UDC 621.865.8 https://doi.org/10.20998/2074-272X.2026.4.01

S. Ben Hadj Mohamed, A. Mahjoub, C. Ben Njima, A. Benamor

A novel interception and trajectory tracking control approach for a mobile robot

Introduction. *In nature, interception is a hunting strategy where a predator moves to a point ahead of a moving prey's trajectory to catch it, rather than directly pursuing it. Also, in transportation and manufacturing sectors, trajectory interception is carried out by the correspondence of the position and velocity of a target object with those of the robot interceptor. It is within this context that our research work takes place. The* ***problem*** *of the work consists in the development of a new intercepting and trajectory tracking strategy of a two-wheeled differential-drive mobile robot.* ***Goal****. To propose a novel intercepting and trajectory tracking technique whose principle is based on the orientation angle of the mobile robot interceptor guarantees a faster convergence with a minimum error and lower energy consumption.* ***Methodology****. The problem is solved using both a sliding mode controller and a backstepping controller to test the proposed strategy based on particle swarm optimization (PSO).* ***Results.*** *The results proved the effectiveness of the new approach especially in fast reaching-time and energy consumption compared to direct pursuit. In other words, the results indicate that the proposed approach achieves a noticeable reduction in convergence time (up to 82.5% faster) and significantly lowers oscillations in the control signals compared to classical methods.* ***Scientific novelty****. To get interception and accurate tracking in a reduced reaching-time, an original control technique based on PSO is implemented using two different controllers.* ***Practical value.*** *The proposed strategy offers satisfactory control performances such as fast interception and smooth trajectory tracking.* References 24, tables 6, figures 14.

Key words: **trajectory tracking, mobile robot, sliding mode control, backstepping control, particle swarm optimization.**

Вступ*. У природі перехоплення є стратегією полювання, за якої хижак рухається не безпосередньо за здобиччю, а до точки попереду її траєкторії для забезпечення захоплення. Аналогічно у транспортній та виробничій сферах перехоплення траєкторії здійснюється шляхом узгодження положення та швидкості цільового об'єкта з параметрами робота-перехоплювача. Саме в цьому контексті виконано дане дослідження.* ***Проблема****. Робота присвячена розробленню нової стратегії перехоплення та відстеження траєкторії для двоколісного мобільного робота з диференціальним приводом.* ***Мета.*** *Запропонувати новий метод перехоплення та відстеження траєкторії, принцип якого ґрунтується на використанні кута орієнтації мобільного робота-перехоплювача та забезпечує швидшу збіжність, мінімальну похибку і менше енергоспоживання.* ***Методика****. Для розв'язання поставленої задачі використано регулятор ковзного режиму та бекстепінг-регулятор з метою перевірки запропонованої стратегії на основі оптимізації роєм часток (PSO).* ***Результати****. Отримані результати підтвердили ефективність нового підходу, зокрема щодо зменшення часу досягнення цілі та енергоспоживання порівняно з методом прямого переслідування. Зокрема, запропонований підхід забезпечує суттєве скорочення часу збіжності (до 82,5 %) та значне зменшення коливань у сигналах керування порівняно з класичними методами.* ***Наукова новизна****. Для забезпечення перехоплення та точного відстеження траєкторії за мінімального часу збіжності реалізовано оригінальний метод керування на основі PSO із використанням двох різних регуляторів.* ***Практична значимість****. Запропонована стратегія забезпечує високі показники якості керування, зокрема швидке перехоплення та плавне відстеження траєкторії.* Бібл. 24, табл. 6, рис. 14.

Ключові слова: **відстеження траєкторії, мобільний робот, керування ковзним режимом, бекстепінг-керування, оптимізація роєм часток.**

Introduction. Nowadays, the control of mobile robots is swiftly in continuous progress, with a principal focus on the control of robots capable of autonomous movement and tasks performance. Particularly, the trajectory tracking problem represents one of the most important challenges. The robot control study gave rise to several different techniques which have been proposed to get efficient control of mobiles robots.

To control the position of a simulated mobile robot, the authors in [1] used two reinforcement learning algorithms to control the position of a simulated robot. The authors showed the effectiveness for position control in environments with and without obstacles. In [2], the authors addressed the challenge of controlling a mobile robot to reach a target posture in a minimum time. He used a dynamic adaptive PID controller based on genetic algorithm optimization.

Certain studies, to ameliorate the trajectory tracking capabilities of a mobile robot, developed a neural network controller [3]. Also, [1, 4] proposed an artificial intelligence control method to minimize movement time of a mobile robot, and a control gains optimization using genetic algorithm has been done in the first research. These works focus exclusively on the tracking performances and reaching time but they don't emphasize the minimization of the convergence error.

Many other research works used model predictive control strategy; for example [5], focused on controlling mobile robots, they proposed a model predictive controller by minimizing quadratic criterion after linearizing the nonlinear dynamic model of the differential drive mobile robot using input-output linearization technique.

Many research studies have been elaborated concerning the tracking and interception of a moving target in the literature. In [6], the authors developed a control strategy for the interception of a moving object by a wheeled mobile robot. The called guidance approach allowed the mobile robot, when it is faster, to intercept successfully the moving target. Many research studies used sliding mode control techniques to get robust systems with insensitivity to external disturbances and uncertainties [7]. The work [8] proposed a new interception algorithm for a manipulator robot to intercept a moving object. To reach interception objective in a prescribed time, the authors used the combination of a robust second-order sliding mode control with a terminal attractor based on a designed time base generator, while [9] developed a nonlinear sliding mode control approach to achieve trajectory tracking of 2-wheeled mobile robot. These two works focus on the robustness in prescribed time interception and rejecting disturbances, respectively, but they neglect to be interested in the minimization of the convergence error. On the other hand, in [10], the authors addressed the problem of controlling uncertain 4-wheeled mobile robot presenting a WSS phenomenon (wheel slipping, skidding). The sliding mode control [11, 12] technique has been used to achieve a fixed-time prescribed control performance. Despite the robustness achieved and the convergence error that the authors have mentioned, it remains to be verified whether this strategy works with any controller that can be used.

However, most of these research studies don't focus on the reaching time and the minimization of the position and orientation angle error.

The goal of the paper is to propose a novel intercepting and trajectory tracking technique whose principle is based on the orientation angle of the mobile robot interceptor guarantees a faster convergence with a minimum error and lower energy consumption.

To show the effectiveness of this technique, it will be tested using a first-order sliding mode controller and a backstepping controller.

