Trajectory Tracking of a Two-Wheeled Mobile Robot Using Backstepping and Nonlinear PID Controller

Lencho Duguma Fufa[1,2](✉) and Endalew Ayenew[1]

[1] Electrical Power and Control Engineering, School of Electrical Engineering and Computing, Adama Science and Technology University, Adama, Ethiopia
lencho.duguma@mwu.edu.et

[2] Electrical and Computer Engineering, College of Engineering, Madda Walabu University, Bale Robe, Ethiopia

Abstract. Many researchers have become interested in wheeled mobile robot (WMR) trajectory tracking control in recent years. This is due to the increased application of mobile robots in the industry, the military, the home, and public service. Classically, the movement of WMR is controlled depending on its kinematic model. However, in real-time applications, both the dynamic and kinematic models of robots and external disturbance and uncertainty affect system performance. This paper proposes backstepping combined with a Nonlinear Proportional-Integral-Derivative (NPID) controller to control a two-wheeled mobile robot (TWMR). The kinematic and dynamic models of the WMR are derived. The dynamic modeling is derived using a Lagrangian approach, and stability of the system is achieved using the Lyapunov method. The controller gains are optimized using the Genetic Algorithm optimization technique. The proposed algorithms' performance is tested using Matlab software. The simulation result shows that the proposed method achieved preferable reference trajectory tracking with a minimum tracking error. The proposed controller outperforms the GA-based backstepping plus PID controller in terms of root-mean-square (RMS) of trajectory tracking error (47.36% in a linear and 60.32% in a nonlinear case). In addition, it shows good unknown disturbance rejection and initial point change in all scenarios.

Keywords: Backstepping and NPID Controller · Trajectory Tracking · Two-Wheeled Mobile Robot · Genetic Algorithm · Lyapunov Stability Analysis

1 Introduction

A mobile robot is a robot that can move its surroundings and is not fixed to a given physical location. It can be autonomous or semi-autonomous, which means it can navigate in an uncontrolled environment without needing a personal operator or electro-mechanical guidance device. A wheeled mobile robot (WMR) is a mobile robot that navigates its surroundings using a wheel's rotation. It is a widely used mobile robot due to its ability

B. H. Woldegiorgis et al. (Eds.): ICAST 2022, LNICST 455, pp. 290–304, 2023.
https://doi.org/10.1007/978-3-031-28725-1_18

to substitute humans in many fields. For example, in industry, the military, public and private sector services, and production and distribution companies.

When the application of a WMR increases, the performance acquired from the robot becomes a critical issue for researchers. The robot's performance (in the case of trajectory tracking) depends on the position error and orientation error for kinematic control, while a velocity error will affect the performance in the case of dynamic control. The movement or navigation of a WMR depends on three control problems: reference trajectory tracking, line following, and point stabilization [1]. One of the main objectives of reference tracking control is to control a robot's position on a predetermined trajectory with the minimum position and orientation error. Due to nonholonomic restrictions in nonholonomic robots, trajectory tracking and motion control are not independent. Trajectory tracking of nonholonomic robots in the robotic domain involves determining trajectories from a starting point to a final point while considering mechanical limitations and guiding the robot to follow the proposed trajectories. Several works on the WMR's trajectory control problem have been presented. Most of them are focused on a nonholonomic constraint that depends on posture stabilization and trajectory tracking [2]. Trajectory tracking makes the robot follow a predefined trajectory.

The two-wheeled mobile robot (TWMR) used in this work has two standard fixed wheels that are actuated by two similar DC motors and a freewheeler. The freewheeler is used to balance the robot's body frame. Many control approaches have been applied to control WMR trajectory tracking, for example, a kinematics-based backstepping controller [3], Proportional-Integral-Derivative (PID) controller [4], Fuzzy logic controller [5, 6], and Nonlinear PID controller. Furthermore, various optimization techniques (like; Genetic Algorithm (GA), Particle Swarm Optimization, Grey Wolf Optimization, and Neural Network) are used to optimize the controller gains.

In [7, 8], a kinematic controller combined with a torque control law using the backstepping control method is proposed for nonholonomic mobile robot control. Hassani et al. proposed a backstepping method for trajectory tracking control of mobile robots (MR) [9]. The authors presented kinematic and dynamic models. The challenging task of this approach is that controller gains are obtained by trial and error and are less efficient.

In works by [10], a nonholonomic WMR is controlled by a Nonlinear Proportional-Integral-Derivative (NPID) neural controller with a particle swarm optimization algorithm. The Neural Network-based PID controller responds smoothly to the external attenuation disturbance problem. The control method was based on a kinematic model of the system by ignoring the robot's dynamic model. In practical applications, both dynamic and kinematic models of robots affect system performance. In the literature [11] and [12] kinematic-based backstepping controller is applied to control robot coordinates, and a PID controller is used for motor speed control. In [13], a combination of PID and backstepping approach is used for trajectory control. However, it lacks control parameter optimization and leads to the system being controlled only in an interesting region.

Preferable trajectory tracking of WMR is achieved by considering the disturbances, noise, or internal model changes (uncertainty) that will affect the system in real-time applications. Including a WMR system dynamic model is vital for good trajectory tracking and stabilization. The main contribution of this paper is the design of a backstepping controller combined with an NPID controller for stabilization and trajectory control of TWMR. The proposed strategies consist of two approaches. A backstepping controller is used at the control robot's position, and NPID controls the robot's velocity. The Lyapunov method justifies the stability of the system. Kinematic and dynamic models of a proposed framework are taken into account.

