Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Development of an autonomous wheeled robot for efficient photovoltaic panel cleaning

Imen Hassani
University of Sfax
Chokri Rekik
chokri.rekik@enis.tn

University of Sfax
Yasser Almalaq
University of Ha'il, Ha'il 55476
Mansoor Alturki
University of Ha'il, Ha'il 55476
Abdullah Albaker
University of Ha'il, Ha'il 55476
Badr M. Alshammari
University of Ha'il, Ha'il 55476
Tawfik Guesmi
University of Ha'il, Ha'il 55476

Article

Keywords: Mobile robot, photovoltaic panels, path planning, trajectory optimization, Genetic algorithm, trajectory tracking, sliding mode algorithm

Posted Date: April 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9193009/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Development of an autonomous wheeled robot for efficient photovoltaic panel cleaning

Imen HASSANI*, Chokri REKIK*(corresponding author), Yasser Almalaq**, Mansoor Alturki**, Abdullah Albaker**, Badr M. Alshammari** and Tawfik Guesmi**
* Control and Energy Management Lab (CEM Lab) University of Sfax, Sfax Engineering School.
**Department of Electrical Engineering, College of Engineering, University of Ha'il, Ha'il 55476, Saudi Arabia.
E-mails: imenhassani@yahoo.fr; chokri.rekik@enis.rnu.tn; ya.almalaq@uoh.edu.sa; m.alturki@uoh.edu.sa; af.albaker@uoh.edu.sa; bms.alshammari@uoh.edu.sa; tawfik.guesmi@istmt.rnu.tn

***Abstract*—Cleaning tasks are widely considered one of the most relevant applications in the robotics field. This results from the substitution of human workers with robots in basic and various applications for cleaning tasks like houses, hospitals, industries, etc. Currently, cleaning photovoltaic (PV) panels are getting great attention in industry sectors because the industrial applications use the solar panel as an electrical power source. Therefore, a significant engineering challenge in a PV power generation is cleaning the PV panels, as the accumulation of dust and dirt on their surface reduces the performance of the PV system. The core concept of this work is to propose a navigation method based on the control points algorithm for wheeled mobile robots to deal with path planning. Then, a genetic algorithm is used to overcome two main problems of optimization, which are: the optimization of control points and the optimization of mobile robot trajectory, therefore, finding the optimal path. Furthermore, a robust control strategy known as sliding mode control is introduced to ensure the stabilization of an autonomous mobile robot while following a desired trajectory. Finally, simulation results are presented to demonstrate the effectiveness and applicability of the proposed method.**

***Index Terms*—Mobile robot, photovoltaic panels, path planning, trajectory optimization, Genetic algorithm, trajectory tracking, sliding mode algorithm.**

I. Introduction

Nowadays, the use of renewable energy sources, e-specially photovoltaic (PV) panels has attracted much attention. Such interest is due to the high cost of conventional energy sources and the environmental awareness linked to renewable sources. In the first place, a PV cell is defined as an electronic device that converts solar energy into electricity through the f low of electrons between two layers of semiconductors [1]. Solar panels are considered one of the cheapest sources for electricity production compared to gas, geothermal, coal, and nuclear. However, dirty PV cells affect the performance of the PV system because soiling is varies according to location [2]. Environmental factors such as temperature, wind, dust and rain greatly influence the reliability and efficiency of solar panels. That is why, PV photovoltaic panels need to be clean to allow free entrance of photons successfully and maintain the PV panels energy efficiency.

The research developed in [3] demonstrates that during long periods without rain, such as the summer, dust accumulation can lead to daily losses of over 20 % of the efficiency of photovoltaic systems. Also, the work developed in [4] demonstrates that dust accumulation on solar cells allow for a 20 % in panel efficiency. In [5], it is obvious that the efficiency of solar panels decreased by 32 % after only 6 months without any cleaning operation. Researches in [1] indicates that in about 8 weeks without cleaning, the accumulation of dirt on the surface of the panels is responsible for the drop of about 6.9 % in system performance. However, when the cleaning is performed once year, it is noted that there is a recovery of only 0.81 % on the capture of solar energy and when the cleaning procedure is performed regularly, there is a positive impact of 9.8 % on the capture.

Numerous types of cleaning methods are developed in order to solve the cleaning problems. The most-used method for removing soils in life is manual cleaning [6]. The manual cleaning method needs a human operator to clean the surface of the panels manually. Using this method is simple and easy and straight forward to solve the cleaning tasks. The main aim disadvantages of this method are that there is time required and human safety is in danger. Furthermore, manual cleaning takes a long time and may have a negative impact on PV performance. Also, cleaning tasks can solved by using the electrical method. The electric method also provides a solution to cleaning tasks [7]. However, the panels surface will always be charged, which is unsafe in the rain days. The natural powers methods are used to remove the dust, such as wind power, gravitation and, the scour of the rainwater. The work developed in [8] indicates that the effect of this method is not very favorable. Also, it is viable that the solar cell array can be turned to a vertical or oblique position to remove the dusts in the early morning, late evening or night or on a rainy day. However, the rotating the large solar cell array will be very difficult. Autonomous cleaning robots have emerged as a leading technology for cleaning solar panels.

In the literature, several sduties indicate that the development of robots has led to an incredible influence scientifically and socially for several decades [10]. Cleaning tasks are the most frequent because cleaning robots are useful and save time for people. In [1], the Solarbrush UAV Robot solves the cleaning problem. It is composed of 4 sets of propellers and a tail, which has a specific material to perform the cleaning of the panels. According to its developers, the tail is the main

advantage of this type of robot that keeps the surface of the PV panel clean. The autonomous mobile robot Ecoppia E4 is used for large rows of panels located in dry and sandy environments. This type of robot uses microfiber brushes to sweep the dust from the panels and cleaning commands or cleaning statistics can be obtained from any equipment connected to the internet. In addition, the robot has an intelligence programmed to perform cleaning according to the climatic conditions [1]. A NO Water Mechanical Automated Dusting Device (NOMADD) robot is designed specifically to clean dust accumulation from the PV panel cells of the desert without the use of water. Furthermore, this type of robot is fully automated and remotely controlled by a system of control [1]. The vacuum cleaner mobile robot is designed for solving the cleaning tasks [11]. The wheeled mobile robot has two rear wheels and one caster wheel in the front. Also, it is equipped with some sensors in order to navigate in its environment without any collision with any obstacle. This type of robot is operated by a human, which will require energy and time consumption. The Solar Cleano is adopted for cleaning solar panels [12]. The cleaning process is based on brushes and water. The mobile robot system provides brushes for dry and wet cleaning depending on the soiling grade. This type of robot can move vertically and horizontally easily.

An autonomous mobile robot for cleaning PV panels is a type of robot that is designed to move around and clean the surface of solar panels without the need for human intervention. These robots are typically equipped with sensors such as cameras, lidar, and ultrasonic sensors to navigate and avoid obstacles, as well as specialized cleaning tools, such as brushes and water jets, to remove dirt, dust, and debris from the surface of the PV panels. This type of robot can increase the efficiency of solar panels by removing dirt and debris. This can decrease the amount of sunlight that reaches the panels, and also can reduce the need for manual cleaning, which can be dangerous and time-consuming. A wide range of studies explore how to efficiently compute safe and optimal routes for mobile robots. In [**?**], a new optimal path planning algorithm based on convolutional neural network (CNN) is proposed to quickly and efficiently explore the state space. However, it only applies to sampling based algorithms and it needs to train the parameters in advance. This approach is capable of rapidly generating satisfactory solutions while reducing the likelihood of being trapped in a local optimum. However, its main limitation is that the convergence occurs too quickly, which may prevent the determination of the optimal path. Other strategies are developed in [14] to solve the cleaning tasks using mobile robot. The aim objective of the approaches developed in [14] is to determine the control points needed to construct the trajectory of the mobile robot. These points are considered key points for the path planning navigation because they make the design trajectory task very easy. The proposed algorithm offers several advantages, such as reduced computational complexity, high precision, and rapid convergence. In addition, the developed strategies are straightforward and can be easily implemented on a real prototype. Nevertheless, the main drawback is that the resulting path may not be optimal.