Problem formulation. Consider a mobile robot which is placed in a 2D reference mark. The mobile robot is counterclockwise oriented with an angle relative to *X*-axis. Figure 1,*a* shows the mobile robot, the posture errors consisting of linear positions (*x*, *y*) and the orientation angle θ in the global frame *OXY*. The robot's kinematic model is defined by:

$$\dot{q} = S(q)u\,, \tag{1}$$

where $q = (x, y, \theta)^T$ is the actual robot posture; a Jacobian matrix is $[\boldsymbol{S}(q)] = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 1 & 1 \end{bmatrix}$; $[\boldsymbol{u}] = \begin{bmatrix} v \\ w \end{bmatrix}$ is the velocity control vector; v is the linear velocity; w is the angular velocity.

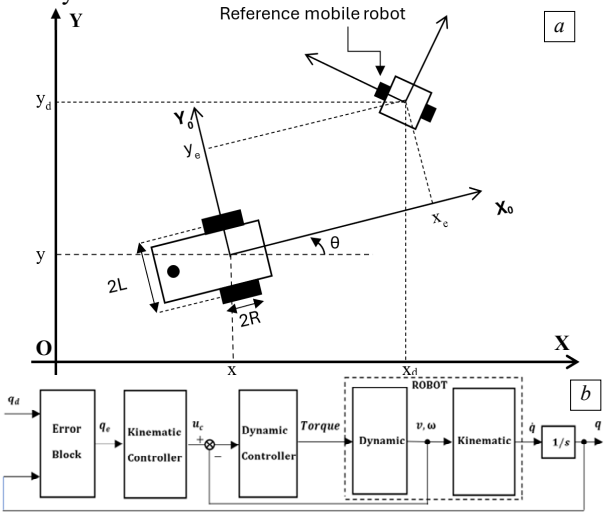

Fig. 1. Mobile robot posture error (*a*); closed loop of controlled system (*b*)

The desired posture and the desired velocity vector are defined as:

$$q_d = (x_d, y_d, \theta_d)\,; \quad V_d = (v_d, w_d)\,. \tag{2}$$

The posture error in the robot's local frame is:

$$q_e = (x_e, y_e, \theta_e)\,, \tag{3}$$

where:

$$\begin{cases} x_e = \cos(x_d - x) + \sin\theta(y_d - y); \\ y_e = -\sin\theta(x_d - x) + \cos(y_d - y); \\ \theta_e = \theta_d - \theta, \end{cases} \tag{4}$$

and the vector of posture error $[\boldsymbol{q}_e]$ has the next form:

$$[\boldsymbol{q}_e] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}, \tag{5}$$

we can also write:

$$[\boldsymbol{q}_e] = \Re(\theta) \times \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix},$$

where $\Re(\theta)$ is the orthogonal rotation matrix.

The nonholonomic constraints equation characterizing the slip-free rolling of a wheel on the ground is:

$$\dot{x}\sin\theta + \dot{y}\cos\theta = 0\,. \tag{6}$$

Using (1), (6), the derivative of (5) is:

$$[\mathrm{d}\boldsymbol{q}_e/\mathrm{d}t] = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} y_e\omega - v + v_d\cos\theta_e \\ -x_e\omega + v_d\sin\theta_e \\ \omega_d - \omega \end{bmatrix}. \tag{7}$$

To build a mobile robot simulation that is close to reality, the robot dynamic model and a dynamic controller are introduced in the block diagram of the control strategy:

$$\begin{cases} \left(m + \dfrac{2}{R^2} I_w\right) - m_c d\omega^2 = \dfrac{1}{R}(T_R + T_L); \\ \left(I + \dfrac{2L^2}{R^2} I_w\right)\dot{\omega} + m_c d\omega v = \dfrac{L}{R}(T_R - T_L), \end{cases} \tag{8}$$

where (T_R, T_L) are the torques input of the mobile robot; L is the half of the track width; R is the wheel radius; d is the distance from the mass center to the middle point between the right and left wheels; m, m_c are the total mass of the robot and the robot mass without its wheels and actuators; I_w is the moment of inertia of each wheel; I is the total inertia moment.

Optimal reference trajectory tracking. In this part, we present our new interception and tracking technique based on a proposed desired orientation angle. Many results have been achieved based on mobile robot control which emphasized on trajectory tracking performances [13, 14]. All these works are differentiated in the method to calculate the posture error and especially in the choice of the desired orientation angle. There are authors who note the desired angle as the orientation of the reference mobile robot as in [12].

Others note the desired angle as the orientation towards the reference mobile robot [15–18]:

$$\theta_d = \theta_r; \tag{9}$$

$$\theta_{e1} = \theta_r - \theta\,; \tag{10}$$

$$\theta_d = \theta_{TR} = \tan^{-1}\left((y_d - y)/(x_d - x)\right). \tag{11}$$

The θ_{TR} expression returns values in the interval $[-\pi/2, \pi/2]$ and may cause division-by-zero errors when $x_d = x$.

The function «atan2» is used instead of $\tan^{-1}$ to return the 4-quadrant inverse tangent, so:

$$\theta_d = \theta_{TR} = \mathrm{atan2}[(y_d - y); (x_d - x)]; \tag{12}$$

$$\theta_{e2} = \theta_{TR} - \theta\,. \tag{13}$$

The proposed method is to take as the desired angle the orientation towards a virtual reference mobile robot, $\theta_d = \theta_{TVR}$.

Figure 2 depicts the graphical the proposed strategy where the virtual reference mobile robot is the predicted posture of the reference mobile robot after a period Δt, with Δt is the minimum-time interception of the moving reference mobile robot such that $\lim\limits_{t\to\Delta t} q_e = 0$. Assumption: the linear velocities of the actual mobile robot and the reference mobile robot are positive. We define: $d_1 = v_d{\cdot}\Delta t$ is the distance between the reference mobile robot and its predicted position; $d_3 = v{\cdot}\Delta t$ is the distance between the virtual reference robot and the actual mobile robot; $d_2 = \sqrt{(y_d - y)^2 + (x_d - x)^2}$ is the distance between the reference mobile robot and the actual mobile robot:

$$\theta_d = \theta_{TVR} = \theta_{TR} + \theta_p, \quad (14)$$

where θ_p is the angle resulted from the prediction

So, it is readily seen that with a rotation of $-\theta_{TR}$ in Fig. 2,*c* we obtain the following Fig. 3.