Additionally, the controller parameters are also optimized using a GA optimization method. The developed control method is tested by simulation on Matlab/Simulink software. The system's performance will be analyzed with unknown disturbances and changes in the initial position. The proposed system tracking capability is compared with a GA-based backstepping controller and a GA-based backstepping plus PID controller.

The next sections of this paper present system modeling, control approach design, results and discussions, and conclusions.

2 System Modeling

2.1 Kinematic Modeling of the TWMR

A system modeling of the differential drive MR platform in this study consists of kinematic and dynamic modeling. As seen in Fig. 1, the WMR is represented in Cartesian coordinates. The robot setup body has two wheels with a radius of r and a distance L from the center P and free castor wheels to balance the robot setup body. C is the robot's center of mass (CoM), and d is the distance between the CoM and the center of the wheel axis. The position of the WMR in the inertial reference frame (X_I, Y_I) with origin point, O, and the local robot frame (X_R, Y_R) they are attached to the body frame. The angle between the local robot and global reference frames is represented as theta (θ) in radian. Moreover, v and ω are robot linear and angular velocities, respectively.

Let us represent the robot coordinates in inertial and local frames as follows [14]:

$$q_I = \begin{bmatrix} x_I \ y_I \ \theta_I \end{bmatrix}^T \tag{1}$$

$$q_R = \begin{bmatrix} x_R \ y_R \ \theta_R \end{bmatrix}^T \tag{2}$$

A robot's motions in the global frame are translated into motions in the local frame by using a standard orthogonal transformation matrix and vice versa. I.e.,

$$\dot{q}_R = R(\theta)\dot{q}_I \tag{3}$$

where:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The contribution of the translational and rotational velocity components of the wheel velocities in the local robot frame is as follows:

$$\begin{bmatrix} \dot{x}_R, \dot{y}_R, \dot{\theta}_R \end{bmatrix}^T = \begin{bmatrix} \frac{1}{2} r(\dot{\varphi}_r + \dot{\varphi}_l), 0, \frac{r}{2L}(\dot{\varphi}_r - \dot{\varphi}_l) \end{bmatrix}^T \tag{5}$$

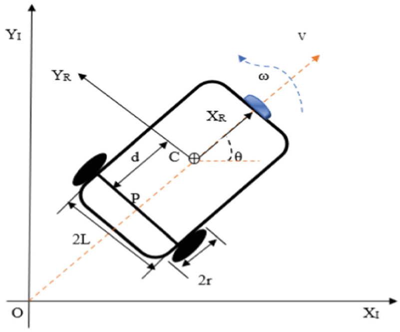

Fig. 1. 2D representation of WMR

where $\dot{\varphi}_r$ *and* $\dot{\varphi}_l$ is the angular velocity of the right and left wheels, respectively. In an inertial frame, the differential drive WMR velocity is given as:

$$\begin{bmatrix} \dot{x} \; \dot{y} \; \dot{\theta} \end{bmatrix}^T = \frac{r}{2}\begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ 1/L & -1/L \end{bmatrix}\begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} \tag{6}$$

In Fig. 1, a wheel presents three kinematic constraints on DWMR. The first constraint is that DMWR cannot slide sideways, i.e., no non-slipping constraint exists, while the others are related to the motion of the wheels. The actuated wheels cannot rotate in the wrong direction. I.e., there are only pure rolling constraints [15, 16].

$$\dot{y}\cos(\theta) - \dot{x}\sin(\theta) - d\dot{\theta} = 0 \tag{7}$$

$$\begin{aligned} \dot{x}\cos(\theta) + \dot{y}\sin(\theta) + L\dot{\theta} - r\dot{\varphi}_r = 0 \\ \dot{x}\cos(\theta) + \dot{y}\sin(\theta) - L\dot{\theta} - r\dot{\varphi}_l = 0 \end{aligned} \tag{8}$$

The nonholonomic constraint in Eqs. (7) and (8) can be written in matrix form as:

$$A(q)\dot{q} = 0 \tag{9}$$

where A(q) is the constraint matrix and $\dot{q}$ is configuration coordinate given as:

$$A(q) = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & -d & 0 & 0 \\ \cos(\theta) & \sin(\theta) & L & -r & 0 \\ \cos(\theta) & \sin(\theta) & -L & 0 & -r \end{bmatrix} \tag{10}$$

$$\dot{q} = \begin{bmatrix} \dot{x} \; \dot{y} \; \dot{\theta} \; \dot{\varphi}_r \; \dot{\varphi}_l \end{bmatrix}^T \tag{11}$$

If the inertial and mass of the wheels were neglected, WMR satisfied pure rolling and non-slipping. As a result, Eq. (10) is reduced to the following matrix:

$$A(q) = \begin{bmatrix} -\sin(\theta) \; \cos(\theta) \; -d \end{bmatrix} \tag{12}$$

Consider S(q) as a smooth and independent vector field distributed in the null space of a matrix A(q).

$$A(q)S(q) = 0 \tag{13}$$

In this case, S(q) is a linearly independent distributed field vector that is used to transform velocities w(t) = [v, ω] in terms of the inertial reference frame given as:

$$S(q) = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & d\cos(\theta) \\ 0 & 1 \end{bmatrix} \tag{14}$$

Therefore, the WMR kinematic model in an inertial frame is formulated as follows:

$$\dot{q} = S(q)w(t) = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & d\cos(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{15}$$

where w(t) is the velocity vector given as $w(t) = \begin{bmatrix} v & \omega \end{bmatrix}^T$, $w \in R^{p \times 1}$ for all t.

