PAPER • **OPEN ACCESS**

Modeling and Optimal Trajectory Tracking Control of an Autonomous Tracked Mobile Robot Using a Modified PID Controller and Backstepping

To cite this article: Abubakr Ahmed *et al* 2025 *J. Phys.: Conf. Ser.* **3058** 012005

Journal of Physics: Conference Series **3058** (2025) 012005 doi:10.1088/1742-6596/3058/1/012005

Modeling and Optimal Trajectory Tracking Control of an Autonomous Tracked Mobile Robot Using a Modified PID Controller and Backstepping

Abubakr Ahmed[1], **Mohamed A. Kamel**[1], **Ehab Said**[1] **and Amr Roshdy**[2]
[1]Department of Mechatronics Engineering, Military Technical College, Cairo, Egypt
[2]Department of Mechanical Engineering, Cairo, Egypt

E-mail: mohamed.atef.kamel@ mtc.edu.eg

Abstract. Tracked Mobile Robots (TMRs) are extensively utilized in applications that demand high mobility across difficult terrains, such as military reconnaissance and search and rescue operations. The trajectory tracking problem for autonomous TMRs has garnered considerable attention and has emerged as a focal point of contemporary research. In this paper, a control strategy is developed using an integration of the kinematic backstepping approach with a modified PID controller. Given that the performance of the proposed controller is highly dependent on the appropriate selection of gain parameters, determining their optimal values is of paramount importance. Consequently, the Particle Swarm Optimization (PSO) algorithm is employed to optimize the parameters of both the PID and backstepping controllers. The mathematical model includes a comprehensive kinematic and dynamic model while considering the effects of external disturbances and slippage, which increases the accuracy of the proposed controller. Finally, simulation results demonstrated the effectiveness of the proposed controller by allowing the tracked robot to precisely follow the desired trajectory.
Keywords: Tracked mobile robot, modified PID controller, tracked vehicle, trajectory tracking control, unmanned ground vehicle

1. Introduction

The rapid advancements in automation, robotics, and communication technologies have significantly increased interest in unmanned and autonomous systems across academic, civilian, and military domains. These systems are increasingly recognized as viable solutions across a broad spectrum of applications [1]. Unmanned systems are programmable, multifunctional machines designed to plan and execute tasks in their environment without human assistance. They can also collect and analyze information from their surroundings using sensory devices and control units [2]. Unmanned systems are categorized according to their operating environment into three main types: unmanned underwater vehicles (UUVs), unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs).

UGVs are used in a variety of applications which are repetitive tasks or high-risk tasks, where having a human operator present could be uncomfortable, hazardous, or impossible [3]. UGVs are used in civilian areas such as forestry and agriculture [4], environmental monitoring [5], mining purposes [6], and assistance purposes [7]. Besides the civilization and industrial applications, UGVs can be deployed for military and tactical operations, including reconnaissance and surveillance [8], search and exploration [9], explosives and landmine

remediation [10], and dangerous tasks such as firefighting, remediating nuclear pollution, and in some places which might contain serious radioactivity or toxic gases [11].

UGVs can be categorized into three distinct classes according to their locomotion mechanisms: wheeled mobile robots (WMRs), tracked mobile robots (TMRs), and legged mobile robots (LMRs)[12]. Attributable to their inherent mechanical simplicity and relatively low energy consumption, WMRs are well-suited for conventional applications. Conversely, LMRs demonstrate superior adaptability in navigating highly irregular, complex, and obstacle-laden terrains, including staircases and rubble-strewn environments[12]. TMRs are well-suited for applications requiring robust locomotion in challenging terrains, including loose soil, sand, snow, and rocky environments. Among off-road ground robots, tracked robots are preferred due to their superior tractive force compared to legged and wheeled robots, as well as their reduced ground pressure, improved weight distribution, structural robustness, stability, and agile maneuverability. These characteristics enable TMRs to traverse a variety of terrains, including mud, runoff slopes, snow, and loose sand [13]. Consequently, TMRs constitute a viable and effective solution for off-road mobility in complex environments.

Due to the nonholonomic, multivariable, complex, and nonlinear nature of the dynamic model of TMRs, achieving precise and effective motion control remains a considerable challenge. The robot slip is also a crucial factor in control, as tracked robots rely on the skid-steer mechanism for steering, neglecting this factor may result in a significant deviation of the robot from its intended track due to the continuous accumulation of errors throughout its motion. Additionally, longitudinal and lateral resistance forces must be accounted for when modeling the robot's dynamics. Therefore, to achieve an autonomous system, the primary necessity is to create an accurate model and choose an appropriate control algorithm.

Differential Drive Wheeled Mobile Robots (DDWMRs) and TMRs utilize similar motion control strategies to govern their movement. However, most existing research has primarily focused on motor-driven control for DDWMRs. In contrast, TMR motion is significantly influenced by track-terrain interactions, load distribution, and friction, requiring a dynamic model-based control approach to effectively account for these factors[14]. The DARPA Grand Challenges (DGC) series in 2004 and 2005 served as a crucial catalyst for significant advancements in the research and development of control algorithms for autonomous vehicles [15, 16]. The inherent control challenges of Unmanned Ground Vehicles (UGVs) have led to the development of a diverse array of control algorithms, including proportional-integral-derivative (PID) control, Lyapunov-based methods, sliding mode control, dynamic feedback linearization, model predictive control, etc. Ahmadi et al. [17], A trajectory tracking control algorithm for a surface drilling machine is proposed to improve tracking accuracy. To reduce friction forces, a feedforward compensator is incorporated into the feedback linearization-based control framework. Additionally, the proposed method regulates the trajectory's forward velocity and lateral offset error while accounting for track-soil interactions. In [18], two control systems are proposed: PID control and fuzzy logic control (FLC) to assess the dynamic performance of tracked robots under slippage and skidding conditions. In [19], a combination of the modified PID computed torque control and a kinematic backstepping method is presented. The suggested controller has considered nonholonomic constraints, skid steering, rough terrain, and vehicle dynamics. Reference [14] proposes a hybrid control strategy that integrates sliding mode control (SMC) and backstepping to design a trajectory tracking controller for tracked vehicles. However, the performance of the controller is significantly influenced by the design of the sliding surface. A surface with inadequate design may respond slowly or become unstable. In [20], a model predictive control (MPC) approach is presented, utilizing a kinematic model for trajectory tracking in tracked vehicles. Model predictive control demonstrates strong potential as an effective approach for managing nonlinear dynamic systems. However, its implementation requires significant computational resources due to its high complexity.