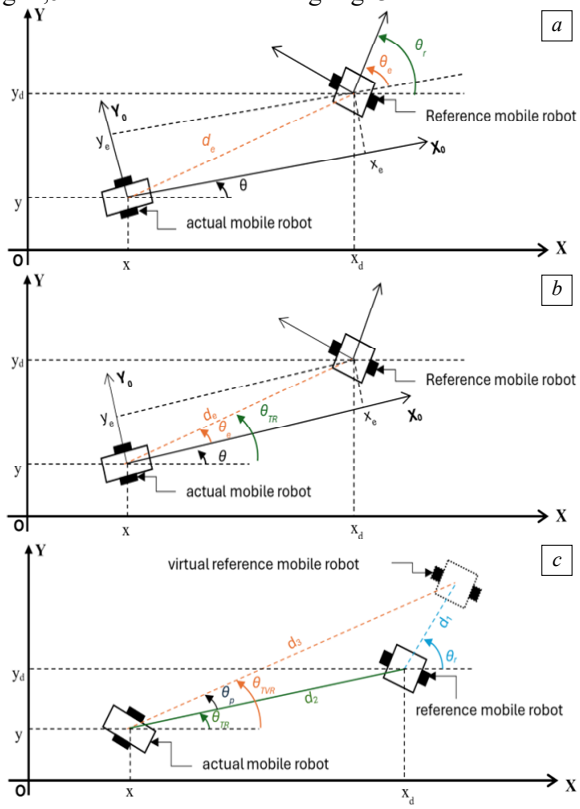

Fig. 2. *a* – the desired angle θ_d is the orientation of the reference mobile robot θ_r; *b* – the desired angle $\theta_d = \theta_{TR}$ is the orientation towards the reference mobile robot; *c* – the proposed strategy $\theta_d = \theta_{TVR}$

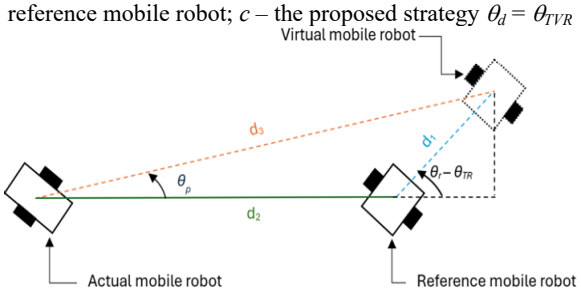

Fig. 3. Rotation of Fig. 2,*c* of $-\theta_{TR}$

Then, we get:

$$\theta_p = \sin^{-1}\left(\frac{v_d}{v}\cdot\sin(\theta_r - \theta_{TR})\right), \quad (15)$$

with $v \neq 0$ and $-1 \le \sin(\theta_r - \theta_{TR}) \le 1$.

The inverse sine «$\sin^{-1}$» returns one value of in the interval $[-\pi/2, \pi/2]$, however there are 2 solutions: θ_p or $\theta_p - \pi$, $|\theta_p| \in [0, |\theta_r - \theta_{TR}|]$. If $\theta_p \ge 0$ then $\theta_{TR} \le \theta_d \le \theta_r$; if $\theta_p \le 0$ then $\theta_r \le \theta_d \le \theta_{TR}$; if $\theta_p = 0$ then $\theta_r = \theta_d = \theta_{TR}$.

According to Fig. 3, we can write:

$$d_3\cos(\theta_p) = d_1\cos(\theta_r - \theta_{TR}) + d_2;$$

$$\cos(\theta_p)V\Delta t = \cos(\theta_r - \theta_{TR})v_d\Delta t + d_2;$$

$$\Delta t = \frac{d_2}{\cos(\theta_p)v - \cos(\theta_r - \theta_{TR})v_d};$$

$\Delta t \ge 0$ leads to

$$\cos(\theta_p)v - \cos(\theta_r - \theta_{TR})v_d > 0;$$

$$\cos(\theta_p)v > \cos(\theta_r - \theta_{TR})v_d.$$

The error angle $\theta_{e3} = \theta_{TVR} - \theta$:

$$\theta_{e3} = \text{atan2}((y_d - y),(x_d - x)) + \sin^{-1}\left(\frac{v_d}{v}\sin(\theta_r - \theta_{TR})\right) - \theta. \quad (16)$$

The angle error is minimized when $|\theta_{e3}| > \pi$ to avoid the long path when the mobile robot rotates:

$$\text{if } \theta_{e3} > \pi \text{ then } \theta_{e3\min} = \theta_{e3} - 2\pi, \quad (17)$$

$$\text{if } \theta_{e3} < -\pi \text{ then } \theta_{e3\min} = \theta_{e3} + 2\pi. \quad (18)$$

So $\theta_{e3\min} \in [-\pi, \pi]$.

The proposed interception strategy flowchart is shown in Fig. 4 and illustrates how the posture error is calculated at each step of time based on the proposed desired angle θ_{TVR}. Here, the control law block and the error block are separated. This separation gives the possibility to apply this approach with any control law and with any model of mobile robot, offering flexibility and adaptability in various control frameworks.

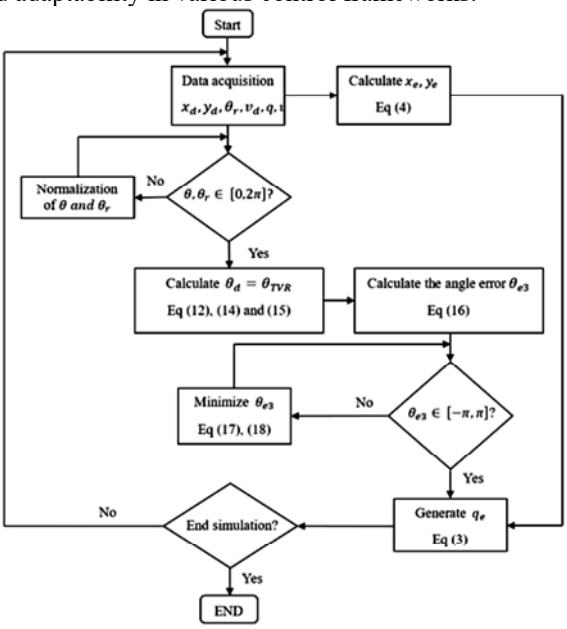

Fig. 4. Proposed interception strategy flowchart

Kinematic controller. In this section, 2 kinematic controllers will be presented for the test with the 3 desired angles $\theta_{d1} = \theta_r$, $\theta_{d2} = \theta_{TR}$, $\theta_{d3} = \theta_{TVR}$. Simulink was used for the development of the mobile robot's control system (Fig. 1,*b*), leveraging its block-based environment to facilitate rapid prototyping, model-based simulation, and to improve analysis and debugging.

The overall architecture of the control system is illustrated in the Simulink model flowchart presented in Fig. 5, which details the interconnections and data flow between the various functional blocks.