On the other hand, the path-tracking control problem has attracted significant attention in the field of robotics [15]. This interest is mainly driven by the widespread use of mobile robots in various applications such as transportation, inspection, security, and planetary exploration. Consequently, considerable research efforts have been devoted to addressing the trajectory-tracking problem. Among the proposed approaches, the backstepping controller has been widely investigated as an effective solution for trajectory tracking [16], [17]. This control technique is known for its robustness and relatively simple implementation in practical applications, particularly for nonlinear systems. In [18], an adaptive neural network approach is introduced to address the path-tracking problem, where the main objective of the control law is to minimize the position errors. However, this method presents certain limitations, such as the difficulty of selecting an appropriate network structure and the relatively slow convergence toward the target [19]. In addition, a proportionalintegralderivative (PID) based tracking strategy is proposed in [20]. Although this method can effectively handle the trajectory-tracking task, it may become unstable when affected by sensor sensitivity issues [21]. Other studies have explored the use of fuzzy logic controllers, as reported in [22]. While this controller can achieve satisfactory tracking performance, its architecture is relatively complex and its response time may be slow. Furthermore, the sliding mode controller has also been widely applied to trajectory tracking problems [23], [24], [25], due to its advantageous properties such as fast response, strong robustness, good transient performance, and stability.

The current manuscript presents a navigation strategy developed for mobile robots. The proposed approach is based on a control-points algorithm, where the control points are defined according to the positions of static obstacles. These points serve as key elements for path planning, as they greatly simplify the trajectory design process. Furthermore, the determined control points are optimized to obtain their most effective configuration, and the resulting navigation trajectory is also optimized. In this work, both optimization tasks are performed using a genetic algorithm. Additionally, the problem of trajectory tracking is addressed. For this purpose, a sliding mode controller is adopted to ensure that the mobile robot follows the desired trajectory accurately and reaches its target position without difficulty.

The remainder of this research is structured as follows. In section 2, the vehicle model is presented. The navigation strategy is detailed in section 3. Section 4 handles to demonstrate the trajectory tracking using sliding mode. Simulation results are given in the section 5 to prove the efficiency of the developed strategy. At last, a conclusion is presented in section 6.

II. Mobile Robot Model

In this section, the wheeled mobile robot Khepera IV [23], [24], [25] is briefly described. This robotic platform is adopted to perform the cleaning operation investigated in this study.

The robot is equipped with two independently driven wheels that allow precise maneuverability. Fig. 1 presents an overview of the Khepera IV platforms configuration.

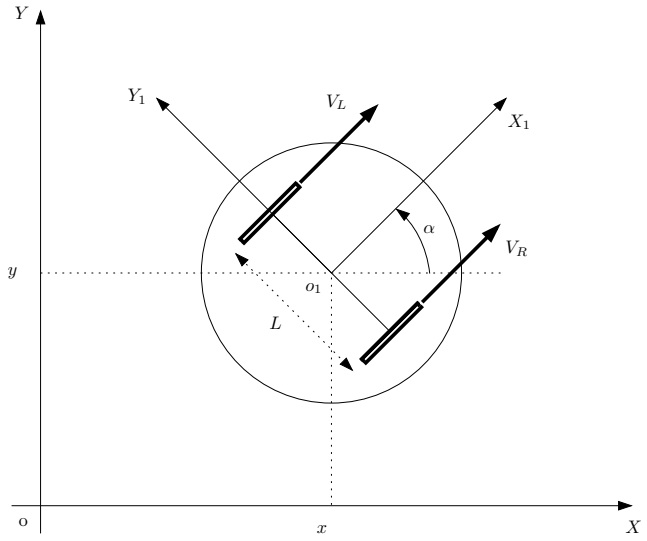


Fig. 1. Schematic model of the Khepera IV.

The motion of the nonholonomic mobile robot can be described by the following kinematic model:

$$\begin{cases} \dfrac{dx}{dt} = \dfrac{V_R + V_L}{2} \cos\alpha \\ \dfrac{dy}{dt} = \dfrac{V_R + V_L}{2} \sin\alpha \\ \dfrac{d\alpha}{dt} = \dfrac{V_R - V_L}{L} \end{cases} \tag{1}$$

In this kinematic model, V_R and V_L represent the velocities of the right and left wheels, respectively. The robot's angular velocity is given by $\frac{d\alpha}{dt}$, and L denotes the distance between the two wheels. The variable α indicates the angle between the robot's heading and the X axis.

In the present work, a wheeled mobile robot is adopted to clean solar panels. The basic cleaning operations of the photovoltaic panels using the mobile robot include trajectory identification, navigation on the surface of the solar panels and cleaning the surfaces of the photovoltaic panels. First, the mobile robot must identify the movement trajectories on the surfaces of the photovoltaic panels. Then, the robot navigates and moves safely on the surface of the solar panels. Furthermore, the robot cleans the surfaces of the solar panels in a given installation using a dedicated cleaning system. These operations are crucial and must be performed autonomously, with minimal or no human intervention.

III. PATH PLANNING ALGORITHM

The primary objective of this work is to introduce a novel strategy that employs a control points search algorithm to address the path-planning problem for robots operating in obstacle-laden environments. This approach offers the benefits of both safe navigation and minimizing the overall path length. In one hand, the proposed algorithm focused on the determination of the controls points necessary for planification. Even, these points are determined, they are optimized by the genetic algorithm. On the other hand, the obtained trajectory is also optimized by the genetic algorithm. On the other hand, the mobile robot must follow its designated trajectory while avoiding collisions with obstacles. To achieve robustness, stability, and responsiveness, a sliding mode control strategy is employed. For clarity, a summary of the main concepts and procedural steps of the proposed algorithm is presented. The following table highlights the different stages and key elements involved in this process.

Navigation Process Steps
1. Initialize control points for the trajectory.
2. Set the initial trajectory based on the control points.
3. Define the optimization function to minimize errors.
4. Start the iterative optimization process.
5. Optimize the trajectory based on control points.
6.Track the optimized trajectory in real-time.
7. The trajectory is followed and adjusted as needed.

A. Control point strategy

In mobile robot navigation, it is essential to construct a secure trajectory. This is why searching for control points can ensure the determination of an adequate trajectory. This is achieved by determining and selecting the appropriate control points. The proposed strategy is inspired by the methods developed in [14]. The main objective of the approaches developed in [14] is to determine the control points in order to construct the mobile robot trajectory. These points are considered key for path planning navigation because they simplify the trajectory design task. The advantages of the algorithm developed in [14] include its low computational cost, high precision, and fast convergence speed. However, a disadvantage of the strategies developed in [14] is that the control point searching algorithm is applied, but the obtained control points are not optimized. To overcome this disadvantage, our developed algorithm focuses on optimizing the control points and improving the work presented in [14]. For the first time, we attempt to optimize the obtained control points. Then, the planned path is optimized a second time. Furthermore, to solve the optimization problem, we use a genetic algorithm (GA).