Although advanced algorithms often result in reliable and accurate tracking performance, their efficacy significantly depends on the processing capacity of the microcontroller being used. Consequently, executing advanced control strategies presents considerable computational challenges for tracked mobile robots, as their nonlinear characteristics influence the system's response time. Complex control strategies require a lot of computational strength, which low-capacity processing units might not be capable of accommodating. Considering these constraints, designing a control architecture that optimally balances computational efficiency with the ability to accurately handle system nonlinearities is essential. To overcome these challenges, this study introduces a modified PID controller that preserves structural simplicity while enhancing control performance. The proposed method effectively manages the nonlinear dynamics of the tracked robot while reducing computational complexity, ensuring competitive trajectory tracking performance without requiring high-performance sensors. To optimize controller parameters, researchers have explored various non-traditional techniques, such as particle swarm optimization (PSO), genetic algorithm (GA), and ant colony algorithm (ACO), for tuning control gains[21]. In this study, PSO is utilized to optimize the PID controller parameters, thereby enhancing tracking performance.

The paper is structured as follows: Section 2 provides a comprehensive analysis of the tracked robot, including the mathematical modeling of its kinematics and dynamics. Section 3 details the control architecture of the enhanced PID controller and explains the implementation of the Particle Swarm Optimization (PSO) algorithm for gain tuning. Section 4 describes the environmental configuration employed in the simulation experiments. Finally, conclusions and future work recommendations are outlined in Section 5.

2. Tracked Vehicle Model Construction

The mathematical modeling of the tracked mobile robot, including its kinematic and dynamic behaviors, is presented in this section. This modeling forms the basis for the application of torque-based PID control. Figure 1 depicts the global frame-G and the local frame-B as the designated reference frames. Initially, the origin (O) of the ground-fixed global reference frame-G is aligned with the robot's center of gravity (CG). In contrast, the local reference frame-B is attached to the robot, with its origin continuously coinciding with the CG throughout the motion.

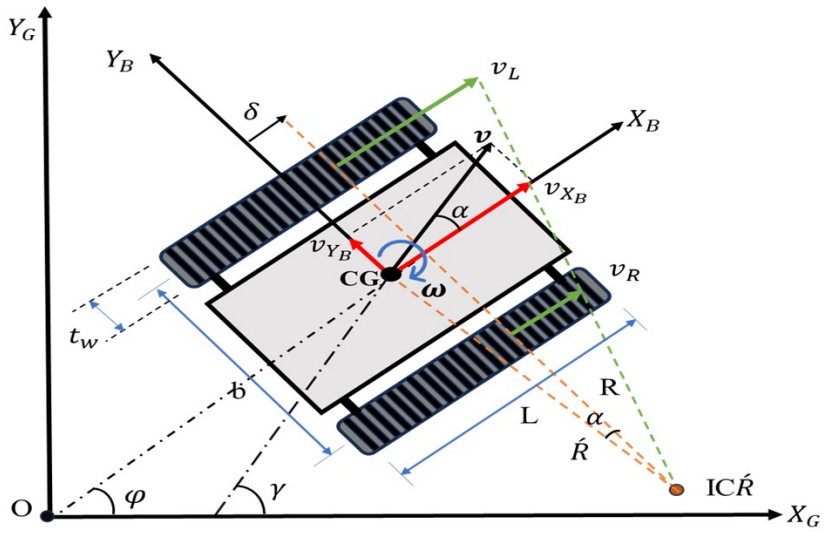


Figure 1: Tracked robot kinematic modeling.

2.1. ***Kinematic Model***

Let $q \in \mathbb{C}$ be the vector that represents the location of the robot, where C is the robot's configuration space. A tracked robot making a right turn around its instantaneous center of rotation (ICR') on flat terrain is seen in figure 1. During this motion, the robot deviates from its reference trajectory due to a slip angle α. As a result of track side slip, the instantaneous center of rotation (ICR') shifts by a displacement α along the Y_B-axis. The angle between the X_B and X_G is referred to as the robot's orientation angle ϕ, while the directional course angle γ is defined as $[\gamma = \phi + \alpha]$. When turning, the linear velocities of both tracks can be calculated as follows, considering the slip ratio:

$$v_R = r\omega_R(1-\sigma_R), v_L = r\omega_L(1-\sigma_L) \tag{1}$$

Here, r denotes the radius of the sprocket, while v_R and v_L represent the forward velocities of both tracks (right and left) of the robot. The lateral slip of the right and left tracks is represented by σ_R and σ_L. Likewise, the angular velocities of both track sprockets (right and left) are denoted by ω_R and ω_L. They are determined as described below:

$$\omega_R = (\frac{v_R}{r}) \quad , \quad \omega_L = (\frac{v_L}{r}) \tag{2}$$

The following equation describes the velocity vector of the robot v in the B-frame:

$$[\overrightarrow{\mathbf{v}}]_B = \begin{bmatrix} v_{x_B} \\ v_{y_B} \end{bmatrix} = \begin{bmatrix} \|v_B\| \cos\alpha \\ \|v_B\| \sin\alpha \end{bmatrix} \tag{3}$$

The linear and angular velocities can alternatively be expressed as follows:

$$\|v_B\| = \frac{v_L + v_R}{2\cos\alpha} = \frac{r\left[\omega_L(1-\sigma_L) + \omega_R(1-\sigma_R)\right]}{2\cos\alpha} \tag{4}$$

$$|\omega| = \frac{v_L - v_R}{b} = \frac{r\left[\omega_L(1-\sigma_L) - \omega_R(1-\sigma_R)\right]}{b} \tag{5}$$

