Journal of Robotics and Control (JRC)
Volume 3, Issue 2, March 2022
ISSN: 2715-5072, DOI: 10.18196/jrc.v3i2.13765

Mobile Robot Navigation Using Planning Algorithm and Sliding Mode Control in a Cluttered Environment

Islem Reguii [1], Imen Hassani [2], Chokri Rekik [3]
[1,2,3] Control and Energy Management Laboratory (CEM Lab), University of Sfax, National Engineering School of Sfax, Tunisia
Email: [1]islemergui@yahoo.fr, [2]imenhassani@yahoo.fr, [3]chokri.rekik@enis.rnu.tn

***Abstract*—The research contribution of the present work is to solve the path planning and path tracking problems in static and dynamic environments. A new Planning Navigation Algorithm Technique is developed in order to solve the problem of navigation with obstacle avoidance. The basic idea of this algorithm searches for a safe path for navigation. First, this algorithm is focused to identify an optimal collision-free route to a spatially defined objective. Then, in each displacement, the developed algorithm handles to maximize the distance between the obstacles and minimize the distance to the goal. This is to obtain the optimal trajectory for navigation. On the other side, a sliding mode controller is adopted to solve the tracking trajectory task. The basic idea of this control system is to allow the robot mobile to track the desired trajectory with minimum error. In addition, the comparative study between the proposed approach and the previous work is presented in order to demonstrate the satisfaction of the proposed strategy. Finally, simulation results which are developed using Matlab software are presented to show the robustness and efficiency of the developed algorithm and the reactivity of the proposed sliding mode controller.**

***Index Terms*—Mobile Robot; Planning Algorithm; Navigation; Obstacle Avoidance; Sliding Mode**

I. Introduction

Mobile robots are widely used in various fields such as space exploration, security, industrial environments, medicine, disaster relief, etc. It is a very diverse field of research, mainly due to the many potential applications. In order to perform tasks in any of these applications, a robot must have the ability to navigate its environment. One of the key issues in the robotic community is the navigation of a mobile robot. That's why path planning is an an important problem in autonomous robotics.

The most important guidelines for the best possible path are the smoothness of the path, shortest distance, and minimum energy consumption. So, the most adopted thing is to obtain the shortest path with the minimal possible time. If we want to define the planning of the trajectory of a mobile robot, we must guarantee secure navigation of the robot to a known target. Therefore, one of the main lines of research in modern robotics today is the search for planning algorithms and control strategies for non-holonomic systems.

In the literature, they are two categories of path planning: local path planning which is the robot has limited information about the environment, and global path planning which is the robot has complete knowledge about the environment [1], [2], [3]. For these two types of environment, many methods have indeed been applied for a solution to global navigation problems, notably Grids [3], B-spline technique [4], artificial potential field [5]. On another side, the mobile robot controls its motion by sensors in the local navigation.

When the mobile robot has a priori knowledge of the environment, many works are used the hierarchical fuzzy logic controller (HFLC) to solve the navigation of an autonomous mobile robot in an unknown environment [6], [7]. In [8], [9], a turning point and free segments algorithms are applied in order to solve the obstacle avoidance problem. The basic idea of these strategies is to search for the safe optimal path by determining the turning point of the safest free segments which gives the shortest path. In [10], a fuzzy logic controller (FLC) is adopted in order to solve the problem of mobile robot navigation in strange motion. The basic idea of this approach is to control the steering of the front four wheels individually to obtain the correct heading angle of the vehicle. Four fuzzy controllers for navigation and tracking the desired heading angle while at the same time are adopted. Also, many techniques calculate the optimal collision-free velocity for a mobile robot-like neural network [11], optimization algorithm [12], control point searching algorithm [13]. These works aim to optimize three objectives which are path length, path safety, and path smoothness to solve the path planning problem.