In this research, the method for constructing the environment is inspired by the work developed in [14], while the idea of building the path for navigation is inspired by the approach presented in [26]. The work in [26] focuses on controlling the mobile robot to navigate in environments encumbered by static obstacles using the control point searching strategy. In [26], planning a suitable path depends on the evaluation of the visibility condition first. Then, based on the value of this condition, the control points are determined to facilitate the design of an adequate path. Suitable geometric methods, such as triangulation and polygonization, are adopted in [26]. The choice of this contribution is justified by its flexibility

in solving path planning problems for mobile robots. Additionally, this method reduces the complexity of navigation and simplifies the planning of the robot's trajectory. The simplicity of the method also makes it easier for researchers to apply this strategy in their own work. Furthermore, the implementation of this approach on a real prototype is straightforward.

In the developed work, a control point is defined as the midpoint of obstacles that are located on the face of the mobile robot [24]. Control points are determined directly using a simple geometric method. To ensure the correct locations of these points, Dijkstra's algorithm is adopted. Firstly, the Dijkstra algorithm is one of the best methods to estimate the minimum path to reach a destination point in a road network. It is a solution to the single-source shortest path problem in graph theory. Secondly, Dijkstra's algorithm finds the shortest path between two nodes, using a single node as the "source" node and computing the shortest paths from the source to all other nodes in the graph, resulting in a shortest path tree [27] (see Algorithm 1). Furthermore, Dijkstra's algorithm uses nodes in its calculations and stores points where the cost of distance is low. It continues its calculations to find the shortest path starting from the initial point, then excludes routes with high distance costs when making updates [28]. The calculation of paths through neighboring nodes is considered temporary if they have not yet been finalized. If the calculated distance of a node is smaller than the current one, Dijkstra will update its calculations. Nodes that have been selected as having the shortest distance from another node will be marked as permanent, ensuring they are stored in the central part of the algorithm [28].

Algorithm 1: Pseudo code for Dijkstra's Algorithm
1. function Dijkstra (Graph, source):
2.dist[source]:=0//Distance from source to source
3. for each vertex v in Graph://Initialisation
4. if $v \neq$ source
5. dist[v]:=infinity//Unknown distance function
6.previous[v]:=undefined//Previous node in path
7. end if//from source

In this research, the first step of the developed algorithm is to determine all control points. Then, the points that give the shortest path are considered as control points for navigation, and the other points are ignored. The number of these points is denoted as n_c. This is done to build the shortest path. Mathematically, control points for planning are the points that give the minimum distance d_{min} between the robot's position R, control points P_{cj} ($j = 1, 2, \ldots, n_c$), and the goal position G. This is done to build the shortest path quickly. The objective of the second step is to use the Dijkstra algorithm to facilitate the location of control points necessary for navigation. This helps reduce the complexity of the navigation and facilitates the obtaining of a suitable trajectory.

Even though the control points are determined, the obtained path is the shortest one, but it is not the optimal path. For this reason, in the next section, we aim to optimize the obtained control points and trajectory for the robot using the genetic algorithm.

B. Genetic Algorithm Approach

Firstly, this algorithm is used to optimize the determined control points. Secondly, in order to drive an autonomous mobile robot from any position to a desired position with an optimal trajectory, the path designed by the optimized control points is also optimized using the genetic algorithm.

A genetic algorithm is a method for solving optimization and search problems that is inspired by the process of natural selection. The basic idea is to start with a population of solutions (often represented as a set of individual "chromosomes" or "genes"), and then repeatedly apply a set of genetic operators (such as selection, crossover, and mutation) to the population to generate new solutions.

Generally, a typical genetic algorithm consists of the following steps:

a) Initialization: Create an initial pool of candidate solutions, usually generated at random. The algorithm begins by producing a randomly initialized population, referred to as Pop, composed of m chromosomes. The structure of this population can be described as follows:

$$Pop = \begin{bmatrix} Chromosome_1 \\ . \\ . \\ . \\ chromosome_m \end{bmatrix} \tag{2}$$

Each chromosome in the population, represented as $Chromosome_i$, corresponds to a candidate solution of the optimization problem, where $i = 1..., m$.

b) Selection: This step aims to select a subset of the fittest solutions to be used in the next generation. This is achieved by evaluating the fitness of each solution in the population using a fitness function that assigns a numerical value to each solution based on how well it solves the problem at hand.

c) Crossover: The crossover operator combines the genetic information of individuals in the population to generate new individuals with better performance than their parents. Each individual is encoded as a string of symbols belonging to an alphabet. The full string contains global information provided by each string element or by a set of elements, also called substrings. Each substring represents a specific feature of the individual.

d) Mutation: The mutation step aims to introduce small random changes to the genetic information of the new solutions, thereby increasing diversity within the population. This step is repeated iteratively until the best solution is obtained.

1) Optimization of Control Points: The fundamental approach to solving an optimization problem using GA is to start with a randomly initialized population of individuals, each representing the coordinates of the control points P_{cj}=(x_{pj}, y_{pj}) ($j = 1 : n_c$).

The algorithm begins with the generation of an initial population called Pop, which is randomly produced and contains m chromosomes, where $m = 10$. The organization of this population is described in Equation (2). Each element of this population, noted as $Chromosome_i$, corresponds to a possible solution of the optimization problem. The representation of an individual chromosome $Chromosome_i$ is defined as follows:

$$Chromosome_i = \underbrace{P_{i1} \dots P_{im}}_{P_{cj}} \tag{3}$$

with $i = 1, ..., m$.

Then, we move to the selection operation, which takes into consideration the fitness of each individual. For this reason, the fitness function is adopted in this algorithm to select the best individuals from the control points P_{cj} that yield the optimal trajectory for navigation. The expression for the fitness function is defined in equation 4:

$$J_c = \sum J_{ci} \tag{4}$$

J_{ci} is expressed as shown in equation 5.

$$J_{ci} = \frac{1}{2} \int (x_{p(i+1)} - x_{p(i)})^2 + (y_{p(i+1)} - y_{p(i)})^2 dt \tag{5}$$

with J_{ci}, describing the distance joining two successive control points.

Once the selection operation is completed, we proceed to the crossover and mutation operators to discover new individuals with better performance than their parents. This step is repeated iteratively until the best solution is obtained.

Once the control points are optimized, we determine the trajectory of the mobile robot. Then, we proceed to optimize the obtained trajectory using the genetic algorithm.

2) Optimization of Trajectory for Navigation: This section addresses the problem of optimizing the trajectory for mobile robot navigation, with the goal of achieving navigation along the optimal path.

The first step of the genetic algorithm is to define the initial population, denoted as Pop, which represents the configuration of the mobile robot (x, y). The population consists of m chromosomes, with $m = 10$. Its overall representation is described in Equation (2).

Each chromosome in this population, indicated by $Chromosome_i$, corresponds to a candidate solution to the optimization problem. The internal structure of every chromosome $Chromomose_i$ can be expressed as follows:

$$Chromosome_i = \underbrace{x_{i1} \dots x_{im}}_{x} \underbrace{y_{i1} \dots y_{im}}_{y} \tag{6}$$

Where $i = 1, ..., m$.