2.2 System Dynamic Modeling

An MR system having an n-dimensional configuration space R, with generalized coordinate $q = [q_1, q_2, \ldots, q_n]$ and subject to m input constraints can be described as [15];

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} = E(q)\tau - A^T(q)\rho \tag{16}$$

where H(q) is an inertial matrix represented as n × n positive definite matrix, $C(q, \dot{q})$ is Coriolis and Centripetal torques represented as n × n matrix, ρ is a vector associated with Lagrange multipliers kinematic constraints, E(q) is the control input of n × m transformation matrix, $A^T(q)$ is matrix associated with constraints, τ is input torque vector. The generalized Lagrangian form of WMR for fixed conventional wheels is given as follows [16]:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = E(q)\tau - A^T(q)\rho \tag{17}$$

The total lagrangian energy is defined as L = T − W, where T is kinetic energy and W is potential energy (equal to zero because there is no lateral movement, only horizontal movement). The total lagrangian energy of the robot is calculated by considering a dynamic constraint on a fixed standard wheel.

$$L = \frac{1}{2}m_t(\dot{x}^2 + \dot{y}^2) + m_t(\dot{x}d\dot{\theta}\sin(\theta) - \dot{y}d\dot{\theta}\cos(\theta)) + \frac{1}{2}I\dot{\theta}^2 + \frac{1}{2}I_w(\dot{\varphi}_r^2\,\dot{\varphi} + \dot{\varphi}_l^2) \tag{18}$$

where $m_t = m_c + 2m_w$ and $I = m_c d^2 + I_c + 2m_w\left(d^2 + L^2\right) + 2I_m$. m_c is a mass of WMR devoid of wheels and motors, and m_w is the combined mass of wheel and motor, and m_t is the total mass of the robot. I_c is the inertia of WMR without wheels and motor, I_w is the inertia of a single wheel and motor around its axis, I_m is the inertia of the single wheel and motor about the y-axis is parallel to the wheel plane, and I is the total inertia of the robot.

The MR dynamic modeling is obtained by substituting the Lagrangian energy expression in Eq. (18) into Eq. (17) and, after rearrangements, we obtained as follows:

$$\begin{bmatrix} m_t & 0 & m_t d\sin(\theta) \\ 0 & m_t & -m_t d\cos(\theta) \\ m_t d\sin(\theta) & -m_t d\cos(\theta) & I \end{bmatrix}\ddot{q} + \begin{bmatrix} 0 & 0 & m_t d\dot{\theta}\cos(\theta) \\ 0 & 0 & m_t d\dot{\theta}\sin(\theta) \\ 0 & 0 & 0 \end{bmatrix}\dot{q}$$
$$= \frac{1}{r}\begin{bmatrix} \cos(\theta) & \cos(\theta) \\ \sin(\theta) & \sin(\theta) \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} + \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \\ -d \end{bmatrix}(-m_t(\dot{x}\cos(\theta) + \dot{y}\sin(\theta))) \tag{19}$$

This dynamic model in Eq. (20) is simplified and transformed into a proper representation to eliminate the constraint terms [17]. A simplified dynamic model is given as:

$$\begin{bmatrix} m_t & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} + \begin{bmatrix} 0 & -m_t d\dot{\theta} \\ m_t d\dot{\theta} & 0 \end{bmatrix}\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{20}$$

3 Proposed Method Design

This section develops two proposed control algorithms: the backstepping controller and the NPID controller. Figure 2 shows the proposed controller with a TWMR system.

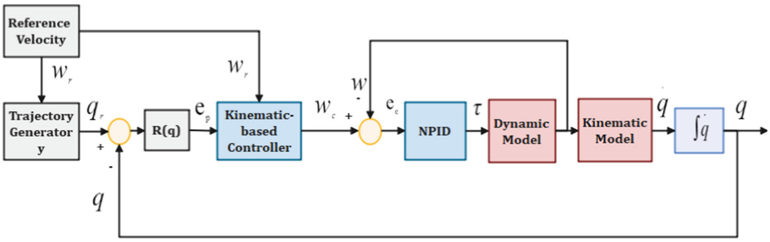

Fig. 2. Kinematic-based backstepping and NPID controller with a proposed system model

3.1 Backstepping Controller

A backstepping control method is a nonlinear controller designed in a recursive way that combines the choice of a Lyapunov function with the design of a feedback controller. A Lyapunov stability analysis function guarantees the global asymptotic stability of the designed controller [18]. In this paper, a backstepping controller is designed based on a nonlinear kinematic controller, as shown in Fig. 2. The main objective of this controller is to determine the system's angular and linear velocity control laws to track a given trajectory with reference velocity inputs and a posture error configuration. A nonholonomic mobile robot trajectory track can be described and formulated as follows:

$$\dot{q}_r = \left[\dot{x}_r\ \dot{y}_r\ \dot{\theta}_r\right]^T = \begin{bmatrix} \cos(\theta_r) & 0 \\ \sin(\theta_r) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \tag{21}$$