First order sliding mode controller (FOSMC). The trajectory tracking control law developed by [19] is based on the FOSMC technique using a non-singular coordinate transformation. This formulation enables improved stability and control near the target by avoiding singularities that occur in Cartesian coordinates. Therefore, the control law is defined as:

$$[\boldsymbol{u}_c] = \begin{bmatrix} v_c \\ w_c \end{bmatrix} = -B_0^{-1}A_{0_\sigma} - G\text{sat}(\sigma^*), \quad (19)$$

with

$$[\boldsymbol{A}_{0\sigma}]=\frac{\partial\sigma}{\partial q_e}A_0=\begin{bmatrix}\lambda_1 v_d\cos(\theta_e)\\ \lambda_2 v_d\sin(\theta_e)+\lambda_3\omega_d\end{bmatrix};$$

$$[\boldsymbol{B}_{0\sigma}]=\frac{\partial\sigma}{\partial q_e}B_0=\begin{bmatrix}-\lambda_1 & \lambda_1 y_e\\ 0 & -\lambda_1 x_e-\lambda_3\end{bmatrix};$$

$$\left[\boldsymbol{\sigma}^*(q_e,t)\right]=\begin{bmatrix}\sigma_1^*\\ \sigma_2^*\end{bmatrix}\begin{bmatrix}-\lambda_1^2 x_e\\ \lambda_1 x_e y_e-(\lambda_2 x_e+\lambda_3)(\lambda_2 y_e+\lambda_3\theta_e)\end{bmatrix}. \quad (20)$$

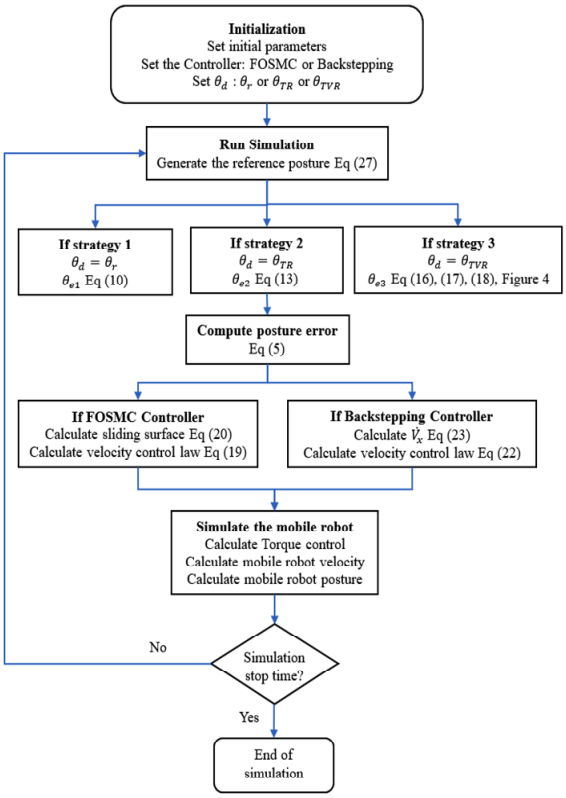

Fig. 5. Simulink model flowchart of the control system

In [19] the «sign» function was replaced by a saturation function to reduce the chattering phenomenon, where sat(σ^*) is component wise discontinuous and defined as:

$$[\text{sat}(\sigma^*)]=\left[sat(\sigma_1^*)\quad sat(\sigma_2^*)\right]^T;$$

$$[\text{sat}(\sigma^*)]=\begin{cases}\dfrac{\sigma_i^*}{\phi}, \text{ if } \left|\sigma_i^*\right|\le\phi;\\ \text{sign}(\sigma_i^*), \text{ if } \left|\sigma_i^*\right|>\phi;\end{cases}$$

where ϕ is the boundary layer thickness (i = 1, 2);

$$G=\text{diag}([g_1, g_2]). \quad (21)$$

To ensure that $B_{0\sigma}$ is always nonsingular, the following condition must be satisfied:

$$\lambda_2=\lambda_3 k \quad \text{with} \quad 0\le k\le\frac{1}{|x_e|+1}.$$

Backstepping controller. In [21, 22], the tracking error x_e is replaced by V_x in the backstepping model so that we obtain a new backstepping controller based on biologically-inspired shunting neural model, the control law is defined as follows:

$$[\boldsymbol{u}_c]=\begin{bmatrix}v_c\\ w_c\end{bmatrix}=\begin{bmatrix}k_1 V_x+v_d\cos\theta_e\\ \omega_d+k_2 v_d y_e+k_3 v_d\sin\theta_e\end{bmatrix}, \quad (22)$$

where $k_1 - k_3$ are the given positive constants;

$$\dot{V}_x=-AV_x+(B-V_x)f(x_e)-(D+V_x)g(x_e). \quad (23)$$

where V_x is the biological model control output voltage, so that $f(x_e)$ = max(x_e, 0) and $g(x_e)$ = max(–x_e, 0); x_e is the biological model control input; A, B, D are the positive constants such as $B = D$.

PSO optimization. In this part, we developed a MATLAB code based on the predefined function «particleswarm» to optimize controllers' gains in a Simulink model. The gains in the controllers of sliding mode and backstepping directly influence the performance of the control technique for interception and trajectory tracking. Correctly adjusted gains ensure fast convergence and enhanced robustness of the control system. To achieve optimal controller performance, gain tuning, which can be a time-consuming and complicated phase, is a primordial task. To optimize the parameters of the controllers, we adopt the particle swarm optimization (PSO) algorithm [23]. The PSO algorithm is applied within the trajectory tracking control framework to get right control gains by optimizing a predefined objective function.

An integral time square error (ITSE) based on the posture error vector is selected as the objective function:

$$J_{ITSE}=\int t\cdot q_e^2\,\mathrm{d}t. \quad (24)$$

In the PSO algorithm, iterative computation is performed with the objective of minimizing the cost function J_{ITSE}. Each particle in the swarm represents a candidate solution, where its position corresponds to a specific set of control gains to be optimized. During the optimization process, particles move through the solution space with a certain velocity, updating their positions iteratively. Through this process, the swarm converges toward the global optimal solution.

Inspired from the problem-based approach to optimize a function using particle swarm, a custom objective function file was developed. To overcome the limitations of converting the function file to an optimization expression, the Simulink model is executed, and the resulting ITSE value is returned to «particleswarm» as a «black-box» cost function.

Figure 6 shows the interaction between MATLAB and Simulink during the optimization process and the PSO algorithm can be described in the next main steps.

Initialization: particles are randomly initialized within the parameter bounds. The objective function is evaluated for each particle. Each particle's best position p_i and the global best position g are recorded.