The selection operation takes into consideration the fitness of every individual. The fitness function J is adopted in this algorithm in order to select the best individuals that give the minimum distance d_{min} between the current position of robot and its desired position. Furthermore, the fitness function is defined as shown in equation 7.

$$J = \int d_{min}^2 dt \tag{7}$$

This procedure is performed repeatedly through successive iterations until the optimal individual is obtained.

Next, we proceed to the crossover and mutation operators to generate new individuals with better performance than their parents. This step is repeated iteratively until the best solution is obtained. However, the mobile robot must be controlled by a control system to help it track the planned trajectory. For this reason, a sliding mode controller is presented in the next section to control the robot.

IV. Sliding Mode Control

A Sliding Mode Controller (SMC) is a type of nonlinear control strategy known for its robustness, simplicity, and ability to handle system uncertainties and disturbances. It is widely used in robotics, power electronics, automotive systems, and other applications where high reliability is required. Also, the sliding mode controller is a robust control method used in systems that are subject to uncertainties, external disturbances, or model nonlinearities. It is especially useful for systems that require high precision and stability under challenging conditions, such as mobile robots, due to its ability to handle uncertainties and maintain system performance despite external disturbances.

The principle of sliding mode control consists in forcing the system states to converge toward a chosen sliding surface and maintaining them on this surface using a switching control law [29]. This control technique is selected because of its effectiveness and robustness in addressing trajectory tracking tasks for wheeled mobile robots. Moreover, it has proven to be an effective solution for various tracking control challenges [24], [25] and [30].

The proposed control law offers several important advantages, including strong robustness, fast response, and high tracking accuracy. The implementation of this control approach requires the definition of two fundamental configurations: the reference configuration and the current configuration of the robot (see Fig. 2). The reference configuration, denoted by $p_r = (x_r, y_r, \alpha_r)$, represents the desired pose that the robot must follow. In contrast, the robot's actual pose is expressed as $p = (x, y, \alpha)$.

The difference between the reference pose p_r and the current pose p defines the tracking error, denoted by $p_e = (x_e, y_e, \alpha_e)$. The mathematical formulation of p_e is provided in Equation (8) as follows:

$$p_e = \begin{bmatrix} x_e \\ y_e \\ \alpha_e \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \alpha_r - \alpha \end{bmatrix} \tag{8}$$

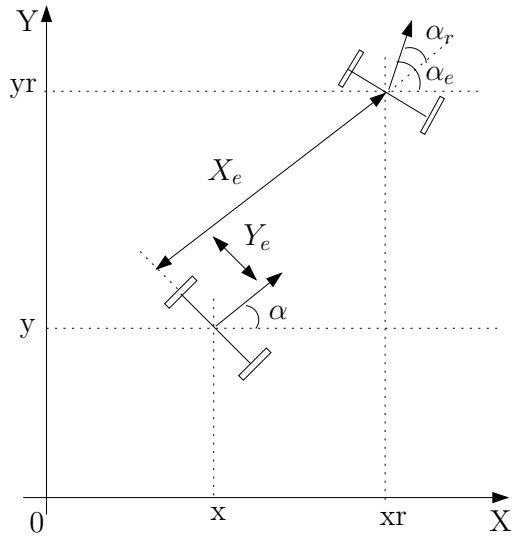


Fig. 2. Tracking error.

Moreover, addressing the trajectory tracking task requires determining the appropriate control vector $q = (v, w)^T$, where v represents the linear velocity of the wheeled mobile robot, and w denotes its angular velocity. The corresponding expressions for the right and left wheel velocities, V_R and V_L, are provided in equation (9).

$$\begin{cases} V_R = v + \dfrac{L.w}{2} \\ V_L = v - \dfrac{L.w}{2} \end{cases} \tag{9}$$

To assess the stability of the control system, a Lyapunov candidate function is introduced, as presented in equation 10. Additionally, the first switching function is set to $x_e = 0$ to facilitate the solution of the problem [29].

$$L = \frac{1}{2} y_e^2 \tag{10}$$

The Lyapunov function L can be derivative as follows:

$$\begin{aligned} \dot{L} &= y_e \dot{y_e} = y_e(-x_e w + v_r \sin(\alpha_e)) \\ &= -x_e y_e w - v_r y_e \sin(\arctan(v_r y_e)). \end{aligned}$$

From the previous result, it is evident that $\dot{V} \leq 0$ since $v_r y_e \sin(\arctan(v_r y_e)) \geq 0$. To enhance the performance of the sliding mode controller, we introduce $\alpha_e = -\arctan(v_r y_e)$ as a candidate switching function. Furthermore, the switching surface vector s for the mobile robot is defined as in equation 11 [29], and the corresponding saturation function $sat(s)$ is provided in equation 12 [29].

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_e \\ \alpha_e + \arctan(v_r y_e) \end{bmatrix} \tag{11}$$

$$sat(s) = \begin{cases} \dfrac{s}{\xi} & \text{if } \dfrac{s}{\xi} \leq 1 \\ sat(s) & \text{if } \dfrac{s}{\xi} > 1 \end{cases} \tag{12}$$

Here, ξ is a positive constant that determines the boundary layer width of the saturation function.

Subsequently, the switching function is defined using this saturation function, as illustrated in equation 13.

$$\dot{s} = -k sat(s). \tag{13}$$

Sliding surface vector is defined as follows:

$$\begin{aligned} \dot{s} &= \begin{bmatrix} \dot{x_e} \\ \dot{\alpha_e} + \dfrac{\partial \gamma}{v_r} \dot{v_r} + \dfrac{\partial \gamma}{y_e} \dot{y_e} \end{bmatrix} \\ &= \begin{bmatrix} y_e w + v_r \cos \alpha_e - v \\ w_r + \dfrac{\partial \gamma}{\partial v} \dot{v} + \dfrac{\partial \gamma}{\partial y_e}(v_r \sin \alpha_e - x_e w) - w \end{bmatrix} \end{aligned} \tag{14}$$

Finally, the resulting control law is [29]:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} y_e w + v_r \cos \alpha_e + k_1 sat(s_1) \\ w_r + \dfrac{\partial \gamma}{\partial v_r} \dot{v_r} + \dfrac{\partial \gamma}{\partial y_e}(v_r \sin \alpha_e) + k_2 sat(s_2) \end{bmatrix} \tag{15}$$

where $\gamma = \arctan(v_r y_e)$; $w_r = \dot{L}$; $v_r = 1$
$\dfrac{\partial \gamma}{dv_r} = \dfrac{y_e}{1 + (v_r y_e)^2}$ and $\dfrac{\partial \gamma}{dy_e} = \dfrac{v_r}{1 + (v_r y_e)^2}$.

V. Simulation Results and Discussion

This section provides the simulation results used to assess the effectiveness of the proposed optimization and control strategies for mobile robot navigation. The simulations were designed to assess the effectiveness of the control law and the trajectory planning system under various conditions, including obstacle avoidance,path efficiency, and response to static environments. In the mobile robot field, designing a suitable path for navigation is essential. Therefore, demonstrating the validity of the developed algorithm is a very important issue in robotics. For this reason, this section is devoted to illustrating the effectiveness of the proposed strategy. To further confirm the applicability of this method, several simulation tests have been carried out in different environments. In this research, three navigation environments are considered (see Fig.4,Fig.7 and Fig.10).

In all simulation results, the initial robot position is (x_r, y_r)=(0, 0) and the target position is (x_T, y_T)=(29, 29). These results were simulated using MATLAB software.