To determine the turning radius R' using equation (4) and equation (5), as follows:

$$R' = \frac{\|v_B\|}{|\omega|} = \frac{b}{2\cos\alpha}\frac{\left[\omega_L(1-\sigma_L) + \omega_R(1-\sigma_R)\right]}{\left[\omega_L(1-\sigma_L) - \omega_R(1-\sigma_R)\right]} \tag{6}$$

where b is the track gauge. Using the rotation matrix Q to transform the robot velocity $[\overrightarrow{\mathbf{v}}]_B$ into the G-frame:

$$\begin{aligned} \mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = [Q][\overrightarrow{\mathbf{v}}]_B = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} \|v_B\|\cos(\alpha) \\ \|v_B\|\sin(\alpha) \end{bmatrix} \\ = \frac{r}{2}\left[(1-\sigma_L)\omega_L + (1-\sigma_R)\omega_R\right] \begin{bmatrix} \cos(\varphi) - \sin(\varphi)\tan(\alpha) \\ \sin(\varphi) + \cos(\varphi)\tan(\alpha) \end{bmatrix} \end{aligned} \tag{7}$$

Thus, the tracked mobile robot's kinematic model in G-frame can be expressed this way:

$$\dot{x} = \frac{r}{2}\left[(1-\sigma_L)\omega_L + (1-\sigma_R)\omega_R\right](\cos(\varphi) - \sin(\varphi)\tan(\alpha)) \tag{8}$$

$$\dot{y} = \frac{r}{2}\left[(1-\sigma_L)\omega_L + (1-\sigma_R)\omega_R\right](\sin(\varphi) + \cos(\varphi)\tan(\alpha)) \tag{9}$$

$$\dot{\varphi} = \frac{r}{b}\left[(1-\sigma_L)\omega_L - (1-\sigma_R)\omega_R\right] \tag{10}$$

To ensure dimensional consistency in the dynamic model, the generalized coordinates, which exhibit dimensional non-homogeneity, are initially defined as $q = [x, y, \phi]^T$. The radius of gyration is given by $R_g = \sqrt{\frac{I_Z}{m}}$, where I_Z and m represent the robot's moment of inertia and mass about its center of gravity (CG). This principle is applied to address the problem, as proposed in [19]. Consequently, the generalized coordinates are reformulated as $q(t) = [x, y, R_g\varphi]^T$ to achieve dimensional consistency. The constraint on the robot's lateral motion necessitates the enforcement of a nonholonomic condition, which is mathematically expressed in Pfaffian form [19] as follows:

$$-\dot{x}\sin(\varphi) + \dot{y}\cos(\varphi) + \dot{\varphi}\delta = A^T(q)\dot{q} = 0 \tag{11}$$

Here

$$A(q) = \begin{bmatrix} -\sin(\varphi) \\ \cos(\varphi) \\ \delta/R_g \end{bmatrix} \quad , \quad \dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ R_g\dot{\varphi} \end{bmatrix} \tag{12}$$

Now, a full-rank matrix $S(q) \in \mathbb{R}^{3\times 2}$ is defined, with columns orthogonal to $A(q)$, as presented below:

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

Thus, $S(q)$ is expressed in the following way:

$$S = \begin{bmatrix} \cos(\varphi) & \left(\frac{\delta}{R_g}\right)\sin(\varphi) \\ \sin(\varphi) & -\left(\frac{\delta}{R_g}\right)\cos(\varphi) \\ 0 & 1 \end{bmatrix} \tag{14}$$

As indicated by (Fierro et al.) [22], based on equations (13) and (14), the velocity $V(t) = [v, \dot{\varphi}]^T \in \mathbb{R}^2$ is determined in such a way that:

$$\dot{q} = S(q)V(t) \tag{15}$$

2.2. ***Dynamic Model***

To predict the robot's dynamic behavior, the dynamic model defines the relationships between forces, torques, and component motion. Tractive forces characterize the magnitude of effort necessary to drive the robot along the intended trajectory. Achieving forward motion necessitates overcoming the combined effects of friction, shear resistance, and inertia induced by track-terrain interactions. As illustrated in Figure 2, the tractive forces are represented by F_l, F_r, while the longitudinal resistance forces are represented by R_l and R_r. Additionally, the lateral resistance force resulting from the lateral distribution of soil shear is denoted by F_y.

The tracked robot's dynamic model is formulated in the body-fixed frame B as follows:

$$m\ddot{x}_B = F_r + F_l - R_r - R_l \tag{16}$$

$$m\ddot{y}_B = F_y \tag{17}$$

$$I_z\ddot{\varphi} = M - M_r \tag{18}$$

Where m represents the robot's mass, M denotes the turning moment, and M_r corresponds to the resisting moment opposing the turn. Assuming an equal weight distribution across both tracks, the longitudinal resistance forces R_l and R_r, acting on the two tracks, are expressed as follows[19]:

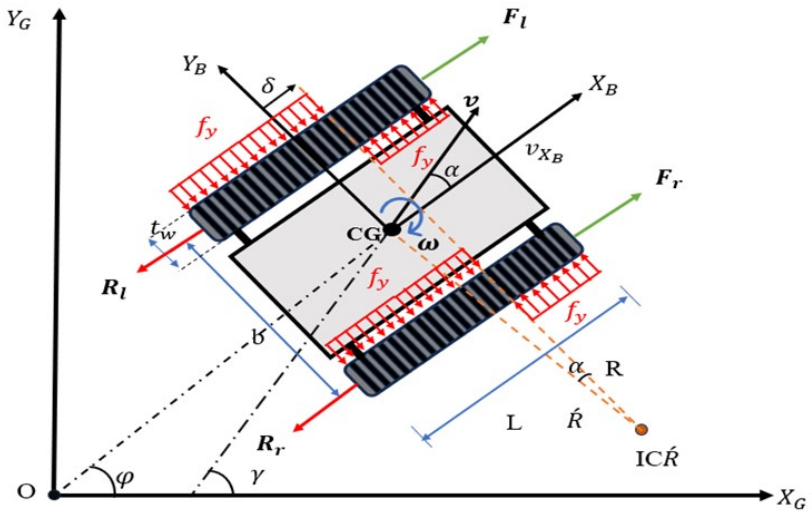


Figure 2: Tracked Robot dynamic modeling.