Velocity update: the velocity of each particle is updated using the following equation:

$$v_{i+1}=Wv_i+y_1u_1(p_i-x_i)+y_2u_2(g_i-x_i), \quad (25)$$

where W is the inertia weight; y_1, y_2 are the acceleration coefficients (self and social adjustment); u_1, u_2 are the random numbers uniformly distributed in (0, 1).

Position update: each particle's position is updated by:

$$x_{i+1}=x_i+v_{i+1}. \quad (26)$$

Boundary constraints are applied to ensure all parameters remain within their valid ranges.

Evaluation and update: The new position is evaluated using the cost function. If it yields a better result, the particle's personal best is updated. The global best is also updated if a better solution is found across all particles.

Adaptation: the neighborhood size and inertia weight W are adapted based on the improvement of the global best solution to enhance convergence behavior.

Termination: the algorithm repeats until a stopping condition is met, such as a maximum number of iterations or a convergence threshold.

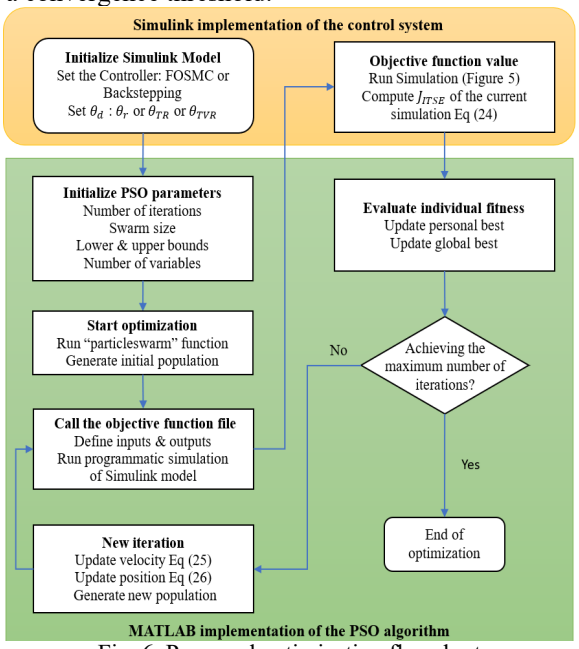

Fig. 6. Proposed optimization flowchart

By optimizing the controller gains through PSO in offline mode, the control system achieves better performance in terms of tracking accuracy and faster convergence, all while maintaining low computational complexity during robot simulation.

Simulation results and discussion. To highlight the advantages and the effectiveness of the proposed control approach, a simulation is carried out on the trajectory tracking of a circle to verify the effectiveness of the strategy used.

The initial robot state x, y and θ are respectively $6m$, $2m$ and π. The initial states of the reference robot are $x_d = m$, $y_d = 0$, $\theta_r = \pi/2$.

The expected trajectory of circle can be concluded as:

$$\begin{cases} x_r = v_d \cos(\omega_d t); \\ y_r = v_d \sin(\omega_d t); \\ \theta_r = \omega_r t, \end{cases} \quad (27)$$

where $v_d = 4$ m/s, $\omega_d = 1$ rad/s.

We consider that the mass center of the mobile robot is at the middle point between the right and left wheels (d=0), and the moment of inertia of each wheel is neglected ($I_w = 0$). The robot parameters are listed in Table 1.

Table 1

Robot parameters

L, m	R, m	m, kg	I, kg·m^2
0.15	0.03	4.5	3

The parameters of kinematic controllers in Table 2 remain constant throughout the simulation experiments.

While gains g_1, g_2 of the FOSMC controller and A, B, D of the backstepping controller are optimized by the PSO each time the desired angle is selected. The optimization process is executed offline six times for the two kinematic controllers and the three desired angles. Tables 3, 4 show the optimal controllers' gains and the total simulation errors reflected by the cost function J_{ITSE}.

Table 2

Constant parameters of the controllers

FOSMC controller		Backstepping controller	
ϕ	0.5	k_1	2
λ_1	1	k_2	6
λ_2	6	k_3	6
λ_3	1		

Table 3

Optimal gains of the FOSMC controller

Desired angle	g_1	g_2	J_{ITSE}
$\theta_{d1} = \theta_r$	32.5166	44.9584	0.1152
$\theta_{d2} = \theta_{TR}$	0.9444	82.9444	0.4175
Proposed $\theta_{d3} = \theta_{TVR}$	13.1741	32.7398	0.0588

Table 4

Optimal gains of the backstepping controller

Controller	Optimal FOSMC controller		Optimal backstepping controller	
Classical strategies	$\theta_{d1} = \theta_r$	$\theta_{d2} = \theta_{TR}$	$\theta_{d1} = \theta_r$	$\theta_{d2} = \theta_{TR}$
Time reduction using our strategy $\theta_{d3} = \theta_{TVR}$	30 %	82.5 %	8.33 %	81.66 %

The minimal error obtained by the cost function while $\theta_{d3} = \theta_{TVR}$ indicates the high accuracy of the proposed strategy. The responses of both controllers in Fig. 7, 8 show a fast and smooth tracking trajectory in the X-Y plane using the proposed desired angle θ_{TVR}.

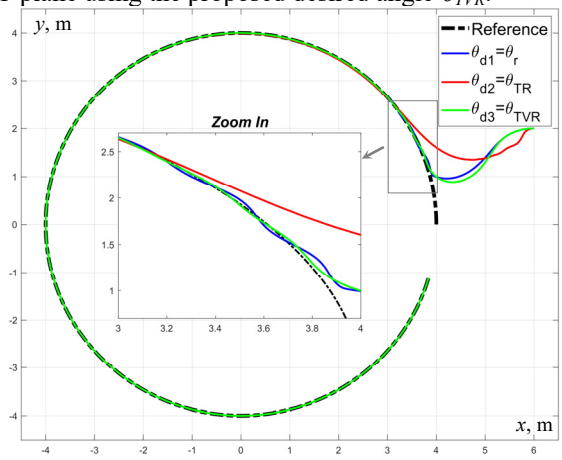

Fig. 7. Circular trajectory tracking – optimal FOSMC controller

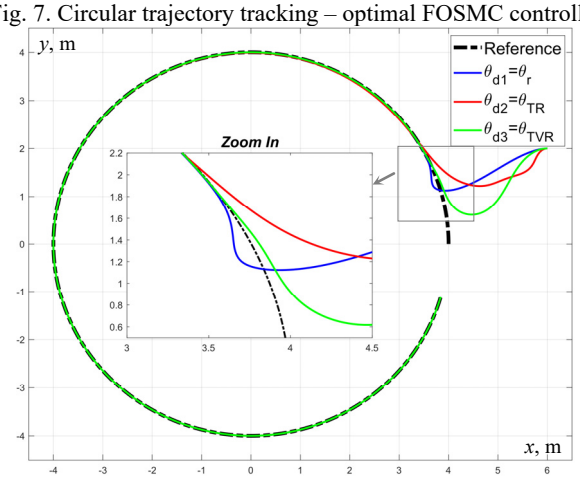

Fig. 8. Circular trajectory tracking – optimal backstepping controller

The convergence time values of posture error $q_e = (x_e, y_e, \theta_e)^T$ to the steady-stable value can be extracted from Fig. 9–12 and all listed in Table 5. During the transient phase, as it is clear in Fig. 11, the optimal FOSMC controller exhibits less oscillations around the reference trajectory while trajectory while $\theta_{d3} = \theta_{TVR}$.