A. robot path planning

In the Fig.3, environment 1 for navigation is constructed by placing obstacles. It is evident that the control points algorithm has been applied, and the control points are determined. In this case, the number of control points is $n_c = 15$. This facilitates the planning of the navigation trajectory and the mobile robot navigates without any collision problems.

Once the control points are determined, the next step is to optimize these points using the genetic algorithm. As illustrated in Fig. 4, the genetic algorithm is implemented, and the resulting navigation trajectory (colored red) is planned while considering the optimized control points. Based on the results presented in Fig.4, it is clear that the navigation path has been improved.

Finally, the Fig.5 shows the optimized navigation trajectory (colored green) obtained using the genetic algorithm.

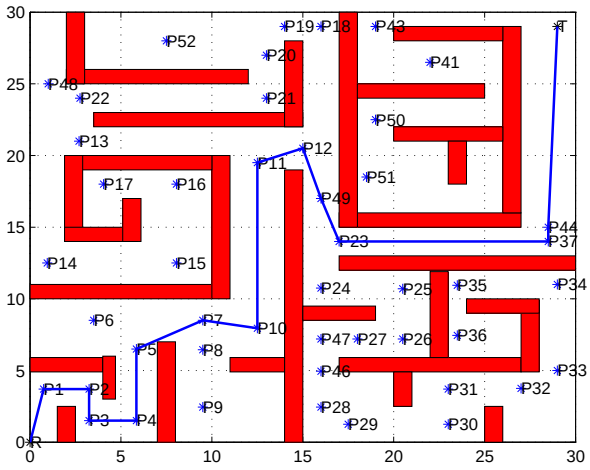


Fig. 3. Environment 1:placement of control points.

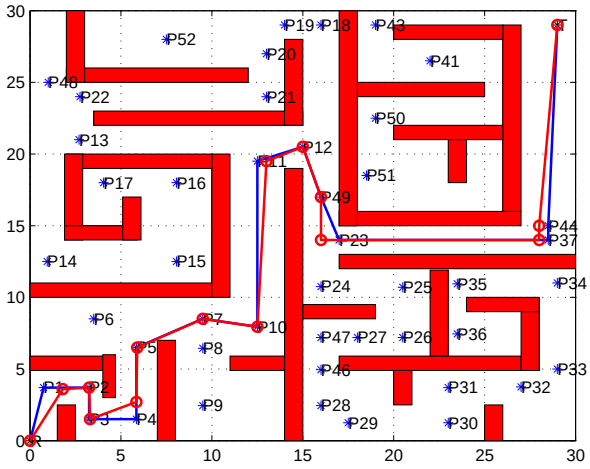


Fig. 4. Optimization of control points.

To provide a quantitative evaluation, the relative error between the planned and optimized trajectories is calculated. Table 1 presents the relative errors for each control point:

The results in Table III show that the genetic algorithm effectively reduces deviations, confirming its ability to optimize the robots navigation trajectory. As shown in Table III,

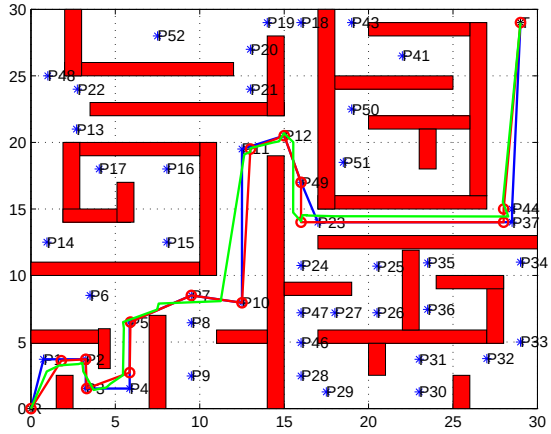


Fig. 5. Optimization of trajectory for navigation.

TABLE I
RELATIVE ERROR BETWEEN PLANNED AND OPTIMIZED CONTROL POINTS.

Planned Position	Optimized Position	Relative Error (%)
(0, 0)	(0, 0)	0.00
(1, 3)	(1, 3)	0.00
(3, 3)	(3, 3)	0.00
(3, 1)	(3, 1)	0.00
(6, 2)	(6, 2.5)	25.00
(6, 6)	(6, 6)	0.00
(9.8, 9)	(9.8, 9)	0.00
(14, 8)	(14, 8)	0.00
(13, 20)	(13.3, 20)	2.31
(16, 21)	(16, 21)	0.00
(16, 17)	(16, 17)	0.00
(17, 14)	(16, 14)	5.88
(27.5, 14)	(27, 14)	1.82
(27.5, 15)	(27, 15)	1.82
(29, 29)	(29, 29)	0.00

the optimization process has successfully minimized the deviations between the planned and optimized control points. The majority of the control points exhibit zero relative error, which indicates that the optimized positions are identical to the planned ones, demonstrating the algorithm's precision. Notable points where the relative error is zero include control points 1, 2, 3, 4, 6, 7, 9, 10, and 15, all of which have been perfectly optimized. These results suggest that the optimization technique has effectively minimized the error for these particular points. However, some control points exhibit small relative errors. For example, control point 5 has a relative error of25%, while control points 13 and 14 show small errors of 5.88% and 1.82%, respectively. These deviations may be attributed to various factors such as the constraints of the optimization process, the complexity of the system being optimized, or the limits of numerical precision in the algorithm. Despite these small deviations, the overall results still indicate a significant improvement in the trajectory optimization.

Even the environment 2 of navigation is presented (see Fig.6), the mobile robot navigates without collision with problem. In this simulation, 13 control points are obtained and

the path is planned using this points. Then, to ameliorate the path of navigation for the mobile robot, the genetic algorithm is applied in order to optimize the control points and the trajectory for navigation. As shown in the Fig.7, the trajectory of navigation (red color) is ameliorated by optimizing the determined control points. The result presented in the Fig.8 demonstrates that the genetic algorithm is an effective solution to solve the problem of optimization.

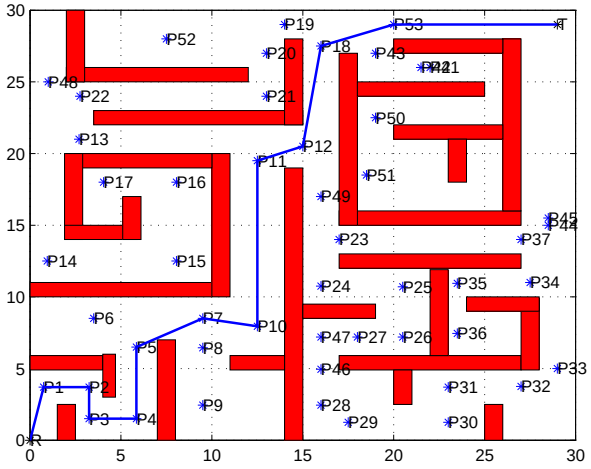


Fig. 6. Environment 2:placement of control points.

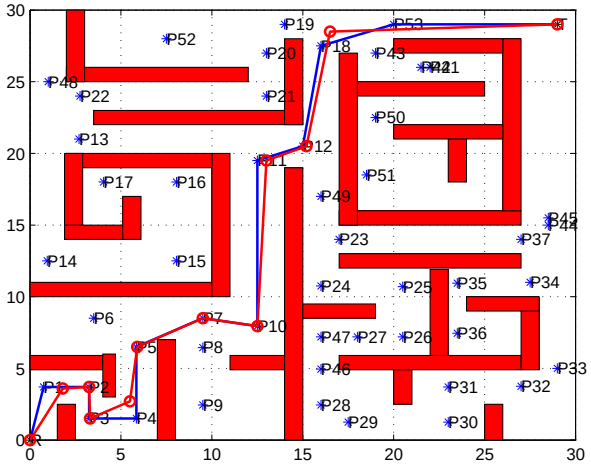


Fig. 7. Optimization of control points.