In the area of artificial intelligence, the most important technique of path planning is a genetic algorithm (GA), which has been employed in optimization problems [14]–[16]. In [17], a new dynamic planning navigation algorithm based on a genetic algorithm (DPNA) in an unknown environment is proposed. The basic idea of the contribution developed in [17] is to ensure the navigation of the mobile robot in a dynamic environment with obstacle avoidance. Furthermore, the genetic algorithm is adopted in order to obtain the optimal trajectory. In the present work, we propose a new strategy for mobile robot navigation using the dynamic planning navigation algorithm. First, this approach is based on the potential fields method: the robot is attracted to the target and pushed from the obstacles. Then, the proposed algorithm handles to maximize the distance between the obstacles and to minimize the distance to the target. So, for each displacement, a local target is considered and the robot replanned its route until he found the target.

On the other hand, many several kinds of research are focused to solve the problem of tracking trajectory [18], [19]. In [20], [21], an adaptive neural network is used to control the mobile robot. The aim objective of this control law system is to minimize the error positions. The Proportional Integral Derive (PID) is developed in [22] in order to solve the problem of trajectory tracking. However, this control system becomes unstable especially when it is affected by the sensor sensitivity [23]. Some researchers are focused on the fuzzy logic controller such as [24], [25]. The use of this control system helps the mobile robot to follow the desired trajectory. However, this control system has a complex architecture and needs more time in the simulation. The sliding mode controller is adopted to solve the path tracking problem in [26]–[28]. The aim advantages of this control law system are that it has good performance such as fast response, good transient, and stability.

The contribution of our work is to develop a new algorithm for solving the problem of robot path planning in the environment a priori unknown. This strategy of navigation ensures the safety and the shortest trajectory. With this strategy, the robot becomes able to navigate in the surrounding environment without any collision with obstacles. In this approach, a set of distance sensors are used in order to detect the positions of the obstacles. The first step of this algorithm is to generate a route composed of M local objectives in order to reach the goal. Then, the local objective in any displacement is defined by the smallest evaluation function. This function has the same principle as the potential field technique. After that, the robot moves to the local objective. This helps the mobile robot to follow the desired position. The present contribution is inspired by the work developed in [17]. In [17], a genetic algorithm is used in order to obtain the optimal path. However, in this research, a path planning algorithm is developed in order to obtain a safe path. The aim advantage of this research is that the shortest path was obtained without using the genetic algorithm (GA). As result, the algorithm is more reduced and becomes very easy to implement. Using this strategy, we can rapidly determine the safest and the shortest path. Moreover, once the path is planned, a tracking law based on a sliding mode controller is used for the robot to follow the designed trajectory.

The remainder of this paper is structured as follows. Section 2 introduces the kinematic model of the Khepera II mobile robot. The proposed path planning algorithm is developed in section 3. The tracking path problem is solved by using the sliding mode controller in section 4. Section 5 presents the simulation results in order to prove the efficiency of the developed strategy. In the end, a small conclusion is given in section 6.

II. Khepera II model

In this paper, a non-holonomic mobile robot Khepera II is used as a platform for simulation. Khepera II is a miniature mobile robot of circular shape (diameter $55mm$, height $30mm$ and weight $70g$), designed as a research and education tool [7]. The aim advantage of this mobile robot is that it allows real-world testing of algorithms developed in simulation for trajectory planning, obstacle avoidance, pre-processing of sensory information. The control of the mobile robot is ensured by the acting on the wheel speeds (speed right and left). Fig. 1 shows the schematic model of this robot. Thus, the kinematic model of the used robot is given by:

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

Where V_R and V_L are respectively, the robot's right and left wheel velocities, α the angle which separates the robot direction from the X-axis and Ł is the distance between two wheels.

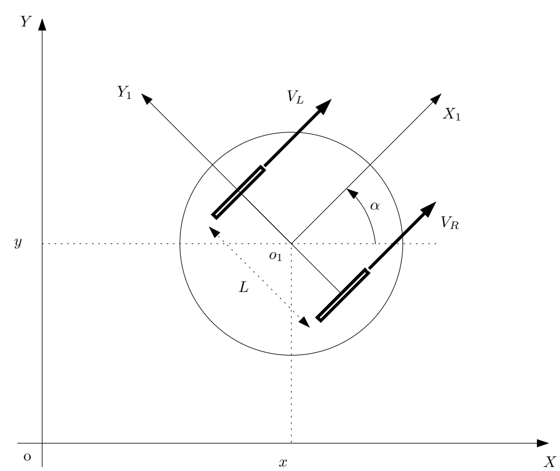


Fig. 1. Schematic representation of Khepera II.