$$R_l = R_r = \mu_l \left(\frac{W}{2}\right) \tag{19}$$

This equation calculates the total lateral friction force[19]:

$$F_y = 2\,\mathrm{sgn}(\omega)\mu_t \delta \left(\frac{W}{L}\right) \tag{20}$$

Where μ_l and μ_t denote the tracked robot's resistance coefficients in the longitudinal and lateral directions. Due to track sideslip, the instantaneous center of rotation shifts by a distance δ from the Y_B-axis, which is calculated using the following equation [14]:

$$\delta = \frac{La_y}{2\mu_t g}\cos\alpha \tag{21}$$

Where a_y denotes the centrifugal acceleration, given by ($a_y = \frac{v^2}{R}\sin\alpha$), sign() denotes the signum function, which is defined as follows [14]:

$$\mathrm{sgn}(\varepsilon) = \begin{bmatrix} +1, & \text{if } \varepsilon > 0 \\ 0, & \text{if } \varepsilon = 0 \\ -1, & \text{if } \varepsilon < 0 \end{bmatrix} \tag{22}$$

The turning moment M can be calculated using the longitudinal resistance forces and tractive forces, as shown below:

$$M = \frac{b}{2}[(F_l + R_r) - (F_r + R_l)] = \frac{b}{2}[F_l - F_r] \tag{23}$$

The moment of resistance to turning is formulated as[14]:

$$M_r = 2\,\mathrm{sgn}(\omega)\mu_t \frac{W}{L}\left(\delta^2 - \frac{L^2}{4}\right) \tag{24}$$

The dynamic model in the G frame can be obtained by multiplying Newton-Euler dynamic equations of motion equation (16-18) by the rotation matrix, resulting in:

$$m\ddot{x} = (F_l + F_r)\cos(\varphi) - (R_l + R_r)\cos(\varphi) - F_y \sin(\varphi) \tag{25}$$

$$m\ddot{y} = (F_l + F_r)\sin(\varphi) - (R_l + R_r)\sin(\varphi) + F_y \cos(\varphi) \tag{26}$$

$$\frac{I_z R_g \ddot{\varphi}}{R_g^2} = (F_l - F_r)\frac{b}{2R_g} - \frac{M_r}{R_g} \tag{27}$$

The dynamic model of the tracked robot can be reorganized and integrated with the nonholonomic constraint from equation (11) to be reformulated within the generalized Euler–Lagrange framework [23]:

$$M(q)\ddot{q} + f(\dot{q}) + A(q)\lambda(q) = B(q)\tau \tag{28}$$

where

$$\ddot{q} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ R_g\ddot{\varphi} \end{bmatrix}, M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z/R_g^2 \end{bmatrix}, B = \begin{bmatrix} \cos(\varphi) & \cos(\varphi) \\ \sin(\varphi) & \sin(\varphi) \\ b/2R_g & -b/2R_g \end{bmatrix}, \quad A = \begin{bmatrix} -\sin(\varphi) \\ \cos(\varphi) \\ \delta/R_g \end{bmatrix}, \quad \tau = \begin{bmatrix} F_l \\ F_r \end{bmatrix},$$

$$f = \begin{bmatrix} (R_l + R_r)\cos(\varphi) + F_y \sin(\varphi) \\ (R_l + R_r)\sin(\varphi) - F_y \cos(\varphi) \\ M_r/R_g \end{bmatrix}$$

The mass inertia matrix is denoted by $M(q)$, the friction vector by $f(q)$, the input matrix is represented by $B(q)$, the torque input is denoted by τ, $A(q)$ denotes the matrix linked to the nonholonomic constraints and the Lagrange multiplier vector by λ. Differentiating both sides of equation (15) results in the following expression:

$$\ddot{q} = \dot{S}(q)V(t) + S(q)\dot{V}(t) \tag{29}$$

substituting equation (29) in equation (28) and multiplying both sides by S^T, The complete dynamic model of the robot is reformulated as follows:

$$\bar{M}(q)\dot{V}(t) + \bar{C}(\dot{q}, q)V(t) + \bar{f}(q) = \bar{B}(q)\tau \tag{30}$$

where

$$\bar{M}(q) = S^T M S = m\begin{bmatrix} 1 & 0 \\ 0 & \delta^2/R_g^2 + 1 \end{bmatrix}, \quad \bar{C}(\dot{q}, q) = S^T M \dot{S} = \begin{bmatrix} 1 & m\delta\dot{\varphi}/R_g \\ -m\delta\dot{\varphi}/R_g & m\delta\dot{\delta}/R_g^2 \end{bmatrix},$$

$$\bar{f}(q) = S^T f = \begin{bmatrix} 2R_l \\ \operatorname{sgn}(\omega)\mu_t \frac{w}{R_g L}\left(3\delta^2 - \frac{L^2}{4}\right) \end{bmatrix}, \quad \text{and} \quad \bar{B}(q) = S^T B = \begin{bmatrix} 1 & 1 \\ \frac{b}{2R_g} & -\frac{b}{2R_g} \end{bmatrix}$$

3. Controller Design

The proposed control strategy addresses the trajectory tracking problem, which is shown in Figure 3. Details of the proposed controller are presented as follows:

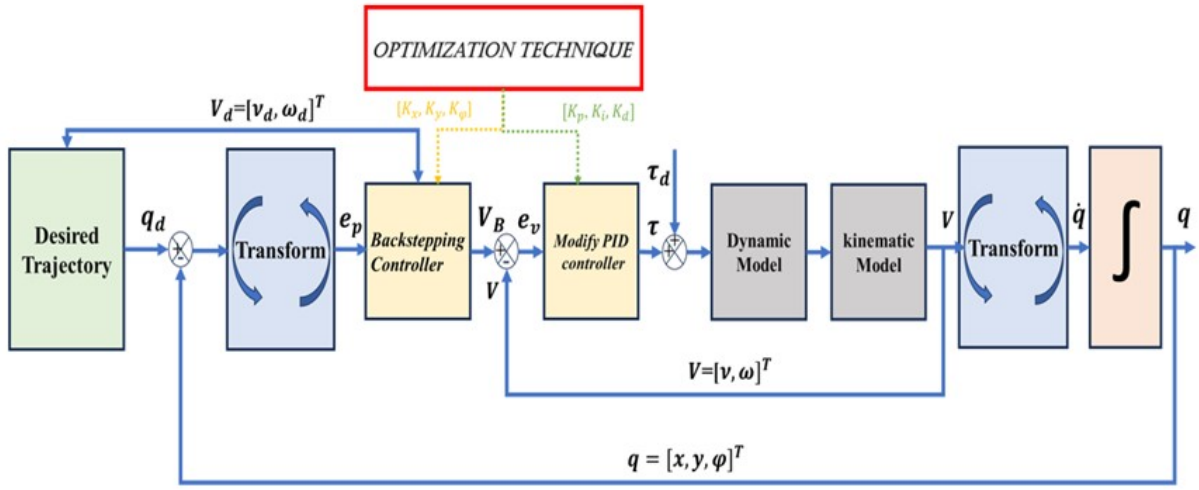


Figure 3: A tracked robot's trajectory tracking control system

3.1. ***Trajectory Tracking Control of TMR***

Let the desired position vector be defined as $q_d(t) = [x_d(t), y_d(t), \varphi_d(t)]^T$ over the time interval $t \in [0,\mathrm{T}]$, and let $q(t) = [x(t), y(t), \varphi(t)]^T$ denote the true position vector derived from sensor measurements. The primary objective of trajectory tracking is to minimize the deviation between the robot's actual posture and the predefined desired trajectory. The desired trajectory $q_d(t)$ is formulated as follows: :

$$\dot{q}_d = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\varphi}_d R_g \end{bmatrix} = \begin{bmatrix} \cos\varphi_d & 0 \\ \sin\varphi_d & 0 \\ 0 & R_g \end{bmatrix} \begin{bmatrix} v_d \\ \omega_d \end{bmatrix} \tag{31}$$

Here, (v_d) and (ω_d) denote the desired linear and angular velocities, are computed as follows:

$$v_d = \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2} \tag{32}$$

$$\omega_d = \pm\frac{\dot{x}_d\ddot{y}_d - \dot{y}_d\ddot{x}_d}{\dot{x}_d^2 + \dot{y}_d^2} \tag{33}$$

Here, the sign $(\pm)$ denotes the direction of the robot's movement, where $(+)$ indicates forward motion and $(-)$ represents backward motion. The desired orientation φ_d is defined as:

$$\varphi_d = \arctan2(\dot{y}_d, \dot{x}_d) \tag{34}$$

The pose tracking error, denoted as e_p, is the deviation between the desired trajectory q_d and the actual pose q, expressed as follows:

$$e_p = q_d - q \tag{35}$$

The error e is then converted from the frame-G to the frame-B. The error model may be expressed as follows:

$$\begin{bmatrix} e_x \\ e_y \\ e_\varphi \end{bmatrix}_B = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ (\varphi_d - \varphi)(R_g) \end{bmatrix} \tag{36}$$

where the forward, lateral, and angular errors are denoted by e_x, e_y,e_φ, respectively. The trajectory tracking error rate is obtained by differentiating equation (36) as follows:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\varphi \end{bmatrix}_B = \begin{bmatrix} -\cos\alpha \\ -\sin\alpha \\ 0 \end{bmatrix} v + \begin{bmatrix} e_y \\ -e_x \\ -R_g \end{bmatrix} \omega + \begin{bmatrix} v_d \cos(e_\varphi / R_g) \\ v_d \sin(e_\varphi / R_g) \\ R_g \omega_d \end{bmatrix} \tag{37}$$

3.2. ***Kinematic backstepping controller***

The backstepping controller is designed to maintain the desired velocity $V_d = [v_d, \omega_d]^T$, while reducing the pose deviation (e_p). To achieve this, the required velocity vector V_B , which serves as the input to the controller, must be determined. According to the kinematic model, V_B is given as follows [24]:

$$V_B = \begin{bmatrix} v_d \cos\left(\frac{e_\varphi}{R_g}\right) + K_x e_x \\ \omega_d + v_d \left[K_y e_y + K_\varphi \sin\left(\frac{e_\varphi}{R_g}\right)\right] \end{bmatrix} \tag{38}$$

where the terms K_x, K_y, and K_φ are controller gain parameters (positive constant). Taking the first derivative of the equation (38), and assuming that the desired velocity vector V_d remains constant, it is obtained as:

$$\dot{V}_B = \begin{bmatrix} K_x & 0 & -v_d \sin(e_\varphi / R_g)/R_g \\ 0 & K_y v_d & K_\varphi \cos(e_\varphi / R_g)/R_g \end{bmatrix} \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\varphi \end{bmatrix} \tag{39}$$

Consequently, the velocity tracking error (e_v) is the variation between the velocity control input vector V_B and the true velocity vector V, which is obtained as follows.

$$e_v = V_B - V \tag{40}$$

3.3. ***Modified PID controller***

The proposed modified PID controller aims to eliminate velocity tracking errors. Therefore, the controller must satisfy equation (40):

$$\lim_{n \to \infty} e_v = V_B - V = 0$$

The following analysis presents the derivation of the proposed modified PID controller, formulated using the computed torque methodology [19]:

$$\bar{M}(q)(\dot{V}_B + K_p e_v) + \bar{C}(\dot{q}, q) V_B + \bar{f}(q) + (K_p e_v + K_i \int e_v dt + K_d \dot{e}_v) = \bar{B}(q)\tau \tag{41}$$

Where K_p, K_i and K_d represent the PID parameters. Substituting equation (41) in equation (30), we obtain:

$$(\bar{M} + K_d)\dot{e}_v + (\bar{C} + \bar{M} K_p + K_p) e_v + K_i \int e_v dt = 0 \tag{42}$$

Equation (42) describes the dynamic behavior of the velocity tracking error within the control system.