To provide a quantitative evaluation, the relative error between the planned and optimized trajectories is calculated. Table 2 presents the relative errors for each control point:

In this study, the relative errors between the planned and optimized trajectories for each control point were calculated to assess the accuracy of the optimization process. As shown in Table 2, the majority of the control points exhibit minimal

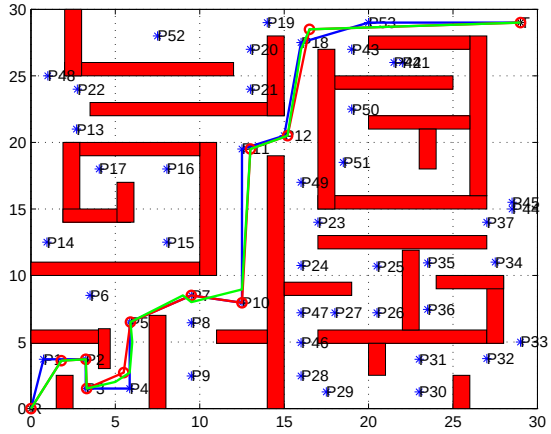


Fig. 8. Optimization of trajectory for navigation.

TABLE II
RELATIVE ERROR BETWEEN PLANNED AND OPTIMIZED CONTROL POINTS.

Planned Position	Optimized Position	Relative Error (%)
(0, 0)	(0, 0)	0.00
(1, 4)	(2, 4)	20.00
(3.5, 4)	(3.5, 4)	0.00
(3.5, 1.5)	(3.5, 1.5)	0.00
(6.5, 1)	(6, 3)	20.00
(6.5, 6)	(6.5, 6)	0.00
(9.8, 8)	(9.8, 8)	0.00
(14, 7)	(14, 7)	0.00
(13.8, 19)	(14, 19)	1.45
(16, 20)	(16, 20)	0.00
(16, 27.5)	(17, 28)	3.08
(20, 28)	(20, 28)	0.00
(29, 29)	(29, 29)	0.00

relative error, indicating that the optimization technique successfully refined the trajectory with high precision. Notably, control points 1, 3, 4, 6, 7, 8, 10, 12 and 13 display zero relative error, meaning that the optimized positions perfectly match the planned positions. Control points 2, 5, 9, and 11 show small errors, which suggest that while the optimization process has improved the trajectory, slight deviations remain. These deviations could be due to factors such as numerical precision, constraints imposed during optimization, or the inherent complexity of the system being optimized.

The third simulation environment is presented to demonstrate the validity of the proposed algorithm (see Figs.9, 10, and 11). As illustrated in these figures, the suitable path (shown in green) is successfully generated, and the mobile robot is able to navigate through the environment safely.

From the simulation results, it is evident that the proposed strategy performs effectively, allowing the robot to successfully avoid obstacles under different scenarios.

The results presented in Table 3 highlight the relative errors between the planned and optimized positions for each control point.

The majority of the control points exhibit a relative error of 0%, indicating that the optimization process has perfectly

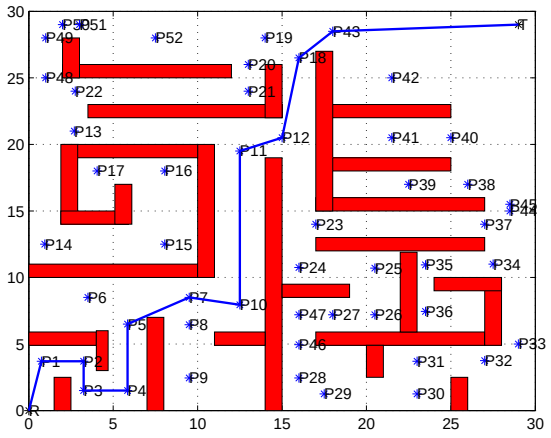


Fig. 9. Environment 3:placement of control points.

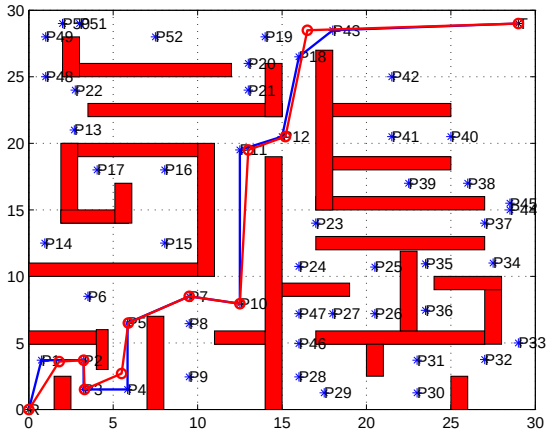


Fig. 10. Optimization of control points.

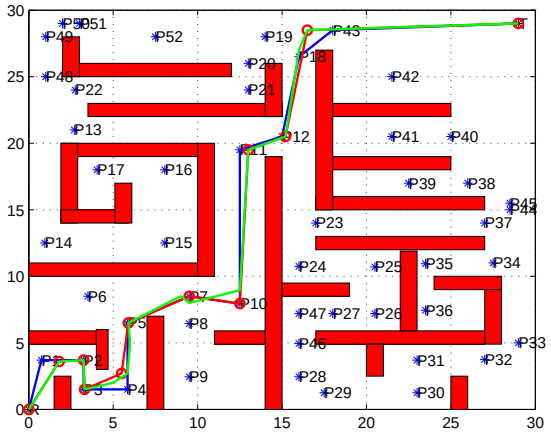


Fig. 11. Optimization of trajectory for navigation.

TABLE III
RELATIVE ERROR BETWEEN PLANNED AND OPTIMIZED CONTROL POINTS.

Planned Position	Optimized Position	Relative Error(%)
(0, 0)	(0, 0)	0.00
(1, 4)	(2, 4)	24.25
(3.5, 4)	(3.5, 4)	0.00
(3.5, 1.5)	(3.5, 1.5)	0.00
(6, 1)	(6, 3)	32.57
(6, 6)	(6, 6)	0.00
(10, 8)	(9.8, 8)	1.56
(13.5, 7.5)	(13.5, 7.5)	0.00
(13.8, 20)	(14, 19)	2.66
(16, 20)	(16, 20)	0.00
(16, 26)	(16, 28)	6.35
(18, 28)	(18, 28)	0.00
(29, 29)	(29, 29)	0.00

matched the planned and optimized positions for those points. This suggests that the optimization algorithm was highly successful in adjusting the trajectory for the majority of the control points.

However, some control points show small deviations, with Control Points 2, 5, 7, 9 and 11 exhibiting relative errors. These minor discrepancies can likely be attributed to factors such as numerical precision, constraints within the optimization process, or the inherent complexity of the system being optimized.

Overall, the optimization approach proves to be effective, with only slight deviations from the planned trajectory in a few instances. Further refinements could focus on reducing these small errors for specific points, ensuring higher accuracy across the entire trajectory. The results suggest that the optimization method provides an effective way to improve trajectory planning, with minimal trade-off in terms of accuracy.

B. Tracking Trajectory Results

Several simulation results are presented to illustrate the effectiveness of the proposed control system in ensuring accurate trajectory tracking.