III. Description of the control strategy

In this section, a description of the proposed control strategy for mobile robot navigation is given. The main idea of this technique is based on the potential field method. In this approach, the robot is attracted to the goal and pushed from the obstacles. In this work, we use a new planning navigation algorithm strategy inspired by the work presented in [17]. Our proposed control strategy is consist of M local displacement to research the final objective u^{of}=(x_{of}, y_{of}). Knowing that u_R=(x_R, y_R) is the robot spatial position that has been returned by the localization sensor. In each m^{th} displacement, the mobile robot moves to the local objective $u_{lo}(m)$=($x_{lo}(m)$,$y_{lo}(m)$).

In this work, we need always to know the current position of the robot $u_R(m)$, the target position u^{of}, and the positions of obstacles detected by the n distance sensors. We record all the positions $u_R(m)$, already visited by the robot up to the m^{th} displacement, in the vector u_R.

$$u_R = \begin{bmatrix} u_R(0) \\ u_R(1) \\ \vdots \\ u_R(m-1) \\ u_R(m) \end{bmatrix} \quad (2)$$

Where $u_R(m) = (x_R(m), y_R(m))$.

When the configuration of the mobile robot is $u_R(m) = (u^{0f}(m))$, the algorithm is finished. The basic idea of the planning navigation strategy is summarized in the algorithm presented in Fig. 2. We describe, in the following section, the different steps of this algorithm.

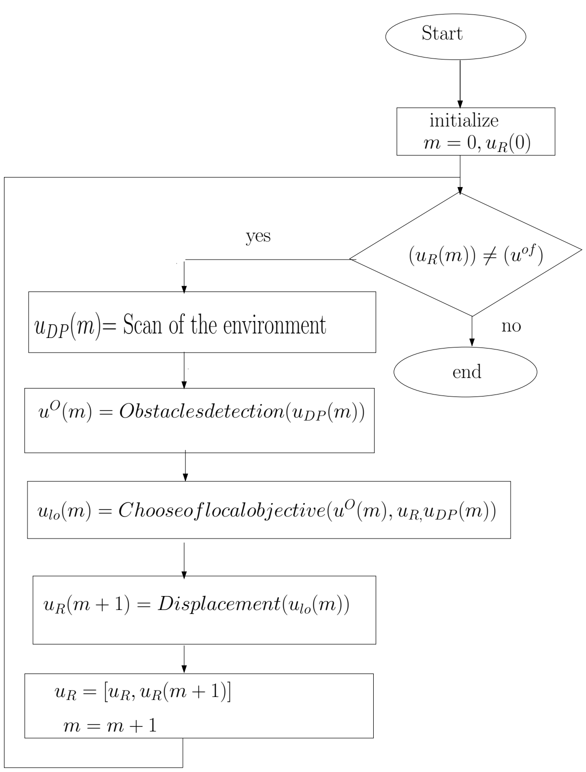


Fig. 2. The proposed algorithm

A. Scan of the environment

In this step, the robot is perform a 360° scan of the environment around its axis. Therefore, each j^{th} sensor returns an analog signal, $s_j(m)$, proportional to the maximum range of the sensors, d_{max}, is expressed in following equation:

$$s_j(m) = \begin{cases} d_j(m) \text{ if } d_j(m) \leq d_{max} \\ d_{max} \quad \text{elsewhere} \end{cases} \quad (3)$$

where, d_j, is the distance measured by the j^{th} sensor attached to the robot.

We define the value of angular displacement, γ, by this expression:

$$\gamma = \frac{360}{n.q}$$

where n is the number of distance sensors, $q-1$ is the number of angular displacement that the robot can make in its axis, for the purpose of decreasing the resolution, and subsequently, reduce the number of sensors to be used.