From Fig. 2, the posture error between a reference trajectory ($q_r = (x_r, y_r, \theta_r)$) and current robot pose ($q = (x, y, \theta)$) is given as:

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{22}$$

If a CoM is at the center of the wheel axes (d = 0), the time derivatives of posture error become:

$$\dot{e}_p = \left[\dot{e}_x\ \dot{e}_y\ \dot{e}_\theta\right]^T = \begin{bmatrix} \omega\, \mathrm{e}_y - v + v_r \cos(e_\theta) \\ -\omega e_x + v_r \sin(e_\theta) \\ \omega_r - \omega \end{bmatrix} \tag{23}$$

The control law, which makes a system asymptotically stable, will be designed based on the pose error dynamics obtained. Auxiliary velocity control inputs as a function of pose error and input velocities will be proposed depending on the pose error's time derivatives using Lyapunov theory. Let us present a scalar function V as a Lyapunov function candidate as:

$$V = \frac{1}{2}\left(e_x^2 + e_y^2\right) + \frac{1}{K_y}(1 - \cos(e_\theta)) \tag{24}$$

Therefore, for $\mathrm{K_y} > 0$ one can see $\mathrm{V} \geq 0$ otherwise, $\mathrm{V} = 0$ if $\mathrm{e_p} = 0$ and $\mathrm{V} > 0$ if $\mathrm{e_p} \neq 0$. The time-derivative of a Lyapunov function V with a posture error as in Eq. (23) is given as follows:

$$\begin{aligned} \dot{V} &= \dot{e}_x e_x + \dot{e}_y e_y + \frac{1}{K_y}\dot{e}_\theta \sin(e_\theta) \\ &= -e_x[v - v_r \cos(e_\theta)] - \sin(e_\theta)\left[\frac{1}{K_y}(\omega_r - \omega) - v_r e_y\right] \leq 0 \end{aligned} \tag{25}$$

The auxiliary velocity control input that satisfies the Lyapunov function candidate by using a backstepping control law is formulated as follows [7]:

$$w_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} K_x e_x + v_r \cos(e_\theta) \\ \omega_r + K_y v_r e_y + K_\theta v_r \sin(e_\theta) \end{bmatrix} \tag{26}$$

where $\mathrm{K_x}$, $\mathrm{K_y}$, and $\mathrm{K_\theta}$ are positive constants adjusted to reduce the posture error. The velocity control rule described above is called a kinematic-based backstepping controller. The stability of this control velocity is verified by using the Lyapunov function.

$$\dot{V} = \dot{e}_x e_x + \dot{e}_y e_y + \frac{1}{K_y}\dot{e}_\theta \sin(e_\theta) = -K_x e_x^2 - \frac{K_\theta}{K_y} v_r \sin^2(e_\theta) \leq 0 \tag{27}$$

It is clear that $\dot{V} \leq 0$, otherwise, $\dot{V} = 0$ if $e_p = 0$ and $\dot{V} < 0$ if $e_p \neq 0$. Therefore, the derivative of a Lyapunov function is a negative-definite function. The system is asymptotically stable at an equilibrium point $e_p = 0$ under the condition that reference velocity $w_r = \begin{bmatrix} v_r & \omega_r \end{bmatrix}^T$ and proposed constant positive gains $\left(K_x, K_y, \text{and } K_\theta\right)$ are bounded and continuous.

3.2 Nonlinear PID Controller

A NPID controller is used to control the velocity of a robot at a dynamic control loop level, as in Fig. 2. A new NPID controller whose architecture is analogous to the standard PID controller is proposed. However, the proportional and derivative actions are linear, whereas the integral action has a nonlinear function. Hence, the error input to the integral action is scaled by a nonlinear gain function in the product of the error and the nonlinear gain. An NPID controller used in this work is similar to an NPID controller used in work [19].

The time-domain equation of the NPID controller for velocity control of TMWR is given as:

$$\begin{aligned} U_k(t) &= K_{pk}\left[e_c(t) + \frac{1}{T_{ik}}\int_0^t v(t)dt + T_{dk}\frac{de_c(t)}{dt}\right] \\ &= K_{pk}\, e_c(t) + K_{ik}\int_0^t v(t)dt + K_{dk}\frac{de_c(t)}{dt},\ \text{k} = 1{,}2 \end{aligned} \tag{28}$$

where $U_k = [U_1, U_2]^T = [u_v, u_\omega]^T$ is NPID controller output, K_{pk} is proportional gain constant, T_{ik} is an integral time, T_{dk} is derivative time, $K_{ik} = K_{pk}/T_{ik}$ is integral gain, $K_{dk} = K_{pk}T_{dk}$ is derivative gain constant in the case of a parallel PID controller. An error signal $e_c(t)$ is given as the difference between the control and actual velocity ($e_c(t) = w_c - w$). The nonlinear scaled error function $v(t)$ in the integral and derivative control action is given as

$$v(t) = k(e)e_c(t);\ where\ k(e) = \exp\left(-\frac{e_c(t)^2}{2\Delta w_c^2}\right) \tag{29}$$

where $k(e)$ is a nonlinear gain function, $\Delta w_c \neq 0$ is controlled velocity change, i.e., $\Delta w_c = w_c(i) - w_c(i-1)$ and i denotes the discrete instant of time. According to [19], a typical value of Δw_c is 0.5, 1, and 1.5.