Table 5

Reduction in convergence time, s

Desired angle	Optimal FOSMC controller	Optimal backstepping controller
$\theta_{d1} = \theta_r$	1	1.2
$\theta_{d2} = \theta_{TR}$	4	6
$\theta_{d3} = \theta_{TVR}$	0.7	1.1

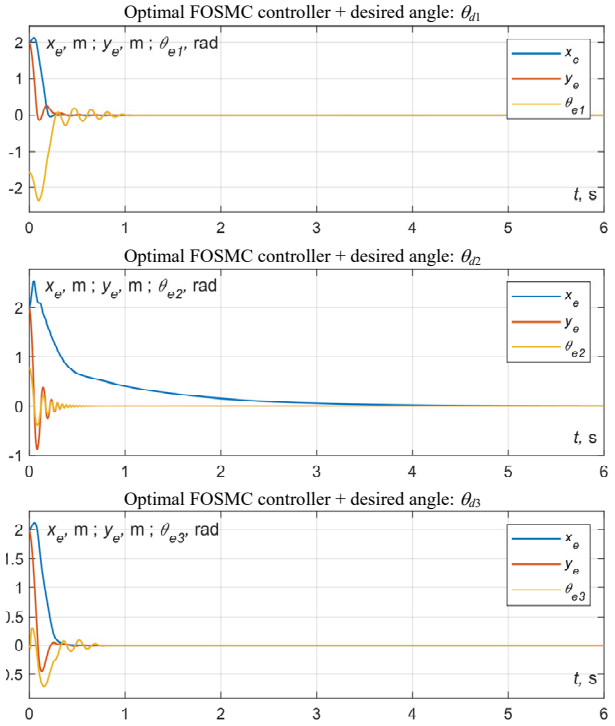

Fig. 9. Posture error – optimal FOSMC controller

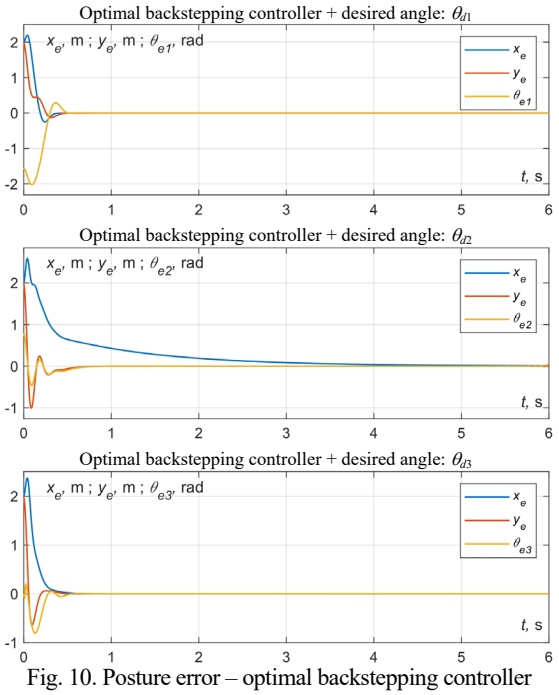

Fig. 10. Posture error – optimal backstepping controller

To evaluate the convergence speed of the proposed strategy, the Table 6 summarizes, for each kinematic controller and each desired angle, the reduction in convergence time of the posture error using the interception strategy.

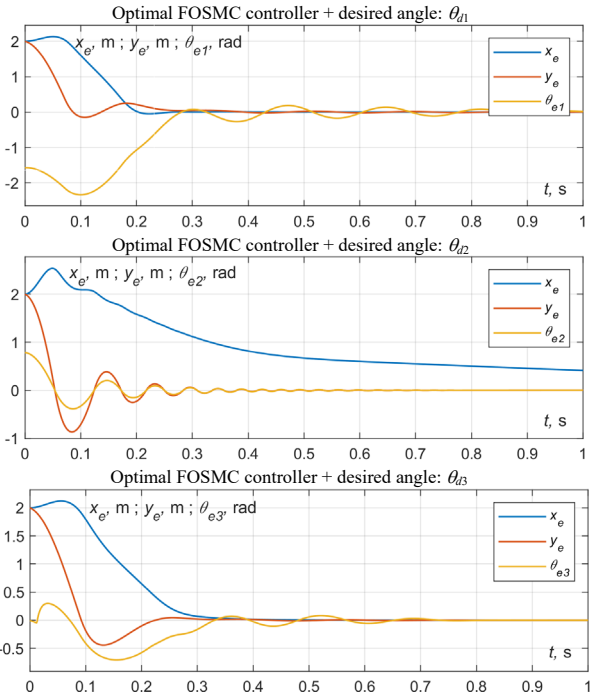

Fig. 11. Zoom in posture error – optimal FOSMC controller

Table 6

Reduction in convergence time, s

Desired angle	A	$B = D$	J_{ITSE}
$\theta_{d1} = \theta_r$	37.2667	100	0.1099
$\theta_{d2} = \theta_{TR}$	94.2135	18.2315	0.4476
Proposed $\theta_{d3} = \theta_{TVR}$	55.1726	100	0.0418

