **1998**;9(4):589–600.
- [32] Velagic J, Osmic N, Lacevic B. Neural network controller for mobile robot motion control, *World Academy of Science. Eng Technol.* **2008**;47:193–198.
- [33] Qian D, Tong S, Xu C. Robust multi-robot formations via sliding mode controller and fuzzy compensator. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije.* **2016**;57(4):1007–1019.
- [34] Liu J. Sliding mode control using MATLAB. Beijing: Academic Press; **2017**.
- [35] Xu L, Yao B. Adaptive robust control of mechanical systems with nonlinear dynamic friction compensation. *Int J Control.* **2007**;81(2):167–176.
- [36] Sastry S, Bodson M. Adaptive control: stability, convergence, and robustness. Mineola, NY: Courier Corporation; **2011**.
- [37] Ranjbar E, Yaghubi M, Suratgar AA. Robust adaptive sliding mode control of a MEMS tunable capacitor based on dead-zone method. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije.* **2020**;61(4):587–601.
- [38] Barsan A. Position control of a mobile robot through PID controller. *Acta Universitatis Cibiniensis, Technical Series.* **2019**;71:14–20.
- [39] Amer AF, Sallam EA, Sultan IA. Adaptive sliding-mode dynamic controller for nonholonomic mobile robots. In 12th International Computer Engineering Conference (ICENCO). IEEE, 2016.
- [40] Al-dujaili A, Cocquempot V, Najjar M, et al. Actuator fault compensation tracking control for multi linked 2WD mobile robots. In 25th Mediterranean Conference on Control and Automation (MED), IEEE, 2017.
- [41] Srebro A. A self-tuning fuzzy PD controller for a wheeled mobile robot operating in the presence of faults. *Challenges of Modern Technology.* **2011**;2(4):11–21.
- [42] Ayyıldız M, Tilki U. Fault Tolerant Control of Wheeled Mobile Sliding Mode Controller. In 2021 International Conference on INnovations in Intelligent Systems and Applications (INISTA), 2021, pp. 1–6. DOI:10.1109/INISTA52262.2021.9548505