At the end of this step, we should defined the delimiting polygon (DP) which is determined by a set of K point. It is expressed by the following vector:

$$u_{DP}(m) = \begin{bmatrix} (u_0^{DP}(m)) \\ (u_1^{DP}(m)) \\ \vdots \\ (u_k^{DP}(m)) \\ \vdots \\ (u_{k-1}^{DP}(m)) \end{bmatrix} \quad (4)$$

Where $u_k^{DP}(m) = (x_k^{DP}(m), y_k^{DP}(m))$ and $k = n.q$

This polygon has an important role to delimit the search space of the local objective (detail in step of the choose of the local objective). The delimiting polygon generate for the proposed technique is represented in Fig. 3.

B. Obstacles detection

In this step, the proposed approach generates a virtual polygon (VP) which is defined by a circumference centered on the position of the robot (u_R) with radium r_{pv}. The expression of r_{pv} is given as follows:

$$r_{pv} = d_{max}(1-\epsilon)$$

where ϵ is the factor limited to the rang $0 < \epsilon \leq 0.1$.

Then, we define a new set of L points represented by the vector u^O.

$$u^O(m) = \begin{bmatrix} u_0^O(m) \\ \vdots \\ u_i^O(m) \\ \vdots \\ u_{L-1}^O(m) \end{bmatrix} \quad (5)$$

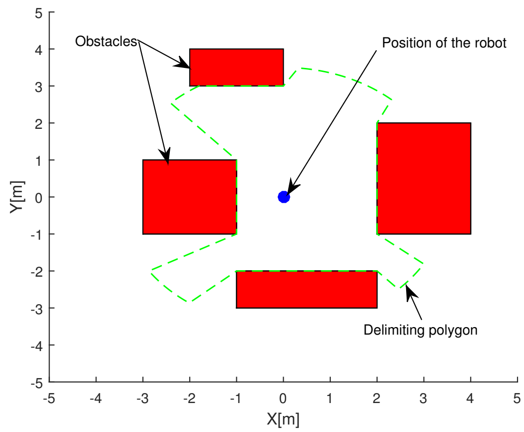


Fig. 3. Example illustrating the delimiting polygon (DP)(dashed green line).

Where $u_i^O(m) = (x_i^O(m), y_i^O(m))$, which satisfies the condition, described by the following equation.

$$u_i^O(m) = u_k^{DP}(m) \text{ if } f_{ed}(u_R(m), u_k^{DP}(m)) \leq r_{pv}$$

Where $L \leq K$.

The function f_{ed} (.,.), which calculates the Euclidean distance between two point p_a and p_b is expressed as follows:

$$f_{ed} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (6)$$

Fig. 4 shows an example of VP (bleu circles) and the set of point u^O(black asterisks).

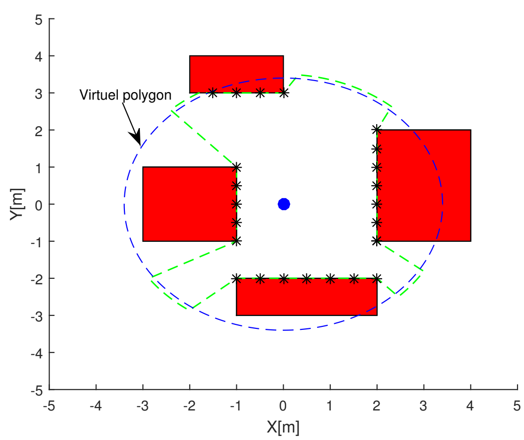


Fig. 4. Example explaining the VP (bleu line) for a case where $\epsilon = 0.1$ and the set of points, u^O, detected (black asterisks) that avoid obstacles.