3.4. ***Optimization Method***

Since the performance of the proposed controller is highly dependent on the appropriate selection of gain parameters, determining their optimal values is crucial. Therefore, the PSO algorithm is utilized to systematically optimize the PID parameters.

In the PSO process, each agent evaluates its fitness level, determining its next position and movement velocity. The best fitness value of the particle will be selected as the representative solution for the optimization problem. The globally best solution identified among all particles and the locally best solution obtained across iterations is two key parameters that influence a particle's movement. These factors determine both the direction and magnitude of the particle's velocity. Initially, the PSO algorithm establishes the position X_i for each particle as: $[X_i = X_{i1}, X_{i2}, X_{i3}, \ldots X_{iN}]$, the previous solutions are retained for each number of particles (i_th) $[P_i = P_{i1}, P_{i2}, P_{i3}, \ldots .P_{iN}]$. and the velocity for each dimension, Vi $[V_i = V_{i1}, V_{i2}, V_{i3}, ...V_{iN}]$.

Throughout the optimization process, the velocity is updated iteratively, facilitating the movement of particles toward their respective optimal positions, referred to as the personal-best and the global-best. The mathematical formulation for updating the final velocity is expressed as follows [25]:

$$V_i^{k+1} = V_i^k \times w + (P_{\text{best}i} - X_i^k) \times c_p \times R_1 + (G_{\text{best}} - X_i^k) \times c_g \times R_2 \tag{43}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{44}$$

The parameter w denotes the inertia weight, which controls the influence of each particle's previous velocity on its current movement. The instantaneous position of particle i at iteration k is denoted as X_i^k, while its corresponding velocity at the same iteration, V_i^k, dictates its movement. The coefficients c_p and c_g as weighting factors that regulate the influence of the local-best and global-best solutions. Furthermore, R_1 and R_2 are randomly distributed variables with uniform distributions in the interval [0–1].

In PSO, the cost function is defined based on two key parameters: the mean square error (MSE) of position tracking, expressed as $e_p = square(e_x^2 + e_y^2)$, and the angle tracking error, denoted as e_φ. The optimization problem is defined such that the objective function aims to minimize the overall tracking error ((e)).To reduce the error, the cost function is expressed as :

$$\min_{\text{cost}} J_e = \frac{1}{n} \sum_{T=1}^{n} \left[(e_p)^2 + (e_\varphi)^2 \right] \tag{45}$$

$$\text{Subject to:} \quad \mathbf{X} = [K_p, K_i, K_d, K_x, K_y, K_\varphi] > 0 \tag{46}$$

Here, n denote the total number of samples,X represents the vector of design variables and T denotes the time step.

4. Simulation results

Simulations were performed to assess the proposed control strategy and validate its effectiveness. The parameters of the controller implemented in the simulations and those of the tracked robot are provided in Tables 1 and 2, respectively. The tracked robot is required to follow a circular trajectory, defined by:

$$x_d = 4\cos\left(\frac{t}{4}\right), \quad y_d = 4\sin\left(\frac{t}{4}\right) \tag{47}$$

The robot's initial position is set to [3, 0, 0], with the desired linear and angular velocities defined as $v_d = 1$ (m/s) and $\omega_d = 0.25$ (rad/s). During the simulation, the vehicle tracks are

subjected to an external disturbance force, τ_d, applied over a time interval of 10 to 15 seconds. The disturbance force is expressed as follows:

$$\tau_d = 500 \sin(3t) \tag{48}$$

Table 1: Robot specifications.

m (kg)	I_Z (kg.m^2)	r (m)	L (m)	t_w (m)	b (m)	μ_l	α (rad)	μ_t
30	3.29	0.04	0.42	0.075	0.4	0.6	0.1	0.9

Table 2: Controllers gains

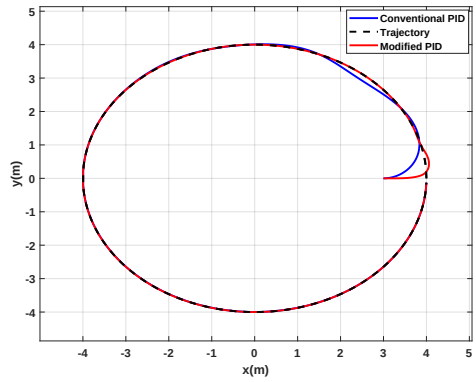
k_x	k_y	k_z	k_p	k_i	k_d
12.59	28.11	1.54	26.64	19.38	8.12

As presented in Table 3, the Modified PID controller demonstrates superior performance compared to the conventional PID, particularly in the presence of external disturbances. While the conventional PID becomes unstable as disturbance forces increase, the Modified PID maintains consistent performance. This enhanced stability results from the modified PID's incorporation of the system's nonlinear dynamics.

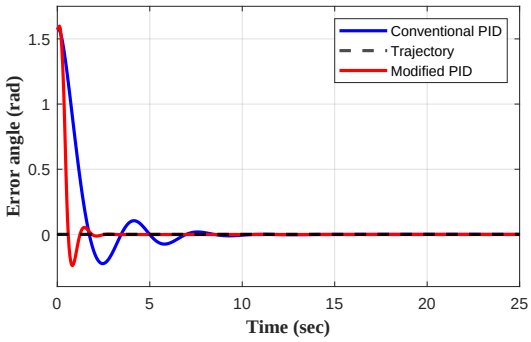
The results of the simulation show that the suggested approach successfully solves the trajectory tracking issue, as illustrated in Figure 4. Moreover, as shown in Figure 7, it compensates for model uncertainties and mitigates the effects of external disturbances. Figure 4(a) demonstrates the effectiveness of the proposed controller by illustrating the trajectory-tracking capability of the tracked robot. The solid red line represents the actual trajectory, which closely follows the desired trajectory depicted by the dashed black line. Figure 4(b, c, and d) illustrate the deviations in pose while tracking the reference coordinates. The robot's capability to precisely follow the desired trajectory is demonstrated by the variations in x, y, and orientation φ over time. The robot's linear and angular velocities are depicted in Figure 5. Figures 6 (a) and (b) present the velocity response under disturbance conditions. Figure 7 depicts the control force responses over time, where the applied forces on the left and right tracks (F_L in blue and F_R in red) effectively compensate for model uncertainties and mitigate the impact of external disturbances, ensuring stable system performance.