3.3 The Controller Gains Optimization by Using a GA

A genetic algorithm is a random search algorithm used to solve nonlinear equations and optimize complex problems. It employs probabilistic transition rules rather than deterministic rules and iteratively evolves a population of potential solutions known as individuals or chromosomes. Each iteration of the algorithm is referred to as a "generation". The evolution of solutions is mimicked using a fitness function and genetic operators such as selection (reproduction), crossover, and mutation [20]. Table 1 shows a pseudo-code of GA optimization processes.

The GA optimization tunes a controller gain to obtain the best gains parameters with a possible minimum objective function. The objective function is obtained using integral time-weighted absolute error (ITAE).

$$ITAE = \int_0^{t_f} t|e(t)|dt \tag{30}$$

where $e(t)$ is an error between desired and actual values and t_f is time duration. ITAE penalizes an error that persists long, resulting in more significant discrimination than IAE or ISE [19].

Table 1. Pseudo-code of the genetic algorithm

Start
Set t = 0;
Generate initial population $\boldsymbol{P(t)}$;
Compute the fitness of an individual in $\boldsymbol{P(t)}$;
do while < Stop condition for not satisfied >
Set t = t + 1;
Select from individual $\boldsymbol{P(t-1)}$ to set a tentative population $\overline{\boldsymbol{P}}(\boldsymbol{t})$;
Perform Cross-over individual in $\overline{\boldsymbol{P}}(\boldsymbol{t})$;
Perform Mutation of an individual in $\overline{\boldsymbol{P}}(\boldsymbol{t})$;
Compute the fitness of an individual in a new population $\boldsymbol{P(t)}$;
end while
Output the best individual $\boldsymbol{P(t)}$ as the best solution;
Stop

4 Results and Discussions

To investigate a controller's performance for a capability to track a given reference trajectory, adaptability, and robustness for an uncertain system model, two scenarios are considered for the reference trajectory: linear reference trajectory and nonlinear reference trajectory.

$$(x_r, y_r) = (t, 2t), (x_r, y_r) = (-5\sin(t/5), 10\sin(t/10)), \forall t \geq 0. \tag{31}$$

Linear Trajectory Tracking Performance: In linear reference trajectory tracking, a robot rapidly follows a given trajectory because the controller can easily anticipate the future behavior of a tracking line or tracking error. The robot simulation parameter configuration is as in [21]. $m_t = 120\,kg$, $L = 0.33\,m$, $d = 0.1\,m$, $r = 0.135\,m$, $I = 15.0565\,kgm^2$. A proposed controller gain parameter is optimized by the GA optimization method. The obtained controller gains are $K_x = 33.3095$, $K_y = 98.76$, $K_\theta = 10.2058$, $Kp_1 = 71.4881$, $Kp_2 = 99.8081$, $Ki_1 = 0.0441$, $Ki_2 = 0.0251$, $Kd_1 = 0.9483$ and $Kd_2 = 0.1803$.

As shown in Fig. 3, a robot tracks a given reference trajectory with a minimum tracking error at $q0 = (0, 0, 0)$ initial states in the x position, the y position, and the robot steering angle. The tracking errors in the x, y, and steering angles converge to zero in a short settling time. However, a limitation of the backstepping controller is that it produces a significant overshoot response when the robot's initial position is changed. To see the capability of the robot to track a given trajectory with a proposed controller, consider a change in the initial state and apply an unknown disturbance to the system. Hence, change the robot's initial position from (0, 0, 0) to (2, 1, pi/2) and see in Fig. 4 the change in a simulated robot system dynamics model.

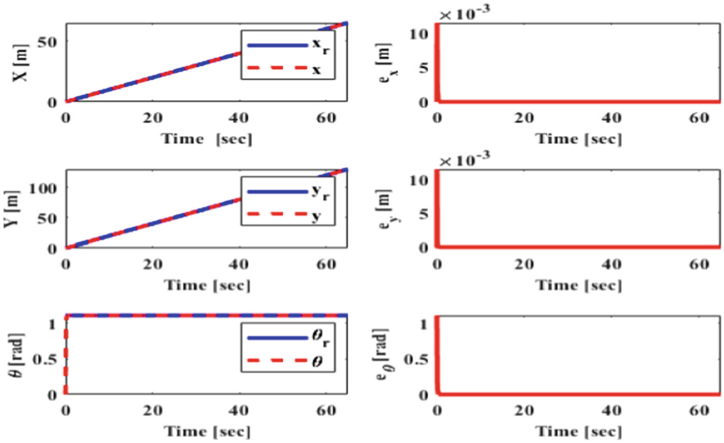

Fig. 3. The x position, y position, and robot angle tracking performance with (0, 0, 0) initial position: linear trajectory

A trajectory tracking performance is shown in Fig. 4, which clearly shows that the robot has a considerable peak value at the start of robot motion due to initial position change. Later this value decreases. The robot adapts to the initial position changes with a short settling time and a slight overshoot in reference position tracking. Thus, the proposed method's trajectory tracking ability in linear reference trajectory inputs is almost perfect, and the obtained results are smooth and robust to an initial position change.