C. Choose of Local Objective

In this step, the navigation strategy is defined as a local objective, u^{lo}, to which the robot can move. In [17], the DPNA-GA strategy has used a GA to find the local objective. In our work, we eliminate the use of GA and we employed a simple resolution which is detailed in the rest of this part. For each m^{th} displacement, we have executed a set of N points, which are characterized by the following vector.

$$u(m) = \begin{bmatrix} u_0(m) \\ \vdots \\ u_j(m) \\ \vdots \\ u_{N-1}(m) \end{bmatrix} \quad (7)$$

Where $u_j(m) = (x_j(m), y_j(m))$ represents the j^{th} position of the m^{th} displacement of the robot.

All the points, which are represented by the vector u, are generated according to the nonlinear restriction given by the following expression.

$$r_d \geq \sqrt{(x_j(m) - x_R(m))^2 + (y_j(m) - y_R(m))^2} \quad (8)$$

Where r_d is the radius of the circumference centered on the position of the robot in the m^{th} instant. The role of this restriction is to limit the number of potential placements of local objectives.

The evaluation function associated with the j^{th} point is:

$$F_j(m) = d_j^{of}(m) + \frac{\alpha(m)}{d_j^O(m)} + \alpha(m)B_j(m)C_j(m) \quad (9)$$

Where $d_j^{of}(m)$ is the Euclidean distance between the j^{th} point and the final objective, u^{of}. The expression of this distance is:

$$d_j^{of}(m) = f_{ed}(u_j(m), u^{of}) \quad (10)$$

and $d_j^O(m)$ is the shortest Euclidean distance between the j^{th} point and all the L obstacles found, such that

$$d_j^O(m) = \min f_{ed}(u_j(m), u_i^O(m)) \quad (11)$$
$$for\ i = 0, ..., L-1$$

So, for $\alpha(m)$, $B_j(m)$ and $C_j(m)$,we can defined as penalty factors added to each j^{th} point.

In the m^{th} displacement, if the environment without obstacle $(L = 0)$, it is assumed that the optimum evaluation function is simply $d_j^{of}(m)$, so

$$\alpha(m) = \begin{cases} 1 & \text{for} \quad L \neq 0 \\ 0 & \text{for} \quad L = 0 \end{cases} \quad (12)$$

Usually, DP occupies most of the circumference with radius r_d. So, a few points are created outside DP. Also, the point generates within the polygon has greater value than point generated outside it. This idea is translated into the penalty factor B_j as

$$B_j(m) = \begin{cases} 0 & \text{if} \in DP \\ \infty & \text{if} \notin DP \end{cases} \quad (13)$$

If a point $u_j(m)$ is located outside of the m circumferences of radius r_d, centered in the vector of the center, u_R, it will be positively penalized. This reduces its chances of selection.

The next is

$$C_j(m) = \begin{cases} 1 \text{ if } \nexists l \in \{0, ..., m-1\} : z < r_d \\ \text{T if } \exists l \in \{0, ..., m-1\} : z < r_d \end{cases} \quad (14)$$

with T is a relatively large number and $z = f_{ed}(u_j(m), u_R(l))$ is the Euclidean distance.

The evaluation function, expressed in equation 12 has the same principle as the potential field technique [17]. Therefore, $d_j^{of}(n, m)$ (the Euclidean distance between the j^{th} point and the final objective, u^{of}) represents the attractive force to the goal, and $(1/d_j^O(n, m))$. The smallest Euclidean distance between the j^{th} point and all the points associated with the obstacles) represents the force of repulsion between the j^{th} point and all the obstacles presented.

We finalized this step by the selection of the local objective, $u_{lo}(m)$, which has the smallest evaluation function, associated with the m^{th} displacement event. Fig. 5 describes the way of calculating the evaluation function for the j^{th} point $u_j(m)$.