Table 3: Comparison between conventional and modified PID controllers

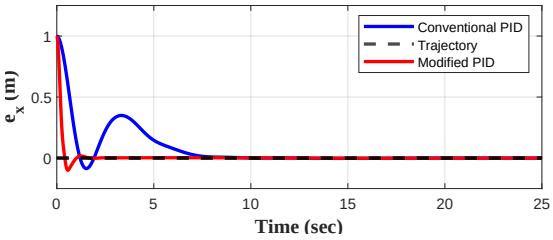
Controller type	Modified PID			Conventional PID		
disturbance	$MSE(e_x)$	$MSE(e_y)$	$MSE(e_\vartheta)$	$MSE(e_x)$	$MSE(e_y)$	$MSE(e_\vartheta)$
$\tau_d = 0$ N	0.0057	0.0008	0.0349	0.0366	0.0115	0.0637
$\tau_d = 250$ N	0.0061	0.0012	0.0350		unstable	
$\tau_d = 500$ N	0.0072	0.0023	0.0352		unstable	
$\tau_d = 750$ N	0.0090	0.0042	0.0355		unstable	



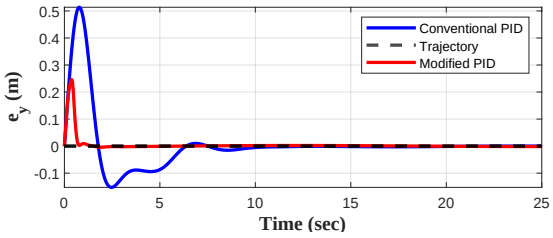
(a) Robot following a desired circular track.



(b) Pose deviation e_φ

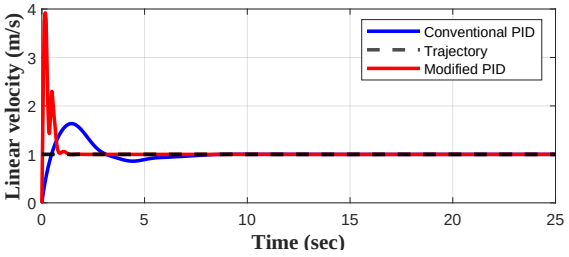


(c) Pose deviation e_x

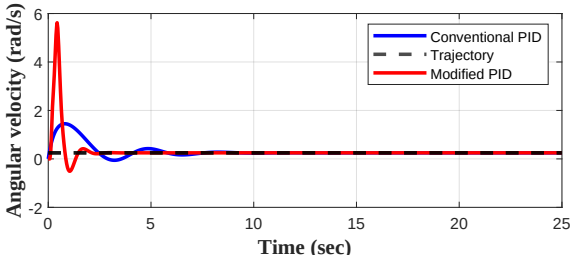


(d) Pose deviation e_y

Figure 4: Simulation results for circular trajectory tracking.



(a)

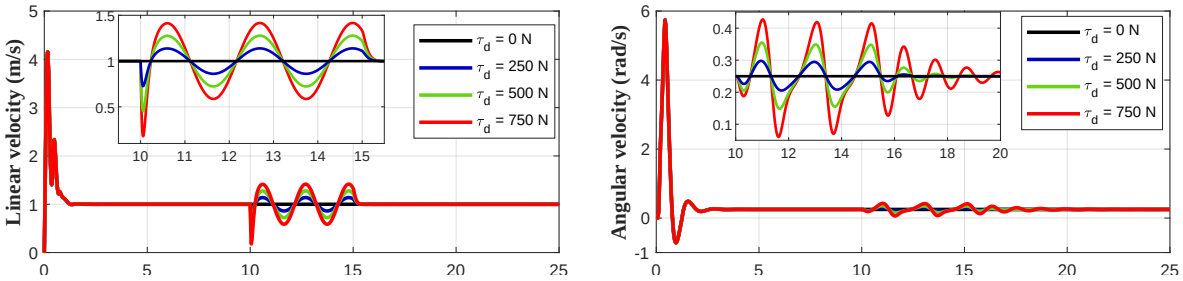


(b)

Figure 5: The desired linear and angular velocities of the tracked robot:(a) linear velocity,(b) angular velocity

Conclusion

This study introduced an enhanced model-based PID controller integrated with kinematic backstepping to ensure accurate trajectory tracking for tracked mobile robots. The proposed controller provides integration of the system dynamics, taking into account the effects of external disturbances and slip. Additionally, PSO was utilized to optimize control gains, further enhancing control performance. To evaluate the effectiveness of the proposed controller, a comparative analysis was conducted against the conventional controller, using MSE metric and accounting for the effects of external disturbances. Simulation results demonstrate that the proposed controller exhibits high adaptability to real-world conditions and achieves precise motion control.