Nonlinear Reference Trajectory Tracking Performance: This scenario applies a sinusoidal reference input to the system. The robot stabilized and followed a given path with a minimum trajectory tracking error, as shown in Fig. 5 and Fig. 6.

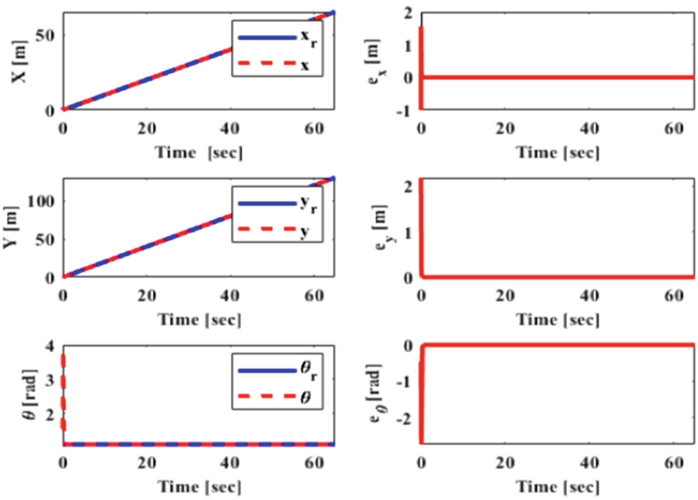

Fig. 4. The x position, y position, and steering angle tracking performance with the initial position is (2, 1, pi/2): linear trajectory

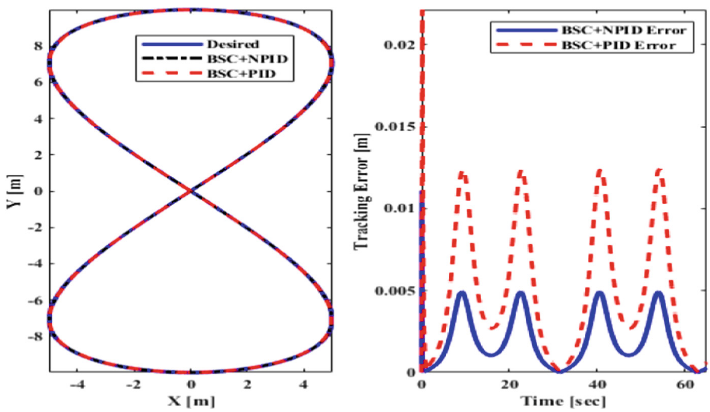

Fig. 5. Nonlinear trajectory tracking and tracking error performance with initial position is (0, 0, 0).

The robot smoothly tracks its reference trajectory in the x and y positions and a reference steering angle with the possible minimum tracking error. A proposed controller has better tracking capability than a backstepping plus PID controller, as shown in Fig. 5. A tracking performance response with (3, 2, pi/2) initial position change in Fig. 7 and Fig. 8 demonstrated that the proposed control method has a better tracking performance even if the initial position changed. The backstepping controller produced a significant overshoot when the robot's initial position changed. However, the robot quickly tracks its position after the initial position changes. The proposed controller has a better tracking performance and can quickly adapt to initial position changes than the backstepping plus PID controller.

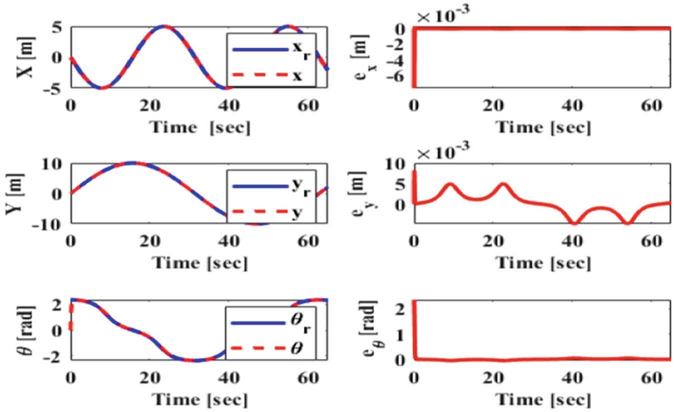

Fig. 6. The x position, y position, and robot angle tracking performance with the initial position is (0, 0, 0): nonlinear trajectory.

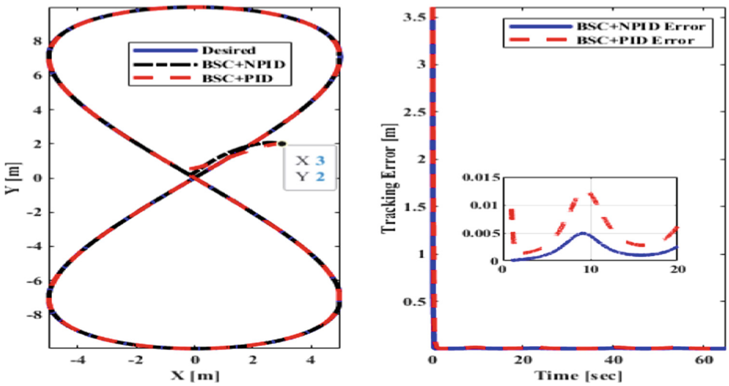

Fig. 7. Nonlinear trajectory tracking performance with initial position change to (3, 2, pi/2).