Fig.12 illustrates that the mobile robot can follow the planned trajectory and reach the target positions without difficulty, while Fig.13 shows that all position errors converge to zero.

When the navigation environment is modified (see Fig.14), the results indicate that the robot continues to track the desired trajectory successfully, with all position errors approaching zero (see Fig.15).

Figs.16 and 17 further highlight the performance of the control law. It is evident that the trajectory tracking task is accomplished with satisfactory accuracy.

The results of these simulations clearly indicate that sliding mode control is an effective solution for the trajectory tracking problem, demonstrating strong performance and reliability

In comparison with the study reported in [25], the simulation results obtained in this work demonstrate superior performance. The main benefit of the proposed approach lies in its capability to optimize the navigation path using a genetic algorithm. Moreover, the algorithm shows higher

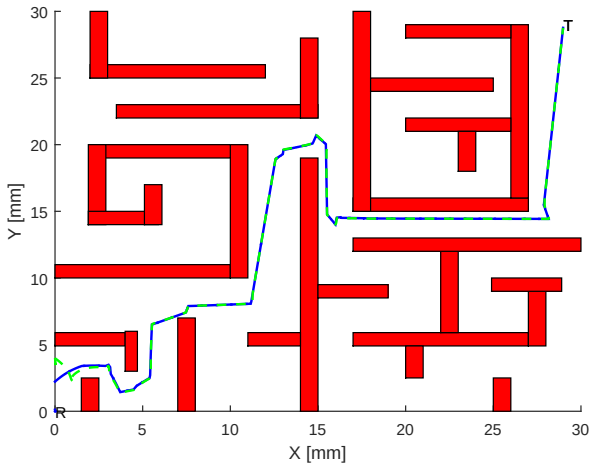


Fig. 12. Tracking trajectory.

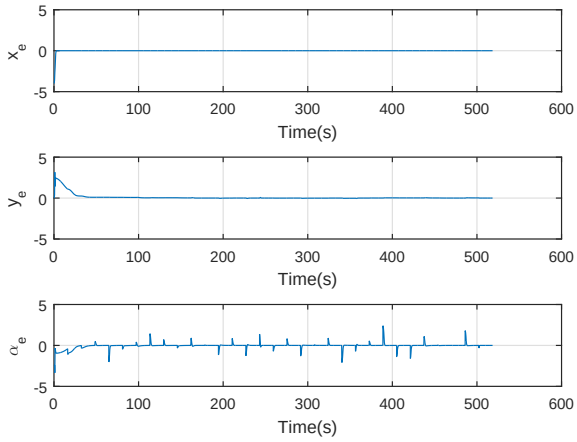


Fig. 13. Error positions.

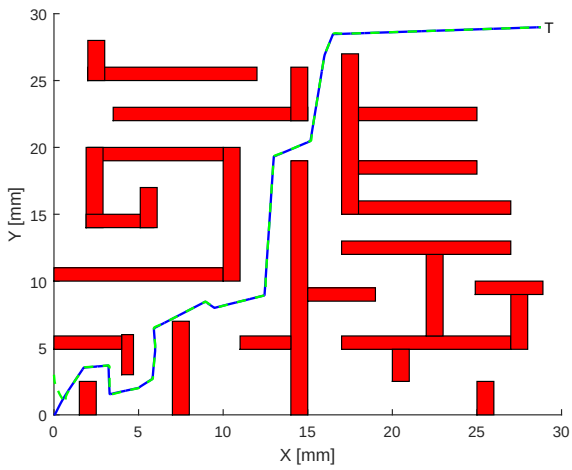


Fig. 14. Tracking trajectory.

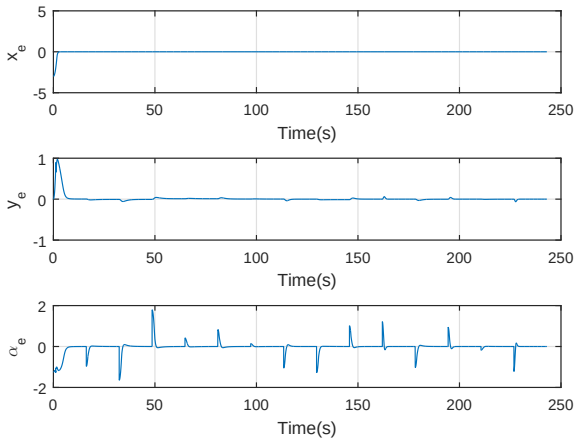


Fig. 15. Error positions.

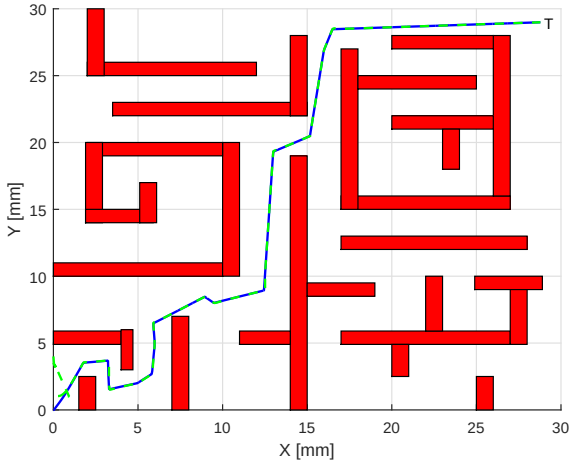


Fig. 16. Tracking trajectory.

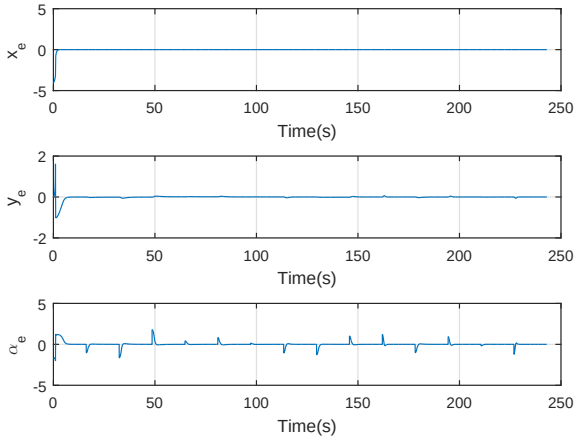


Fig. 17. Error positions.

efficiency than the previous method, as it achieves shorter computation times under various conditions. Additionally, the strategies developed here are simple, effective, and can be easily implemented on a real prototype.

The results and discussion in this paper highlight the effective performance of the proposed navigation and trajectory tracking algorithms for mobile robots. Simulation results from various scenarios clearly demonstrate that the developed strategy enables collision-free navigation in environments with static obstacles. The path optimization achieved using the genetic algorithm (GA) significantly enhances the trajectory, while the sliding mode controller ensures accurate path tracking. Furthermore, the proposed approach outperforms previous solutions in terms of both efficiency and trajectory optimization. This work addresses the navigation problem of mobile robots by developing a control points searching algorithm. The primary objective of this algorithm is to find a suitable path for navigation, ensuring that no collisions occur. To identify these control points, the Dijkstra algorithm is employed. For optimization of the path, a genetic algorithm is applied to find the optimal trajectory. Furthermore, the trajectory tracking task is addressed using a control system. These methods have demonstrated simplicity and flexibility in handling path planning problems, including obstacle avoidance. Additionally, the results emphasize the high performance of the sliding mode controller, as the robot is able to track the target trajectory with minimal tracking error.