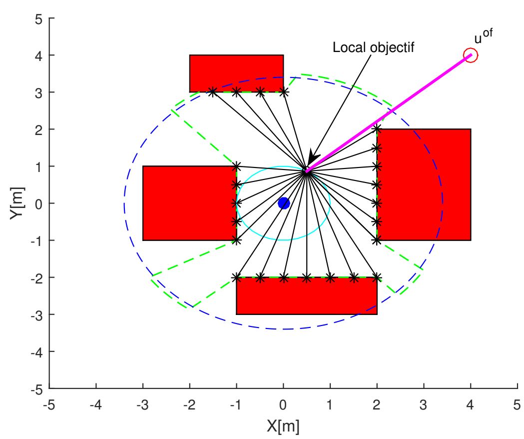


Fig. 5. Example explaining the calculation of the evaluation function for a j^{th} point,$u_j(m)$, in relation to the final objective,$u^{of}$and the obstacles, u^O.

D. Displacement

After these steps, the position of the objective local is defined as $u_{lo}(m)$. By this displacement, a new point $u_R(m+1)$ is generated and described by this expression:

$$u_R(m+1) \neq (u_{lo}(m) \pm \xi) \quad (15)$$

Where ξ is the permissible tolerance related to the local objective. This tolerance is necessary to the robot which is restricted movement, such as non-holonomic robots.

In Fig. 6, the robot can move to the target u_{of} by requiring 9 displacement.

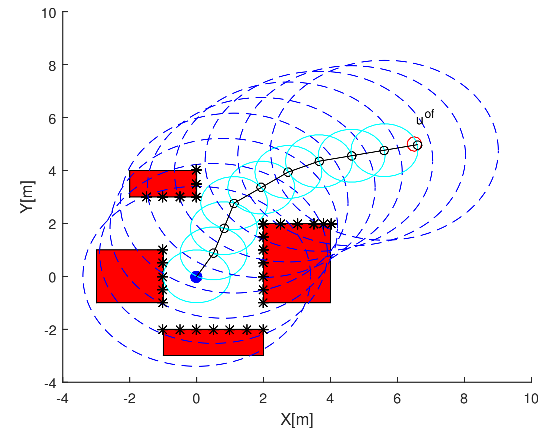


Fig. 6. Example of the displacements (M=9) made by the robot controlled by the proposed strategy to the final point u^{of}.

IV. SLIDING MODE CONTROL

The present section is devoted to solving the path tracking problem using the sliding mode controller. The basic idea of this control system is to guide the wheeled mobile robot in order to follow the desired trajectory with minimum error. To apply this approach, we need to know two essential parameters with are the desired and the current robot configurations. The desired configuration p_r=(x_r, y_r, α_r) is defined as the desired position to be reached and the current robot configuration p=(x, y, α) is defined as its real position at this moment. Then, the difference between the reference position p_r and the current position p is called the tracking error position p_e=(x_e, y_e, α_e). Furthermore, this control system aims to determine the appropriate control vector $q = (v, w)^T$ (v is the linear velocity of the robot and w is its angular velocity). This helps the mobile robot to follow the target position by obliging the tracking error p_e to converge to zero (see Fig. 7).

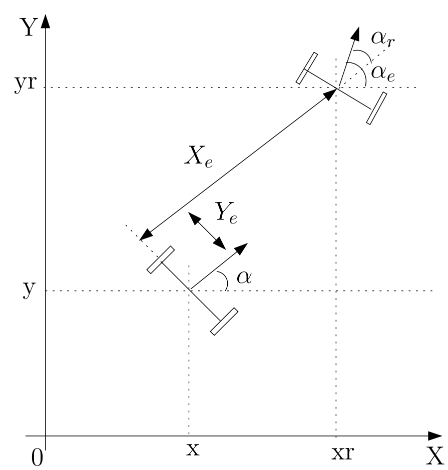


Fig. 7. Tracking error.