Robustness of a Control law with Unknown Disturbance: System model uncertainty may occur during selecting plant parameters for a simple representation because varying parameters and all the precise disturbance are not well known. These directly impact system performance in a real-time application. The unknown and unmodeled disturbance torque in the form of $[\tau_{d1}, \tau_{d2}] = [0.1\sin(2t), 0.1\sin(2t)]$ is applied to both wheels of the motor, and accordingly, linear trajectory and nonlinear trajectory performances are depicted in Fig. 9 and Fig. 10, respectively.

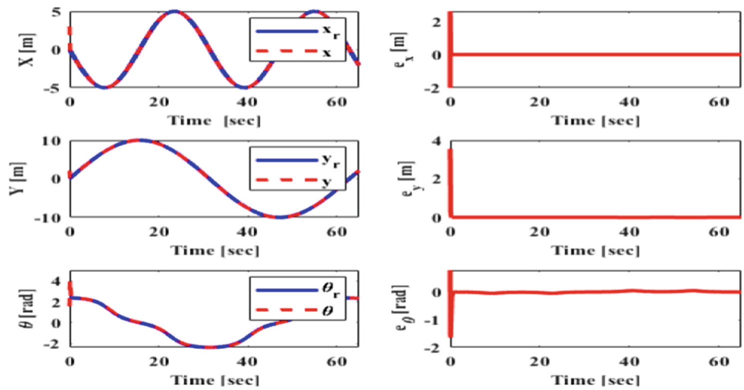

Fig. 8. The x position, y position, and steering angle tracking performance with initial position is (3, 2, pi/2): nonlinear trajectory

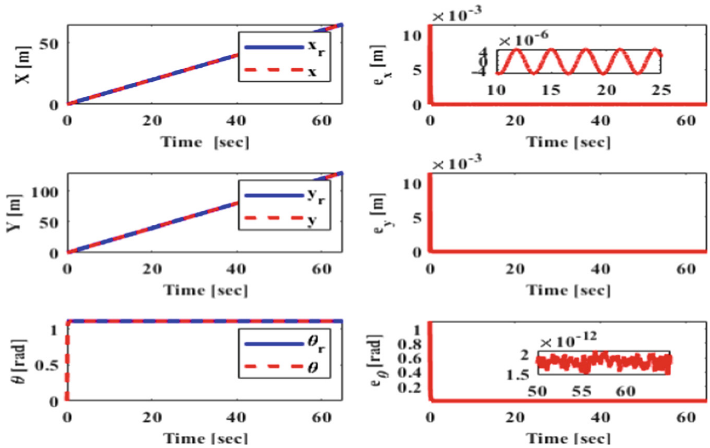

Fig. 9. Robot pose tracking performance with unknown disturbance: linear trajectory

As in Fig. 9 and Fig. 10, the control law has better unknown disturbance rejection. In linear trajectory tracking cases, the response to the unknown disturbance imposes change on a tracking response, while sinusoidal reference inputs were less affected by disturbance. This change has an almost insignificant effect on trajectory tracking errors. Therefore, the control law is smooth and gave a robust response to unknown disturbance and initial position change. The nonholonomic WMR has been controlled and stabilized by a control law in a predefined region with minimum position tracking error. The robot quickly follows a given reference trajectory, and its position tracking error converged to zero. In addition, the proposed controller outperforms the GA-based backstepping plus PID controller in terms of root-mean-square (RMS) of trajectory tracking error (47.36% in a linear and 60.32% in a nonlinear case).

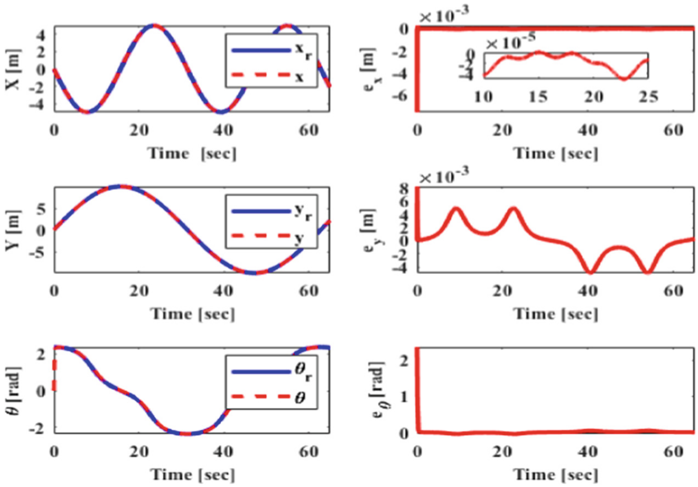

Fig. 10. Robot pose tracking performance with unknown disturbance: nonlinear trajectory

5 Conclusion

This paper presented backstepping combined with a nonlinear PID controller in the trajectory tracking control and stabilization of a two-WMR. The kinematic and dynamic modeling of a TWMR were formulated. The dynamic system models of the robot were derived using Lagrangian approaches. A backstepping plus NPID controller is designed to control a robot's trajectory tracking. The stability of a proposed controller is achieved using the Lyapunov method. The proposed controller achieved better reference trajectory tracking with a minimum tracking error in both scenarios. The robot follows its reference trajectory quickly if its initial position is changed. It also has better unknown disturbance rejection. The control law in this work does not update its parameters if the interest region (desired trajectory) is changed. A control law based on adaptive mechanisms and a self-taught controller is preferable and recommendable for better tracking capability in different environments.