In the future research, a real prototype would be used to simulate the development algorithms in the real time, where obstacles and conditions change over time.

Authors' contribution: I.H and C.R. designed the problem under study, performed the simulations, and obtained the results; Y.A, M.A, A.A, BM.A and T.G. analyzed the obtained results; I.H, C.R, Y.A, M.A, A.A, BM.A and T.G. wrote the paper, which was further reviewed. All authors have read and agreed to the published version of the manuscript.

Funding: No funding is provided for the preparation of manuscript.

Data Availability: The datasets used and analysed during the current study available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interests.

Acknowledgments: There is no acknowledgment involved in this work.

REFERENCES

[1] Y. Wang and H. Li, "High-Precision Path Planning for Robotic Cleaning of Large Solar Arrays", Journal of Solar Energy Engineering", 147(2), 11002-11012, 2025.

[2] L. Nguyen, "Multi-Robot Systems for Photovoltaic Panel Cleaning: Cooperative Path Planning and Coordination", IEEE Transactions on Robotics, 41(2), 235-249, 2025.

[3] S. Christian, B. Stefan and K. Michael, "Impact of Soiling on Iv-Curves and Efficiency of PV-Modules", Journal of Solar Energy, 112, 259-262, 2015.

[4] A. Chellal, J. lina, A. I. Pereira, P. Costa," Innovative Robot Design for Cleaning Solar Panels", In Proceedings of the 11th International Conference on Simulation and Modeling Methodologies, 264-270, 2021.

[5] M. Mostefaoui, A. Ziane, A. Bouraiou and S. Khelifi, "Effect of Sand Dust Accumulation on Photovoltaic Performance in The Sharan Environment", Environmental Science and Pollution Research, 26(1), 259-268, 2019.

[6] P. Miller and R. White, "Development of an Autonomous Mobile Robot for Solar Panel Cleaning: Path Planning and Obstacle Avoidance", Journal of Field Robotics, 43(2), 215-229, 2025.

[7] M. Al-Housani, Y. Bicer and M. Koc, "Assessment of Various Dry Photovoltaic Cleaning Techniques and Frequencies on the Power Output of CdTe-Type Modules in Dusty Environments, Journal of Sustainability", 1-18, 2019.

[8] A. Joon and W. Kowalczyk, "Design of Autonoums Mobile robot for Cleaning The Environment With Obstacles", Journal of Applied Sciences, 11(17), 1-13, 2021.

[9] G. He, C. Zhou and Z. Li, "Review of Self-Cleaning Method for Solar Cell Array", International Workshop on Automobile, Power and Energy Engineering", 16(2011), 640-645, 2011.

[10] J. Kim, A. Ziane, A. K. Mishra, R. L. M. Scafuro, N. Cauli, J. Santos-Victor, B. Mazzolai and F. Cavalo, "Control Strategies for Cleaning Robots in Domestic Applications: A comprehensive review", International Journal of Advanced Robotic Systems, 1-21, 2019.

[11] J. Smith and A. Doe, "Mobile Robotic System for Solar Panel Cleaning: A Review", International Journal of Robotics and Automation, 35(2), 112-130, 2021.

[12] R. Zahedi, P. Ranjbaran, G. B. Gharehpetian, F. Mohammadi and R. Ahmadiahangar, "Cleaning of Floating Photovoltaic Systems: A Critical Review on Approaches from Technical and Economic Perspectives", Jouranl of Energies, 1-25, 2021.

[13] I. Hassani, I. Maalej and C. Rekik, "Control Points Searching Algorithm for Multiple Mobile Robots", International Journal Modelling, Identification and Control (IJMIC), 41(1/2), 120-132, 2022.

[14] B. Hao, H. Du, X. Dai and H. Liang, "Automatic Recharging Path Planning for Cleaning Robots",Journal of Mobile Information Systems, 30(3), 1-19, 2021.

[15] I. Hassani, I. Ergui and C. Rekik, "Turning Point and Free Segments Strategies for Navigation of Wheeled Mobile Robot", International Journal of Robotics and Control Systems, 2(1), 172-186, 2022.

[16] I. Hassani and C. Rekik, "Backstepping Controller for Mobile Robot in Presence of Disturbances and Uncertainties", International Journal of Robotics and Control Systems, 3(4), 934-954, 2023.

[17] S. kumar and Singh "Sliding Mode Control in Nonholonomic Mobile Robots: Theory and Applications" , Journal of Robotics and Automation, 2-5, 207-219,2025.

[18] J. A. Abougarair, K. I. M. Aburakhis and M. M. Edardar, "Adaptive Neural Networks Based Robust Output Feedback Controllers for Nonlinear Systems", International Journal of Robotics and Control Systems, 2(1), 37-56, 2022.

[19] Y. Koubaa, M. Boukattaya and T. Damak, "Intelligent Control for An Uncertain Mobile Robot with External Disturbances Estimator", New Trends in Robot Control, 155-176, 2020.

[20] C. B. Jabeur and H. Seddik, "Optimized Neural Networks-PID Controller with Wind Rejection Strategy for a Quad-Rotor", Journal of Robotics and Control (JRC), 3(1). 62-72, 2022.

[21] L. T. Anh, T. T. T. Nga and V. V. Hoc, "PID-Type Iterative Learning Control for Output Tracking Gearing Transmission Systems", International Journal of Robotics and Control Systems, 1(3), 256-268, 2021.

[22] S. Sahloul, D. Ben Halima and C. Rekik, "An Hybridization of Global-Local Methods for Autonomous Mobile Robot Navigation in Partially-Known Environments", Journal of Robotics and Control (JRC), 2(4), 351-1355, 2021.

[23] I. Hassani, I. Maalej and C. Rekik, "Robot Path Planning with Avoiding Obstacles Using Free Segments and Turning Points Algorithms", Mathematical problem in Engineering, 2018(6), 1-13, 2018.

[24] I. Hassani, I. Maalej and C. Rekik, "Control Points Searching Algorithm for Autonomous Mobile Robot Navigation with Obstacle Avoidance", In the 19th International Conference on Sciences and Techniques of Automatic control and computer engineering (STA'19), Sousse, Tunisia, 158-163, 2019.

[25] A. Santos and Almeida, "Sliding Mode Controllers for Robust Path Tracking in Mobile Robots", Journal of Control and Optimization, 38-4, 45-56, 2025.

[26] J. A. Vazquez-Santacruz, M. Velasco-Villa, R. d. J. Portillo-Velez, L. F. Marin-Urias, M. Vigueras-Zuniga, “Autonomous navigation for multiple mobile robots under time delay in communication”, International Journal of Robot and System, 30(3), 289-298, 2017.

[27] A. Alyasin, E. I. Abbas and S. D. Hasan, “An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method”, in the 4th Scientific International Conference, Najaf-IRAQ, 1-14, 2019.

[28] L. Parungao, F. Hein and W. Lim, “Dijkstra Algorithm Based Intelligent Path Planning With Topological Map and Wireless Communication”, ARPN Journal of Engineering and Applied Sciences, 2753-2763, 2018.

[29] J. H. Lee, C. Lin, H. Lim and J. M. Lee, “Sliding Mode Control for Trajectory Tracking of Mobile Robot in the RFID Sensor Space”, International Journal of Control, Automation, and Systems, 7(3), 429-435, 2009.

[30] Z. Liu, and J. Zhang, “Advances in Sliding Mode Control for Nonlinear Systems: A Comprehensive Review”, International Journal of Control and Automation, 50(3), 345-367, 2025.