(a) (b)

Figure 6: The linear and angular velocity responses of the tracked robot, subjected to external disturbance forces ranging from 250 to 750 N, are presented in subfigures (a) and (b), respectively

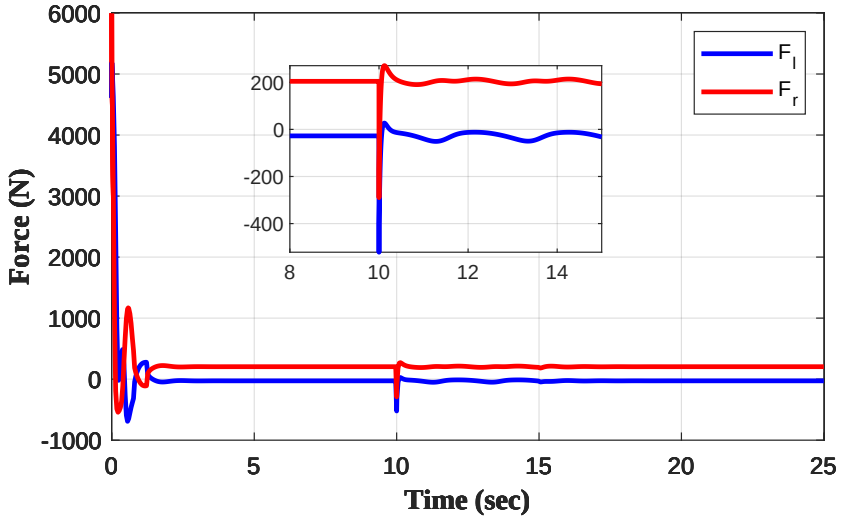


Figure 7: The forces exerted on the left and right tracks under external disturbances are each 500 N

Future studies will focus on integrating obstacle avoidance capabilities, implementing the proposed controller in real time, and exploring advanced control techniques to further enhance system performance.

References

[1] Kuwata Y, Teo J, Fiore G, Karaman S, Frazzoli E and How J P 2009 Real-time motion planning with applications to autonomous urban driving *IEEE Transactions on control systems technology* **17** 1105–1118
[2] Siciliano B, Khatib O and Kröger T 2008 *Springer Handbook of Robotics* vol 200 (Springer)
[3] Al-Milli S, Seneviratne L D and Althoefer K 2010 Track-terrain modelling and traversability prediction for tracked vehicles on soft terrain *Journal of Terramechanics* **47** 151–160
[4] Aguiar A S, Dos Santos F N, Cunha J B, Sobreira H and Sousa A J 2020 Localization and mapping for robots in agriculture and forestry: A survey *Robotics* **9** 97
[5] Jones D O, Gates A R, Huvenne V A, Phillips A B and Bett B J 2019 Autonomous marine environmental monitoring: Application in decommissioned oil fields *Science of the total environment* **668** 835–853
[6] Zeng F, Jacobson A, Smith D, Boswell N, Peynot T and Milford M 2019 Lookup: Vision-only real-time precise underground localisation for autonomous mining vehicles *International conference on robotics and automation (ICRA)* (IEEE) pp 1444–1450

[7] Vasiljević G, Miklić D, Draganjac I, Kovačić Z and Lista P 2016 High-accuracy vehicle localization for autonomous warehousing *Robotics and Computer-Integrated Manufacturing* **42** 1–16
[8] Acevedo J J, Arrue B C, Maza I and Ollero A 2014 A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities *International Conference on Robotics and Automation (ICRA)* (IEEE) pp 4735–4740
[9] Rogers J G, Nieto-Granda C and Christensen H I 2013 Coordination strategies for multi-robot exploration and mapping *Experimental Robotics: The 13th International Symposium on Experimental Robotics* (Springer) pp 231–243
[10] Wasson S R, Guilberto J, Ogg W, Wedeward K, Bruder S and El-Osery A 2004 An unmanned ground vehicle for landmine remediation *Detection and Remediation Technologies for Mines and Minelike Targets IX* vol 5415 (SPIE) pp 1231–1239
[11] Trevelyan J, Hamel W R and Kang S C 2016 Robotics in hazardous applications *Springer handbook of robotics* 1521–1548
[12] Kamel M A, Hafez A T and Yu X 2018 A review on motion control of unmanned ground and aerial vehicles based on model predictive control techniques *Journal of Engineering Science and Military Technologies* **2** 10–23
[13] Sebastian B and Ben-Tzvi P 2019 Physics based path planning for autonomous tracked vehicle in challenging terrain *Journal of Intelligent & Robotic Systems* **95** 511–526
[14] Sabiha A D, Kamel M A, Said E and Hussein W M 2022 Dynamic modeling and optimized trajectory tracking control of an autonomous tracked vehicle via backstepping and sliding mode control *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* **236** 620–633
[15] Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G *et al.* 2006 Stanley: The robot that won the darpa grand challenge *Journal of field Robotics* **23** 661–692
[16] Reinholtz C *et al.* 2009 Team victortango's entry in the darpa urban challenge
[17] Ahmad M, Polotski V and Hurteau R 2000 Path tracking control of tracked vehicles *IEEE International Conference on Robotics and Automation* vol 3 (IEEE) pp 2938–2943
[18] Al-Jarrah A, Salah M and Almomani F 2019 Controlling a skid-steered tracked mobile robot with slippage using various control schemes *International Conference on Research and Education in Mechatronics (REM)* (IEEE) pp 1–7
[19] Zou T, Angeles J and Hassani F 2018 Dynamic modeling and trajectory tracking control of unmanned tracked vehicles *Robotics and Autonomous Systems* **110** 102–111
[20] Wang S, Guo J, Mao Y, Wang H and Fan J 2023 Research on the model predictive trajectory tracking control of unmanned ground tracked vehicles *Drones* **7** 496
[21] Goud H, Sharma P C, Nisar K, Ibrahim A A A, Yadav N S, Swarnkar P, Gupta M and Chand L 2022 Metaheuristics algorithm for tuning of pid controller of mobile robot system *Computers, Materials & Continua* **72**
[22] Fierro R and Lewis F L 1997 Control of a nonholomic mobile robot: Backstepping kinematics into dynamics *Journal of robotic systems* **14** 149–163
[23] Tzafestas S G 2013 *Introduction to mobile robot control* (Elsevier)
[24] Kanayama Y, Kimura Y, Miyazaki F and Noguchi T 1990 A stable tracking control method for an autonomous mobile robot *IEEE International Conference on Robotics and Automation* (IEEE) pp 384–389
[25] Ahmad M, Hudha K, Abd Kadir Z, Subari M, Amer N and Dardin S 2020 Speed tracking control of tracked vehicle using pid controller optimized by particle swarm optimization *IEEE International Conference on Automatic Control and Intelligent Systems (ICACIS)* (IEEE) pp 107–111