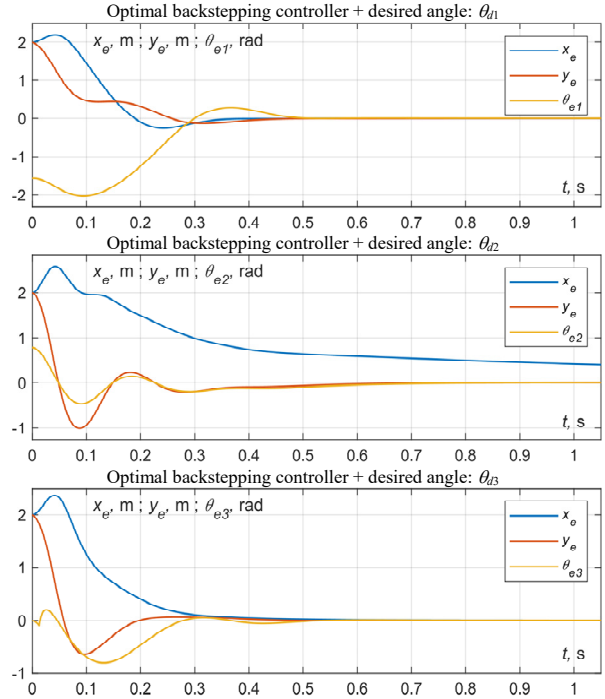

Fig. 12. Zoom in posture error – optimal backstepping controller

As shown in Fig. 13, 14, the kinematic controllers significantly reduce the chattering phenomenon during the transient phase, resulting in a smoother and less oscillatory control signal compared to the conventional approach. This strategy is crucial to our energy-saving strategy, as it effectively reduces transient oscillations, which are known to increase heat losses and decrease the efficiency of a robotic system's actuators [24].

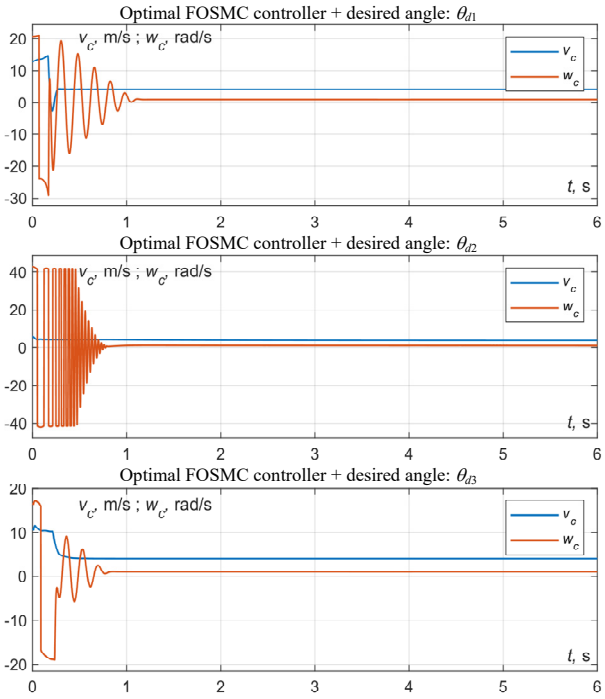

Fig. 13. Output of the optimal FOSMC controller

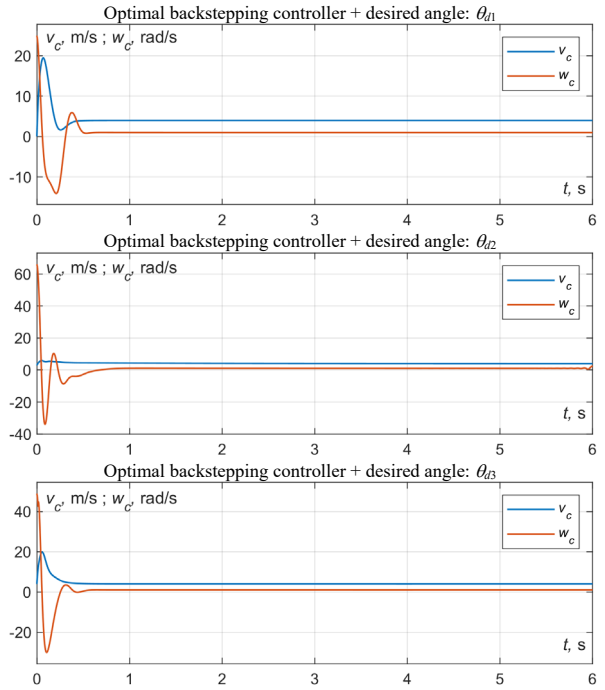

Fig. 14. Output of the optimal backstepping controller

Conclusions. This paper introduced a novel interception-based trajectory tracking control strategy for differential-drive mobile robots, and a new approach for optimizing controller gains was introduced, which effectively combines the global optimization capabilities of MATLAB's «particleswarm» function with the block-based modeling environment of Simulink. The results indicate that the proposed approach achieves a noticeable reduction in convergence time (up to 82.5 % faster) and significantly lowers oscillations in the control signals compared to existing classical methods from the literature. These enhancements are directly linked to the newly designed reference orientation and the systematic gain optimization that form the core contributions of this work. The simulation results show the effectiveness of the proposed interception and trajectory tracking strategy, regardless of the controller used. Future efforts will focus on performing real-time experiments on an embedded control platform to further validate robustness against physical disturbances and unmodeled dynamics. Additionally, expanding the framework through adaptive or learning-based mechanisms is expected to support broader operational conditions while maintaining strong tracking performance.

Conflict of interest. The authors declare that they have no conflicts of interest.

REFERENCES

1. Quiroga F., Hermosilla G., Farias G., Fabregas E., Montenegro G. Position Control of a Mobile Robot through Deep Reinforcement Learning. *Applied Sciences*, 2022, vol. 12, no. 14, art. no. 7194. doi: https://doi.org/10.3390/app12147194.

2. Park S.-C., Kim M., Park H.-M., Park J.-H. Optimal Control Gains Optimization for Mobile Robot Considering Dynamic Constraints. *IEEE Access*, 2024, vol. 12, pp. 180079-180092. doi: https://doi.org/10.1109/ACCESS.2024.3507739.

3. Trujillo D., Morales L.A., Chávez D., Pozo D.F. Trajectory Tracking Control of a Mobile Robot using Neural Networks. *Emerging Science Journal*, 2023, vol. 7, no. 6, pp. 1843-1862. doi: https://doi.org/10.28991/ESJ-2023-07-06-01.

4. Park J.-H. Optimal Controller Design for a Mobile Robot Using Genetic Algorithm and Adaptive PID Controller. *IEEE Access*, 2025, vol. 13, pp. 86167-86184. doi: https://doi.org/10.1109/ACCESS.2025.3570472.

5. Bouzoualegh S., Guechi E.-H., Kelaiaia R. Model Predictive Control of a Differential-Drive Mobile Robot. *Acta Universitatis Sapientiae Electrical and Mechanical Engineering*, 2018, vol. 10, no. 1, pp. 20-41. doi: https://doi.org/10.2478/auseme-2018-0002.