References

1. Fierro, R., Lewis, F.L.: Control of a nonholonomic mobile robot using neural networks. IEEE Trans. Neural Netw. **9**(4), 589–600 (1998). https://doi.org/10.1109/72.701173
2. Uddin, N.: Trajectory tracking control system design for autonomous two-wheeled robot. JURNAL INFOTEL **10**(3), 90 (2018). https://doi.org/10.20895/infotel.v10i3.393
3. Ren, C., Ji, J.-H., Yan, H.-Y., Zhang, H., Yue, J.-Z.: A backstepping control method for mobile robot path tracking. In: Proceedings of the 3rd Annual International Conference on Mechanics and Mechanical Engineering (MME 2016), vol. 105 (2017). https://doi.org/10.2991/mme-16.2017.94
4. Chang, H., Jin, T.: Adaptive tracking controller based on the pid for mobile robot path tracking. In: Lee, J., Lee, M.C., Liu, H., Ryu, J.-H. (eds.) ICIRA 2013. LNCS (LNAI), vol. 8102, pp. 540–549. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40852-6_55

5. Moqbel Obaid, M.A., Husain, A.R., Mohammed Al-kubati, A.A.: Robust backstepping tracking control of mobile robot based on nonlinear disturbance observer. Int. J. Electr. Comput. Eng. (IJECE) **6**(2), 901 (2016). https://doi.org/10.11591/ijece.v6i2.9594
6. Xu, Q., Kan, J., Chen, S., Yan, S.: Fuzzy PID based trajectory tracking control of mobile robot and its simulation in simulink. Int. J. Control Autom. **7**(8), 233–244 (2014). https://doi.org/10.14257/ijca.2014.7.8.20
7. Fierro, R., Lewis, F.L.: Control of a nonholomic mobile robot: backstepping kinematics into dynamics. J. Robot. Syst. **14**(3), 149–163 (1997). https://doi.org/10.1002/(SICI)1097-4563(199703)14:3%3c149::AID-ROB1%3e3.3.CO;2-N
8. Fierro, R., Lewis, F.L.: Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In: Proceedings of 1995 34th IEEE Conference on Decision and Control, vol. 4, no. December, pp. 3805–3810 (1995). https://doi.org/10.1109/CDC.1995.479190
9. Hassani, I., Maalej, I., Rekik, C.: Backstepping tracking control for nonholonomic mobile robot. In: 2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), pp. 63–68 (2020). https://doi.org/10.1109/IC_ASET49463.2020.9318221
10. Dagher, K., Al-araji, A.: Design of a nonlinear PID neural trajectory tracking controller for mobile robot based on optimization algorithm. Eng. Tech J. **32**(4), 973–985 (2014)
11. Zangina, U., Buyamin, S., Abidin, M.S.Z., Mahmud, M.S.A., Hasan, H.S.: Nonlinear PID controller for trajectory tracking of a differential drive mobile robot. J. Mech. Eng. Res. Dev. **43**(7), 255–269 (2020). http://eprints.utm.my/id/eprint/90651/1/UmarZangina2020_NonLinearPIDControllerforTrajectoryTracking.pdf
12. Kalyoncu, M., Demirbaş, F.: Differential drive mobile robot trajectory tracking with using PID and kinematic based backstepping controller. Selcuk Univ. J. Eng. Sci. Technol. **5**(1), 1–15 (2017). https://doi.org/10.15317/Scitech.2017.65
13. Yousuf, B.M., Saboor Khan, A., Munir Khan, S.: Dynamic modeling and tracking for nonholonomic mobile robot using PID and backstepping. Adv. Control Appl. **3**(3), 1–12 (2021). https://doi.org/10.1002/adc2.71
14. Ben Jabeur, C., Seddik, H.: Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot. Asian J. Control **23**(1), 23–41 (2021). https://doi.org/10.1002/asjc.2356
15. Mohareri, O., Dhaouadi, R., Rad, A.B.: Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks. Neurocomputing **88**, 54–66 (2012). https://doi.org/10.1016/j.neucom.2011.06.035
16. Ahmad Abu Hatab, R.D.: Dynamic modelling of differential-drive mobile robots using Lagrange and newton-Euler methodologies: a unified framework. Adv. Robot. Autom. **02**(02) (2013). https://doi.org/10.4172/2168-9695.1000107
17. Benchouche, W., Mellah, R., Bennouna, M.S.: The Impact of the dynamic model in feedback linearization trajectory tracking of a mobile robot. Period. Polytech. Electr. Eng. Comput. Sci. **65**(4), 329–343 (2021). https://doi.org/10.3311/PPee.17127
18. .Vaidyanathan, A.T.A.S.: Backstepping Control of Nonlinear Dynamical Systems. Elsevier (2021)
19. Jin, G.-G., Son, Y.-D.: Design of a nonlinear PID controller and tuning rules for first-order plus time delay models. Stud. Inform. Control **28**(2), 157–166 (2019). https://doi.org/10.24846/v28i2y201904
20. Messom, C.: Genetic algorithms for auto-tuning mobile robot motion control. Res. Lett. Inf. Math. Sci. **3**(2002), 129–134 (2002). http://www.massey.ac.nz/~wwiims/research/letters/
21. Martins, N.A., Bertol, D.W.: Wheeled Mobile Robot Control, vol. 380. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-77912-2