6. Belkhouche F., Belkhouche B. On the tracking and interception of a moving object by a wheeled mobile robot. *IEEE Conference on Robotics, Automation and Mechatronics*, 2004, vol. 1, pp. 130-135. doi: https://doi.org/10.1109/RAMECH.2004.1438904.

7. Bkekri R., Benamor A., Alouane M.A., Fried G., Messaoud H. Robust adaptive super twisting controller: methodology and application of a human-driven knee joint orthosis. *Industrial Robot: The International Journal of Robotics Research and Application*, 2019, vol. 46, no. 4, pp. 481-489. doi: https://doi.org/10.1108/IR-09-2018-0198.

8. Flores-Campos J.A., Torres-San-Miguel C.R., Paredes-Rojas J.C., Perrusquía A. Prescribed Time Interception of Moving Objects' Trajectories Using Robot Manipulators. *Robotics*, 2024, vol. 13, no. 10, art. no. 145. doi: https://doi.org/10.3390/robotics13100145.

9. Hameed I.A., et. al. A New Nonlinear Dynamic Speed Controller for a Differential Drive Mobile Robot. *Entropy*, 2023, vol. 25, no. 3, art. no. 514. doi: https://doi.org/10.3390/e25030514.

10. Nguyen V.-C., Kim S.H. A novel fixed-time prescribed performance sliding mode control for uncertain wheeled mobile robots. *Scientific Reports*, 2025, vol. 15, no. 1, art. no. 5340. doi: https://doi.org/10.1038/s41598-025-89126-6.

11. Bouraghda S., Sebaa K., Bechouat M., Sedraoui M. An improved sliding mode control for reduction of harmonic currents in grid system connected with a wind turbine equipped by a doubly-fed induction generator. *Electrical Engineering & Electromechanics*, 2022, no. 2, pp. 47-55. doi: https://doi.org/10.20998/2074-272X.2022.2.08.

12. Sakri D., Laib H., Farhi S.E., Golea N. Sliding mode approach for control and observation of a three phase AC-DC pulse-width modulation rectifier. *Electrical Engineering & Electromechanics*, 2023, no. 2, pp. 49-56. doi: https://doi.org/10.20998/2074-272X.2023.2.08.

13. Goto W.J.N., Martins N.A. Leader-follower formation tracking for differential-drive wheeled mobile robots with uncertainties and disturbances based on immune fuzzy quasi-sliding mode control. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2024, vol. 46, no. 2, art. no. 91. doi: https://doi.org/10.1007/s40430-023-04650-8.

14. Lin L., Wu P., He B., Chen Y., Zheng J., Peng X. The sliding mode control approach design for nonholonomic mobile robots based on non-negative piecewise predefined-time control law. *IET Control Theory & Applications*, 2021, vol. 15, no. 9, pp. 1286-1296. doi: https://doi.org/10.1049/cth2.12122.

15. Tang M., Tang K., Zhang Y., Qiu J., Chen X. Motion/force coordinated trajectory tracking control of nonholonomic wheeled mobile robot via LMPC-AISMC strategy. *Scientific Reports*, 2024, vol. 14, no. 1, art. no. 18504. doi: https://doi.org/10.1038/s41598-024-68757-1.

16. Cao T.N.T., Pham B.T., Tran H.D., Gia L.X., Nguyen N.T., Truong V.N. Non-singular terminal sliding mode control for trajectory-tracking of a differential drive robot. *E3S Web of Conferences*, 2024, vol. 496, art. no. 02005. doi: https://doi.org/10.1051/e3sconf/202449602005.

17. Hassan I.A., Abed I.A., Al-Hussaibi W.A. Path Planning and Trajectory Tracking Control for Two-Wheel Mobile Robot. *Journal of Robotics and Control (JRC)*, 2023, vol. 5, no. 1, pp. 1-15. doi: https://doi.org/10.18196/jrc.v5i1.20489.

18. Bentalba S., Haijaji A.E., Rachid A. Fuzzy sliding mode control of mobile robot. *Proceedings of the 37th IEEE Conference on Decision and Control*, 1998, vol. 4, pp. 4264-4265. doi: https://doi.org/10.1109/CDC.1998.761973.

19. Nardenio A.M., Douglas W.B. *Wheeled Mobile Robot Control Theory, Simulation and Experimentation*. Springer, 2022. 220 p.

20. Mohamed S.B.H., Mahjoub A., Benamor A. Optimal Sliding Mode Control for Differential Drive Mobile Robots. *2025 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2025, pp. 1-6. doi: https://doi.org/10.1109/ICCAD64771.2025.11182005.

21. Yang X., Wei P., Zhang Y., Liu X., Yang L. Disturbance Observer Based on Biologically Inspired Integral Sliding Mode Control for Trajectory Tracking of Mobile Robots. *IEEE Access*, 2019, no. 7, pp. 48382-48391. doi: https://doi.org/10.1109/ACCESS.2019.2907126.

22. Mohamed S.B.H., Mahjoub A., Benamor A. Optimal backstepping control Based on Biologically Inspired model for Trajectory Tracking of Mobile Robots. *2025 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2025, pp. 1-6. doi: https://doi.org/10.1109/ICCAD64771.2025.11099297.

23. Alnaib I.I., Alsammak A.N. Optimization of fractional PI controller parameters for enhanced induction motor speed control via indirect field-oriented control. *Electrical Engineering & Electromechanics*, 2025, no. 1, pp. 3-7. doi: https://doi.org/10.20998/2074-272X.2025.1.01.

24. Happyanto D.C., Aditya A.W. Chattering reduction effect on power efficiency of IFOC based induction motor. *Jurnal Infotel*, 2022, vol. 14, no. 2, pp. 154-160. doi: https://doi.org/10.20895/infotel.v14i2.753.

Received 12.12.2025
Accepted 13.02.2026
Published 03.07.2026

S. Ben Hadj Mohamed [1]*, PhD Student,*
A. Mahjoub[2]*, Associate Professor,*
C. Ben Njima[1]*, Professor,*
A. Benamor[1]*, Professor,*
[1] Electrical Department, National School of Engineers of Monastir, Monastir 5000, Tunisia.
[2] Higher Institute of Applied Sciences and Technologies of Kairoun, University of Kairouan, 3064, Tunisia,
e-mail: adel.mahjoub@isigk.rnu.tn (Corresponding Author)

How to cite this article:
Ben Hadj Mohamed S., Mahjoub A., Ben Njima C., Benamor A. A novel interception and trajectory tracking control approach for a mobile robot. *Electrical Engineering & Electromechanics*, 2026, no. 4, pp. 3-10. doi: https://doi.org/10.20998/2074-272X.2026.4.01