The expression of the error position p_e is given as follow:

$$p_e = \begin{bmatrix} x_e \\ y_e \\ \alpha_e \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \alpha_r - \alpha \end{bmatrix} \tag{16}$$

The movement of the mobile robot is controlled by two speeds which are expressed as:

$$V_R = v + \frac{L.w}{2} \tag{17}$$

$$V_L = v - \frac{L.w}{2} \tag{18}$$

On the other hand, it is clear that the design of a switching function is a difficult problem (see equation 1) because the Khepera II mobile robot is a multiple-input nonlinear system. That's why, x_e is chosen equal to zero at the first switching surface to simplify the calculation, according to [3], [4]. Then, the Lyapunov candidate function is defined as $V = \frac{1}{2}y_e^2$ and the time derivative is of this function is expressed as follows:

$$\begin{aligned} \dot{V} &= y_e\dot{y_e} = y_e(-x_e w + v_r \sin(\alpha_e)) \\ &= -x_e y_e w - v_r y_e \sin(\arctan(v_r y_e)). \end{aligned} \tag{19}$$

Then, we define the global condition $\dot{V} \leq 0$, which is always satisfied because $v_r y_e \sin(\arctan(v_r y_e)) \geq 0$ and $\alpha_e = -\arctan(v_r y_e)$ is considered as a switching candidate function. The vector of sliding surfaces is expressed in the following equation:

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_e \\ \alpha_e + \arctan(v_r y_e) \end{bmatrix} \tag{20}$$

The convergence of s_2 to zero requires that x_e converges to zero. if s_2 converge to zero, trivially $\alpha_e + \arctan(v_e y_e)$ converge to zero, so that α_e becomes equal to $-\arctan(v_e y_e)$ and y_e to 0.

For the determination of the control law, the saturation function is considered as a switching function (see equation 24)

$$\dot{s} = -ksat(s). \tag{21}$$

The vector of the sliding surfaces is expressed as:

$$\begin{aligned} \dot{s} &= \begin{bmatrix} \dot{x_e} \\ \dot{\alpha_e} + \frac{\partial\gamma}{v_r}\dot{v_r} + \frac{\partial\gamma}{y_e}\dot{y_e} \end{bmatrix} \\ &= \begin{bmatrix} y_e w + v_r \cos\alpha_e - v \\ w_r + \frac{\partial\gamma}{\partial v}\dot{v} + \frac{\partial\gamma}{\partial y_e}(v_r \sin\alpha_e - x_e w) - w \end{bmatrix} \end{aligned} \tag{22}$$

Finally, the designed control law is given as follows [3]:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} y_e w + v_r \cos\alpha_e + k_1 sat(s_1) \\ w_r + \frac{\partial\gamma}{\partial v_r}\dot{v_r} + \frac{\partial\gamma}{\partial y_e}(v_r \sin\alpha_e) + k_2 sat(s_2) \end{bmatrix} \tag{23}$$

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

V. Simulation results

In this section, to demonstrate the efficiency of the proposed navigation approach, some simulation results are presented. Four situations are presented in different environments (see Figs. $8-11$). In all simulations, the routes followed by the robot are presented by continuous black lines, and the displacement is presented by circles on the routes lines).

As shown in Fig. 8, the robot which starts from the initial point (5, 5), is able to reach the target position (30, 25) without collision with obstacles. For the scenarios illustrated in Fig. 9, it is clear that the mobile robot is able to follow the goal configuration. In Fig. 9a that the mobile robot follows the goal (800, 1600) safely. When the target position is changed (1600, 1200), the trajectory of the mobile robot is changed also. It is noted that the mobile robot is able to follow the target position even the configuration of the target is modified. Furthermore, in Fig. 9(a), the proposed navigation algorithm requires approximately $M = 11$ displacement. For Fig. 9(b), the proposed navigation algorithm requires approximately $M = 13$ displacement.

In Fig. 10, the robot which starts from an initial point (5, 2), moves easily towards the target u^{of}=(45, 35) without hitting obstacles. In this situation, the proposed approach required 20 displacement to ensure the navigation of the mobile robot to its destination.

To test the efficiency of the proposed navigation strategy, we made a moderately complex environment (see Fig. 11). In this case, the robot starts from the point $(-4, -1.8)$ and it needs 34 displacement ($M = 34$) in order to follow the target position u^{of}=(-3.8,2.2).

On the other hand, the proposed algorithm is applied when the mobile robot navigates in a dynamic environment. As shown in Fig. 12, it is clear that the mobile robot is able to navigate without hitting obstacles. In the same context, Fig. 13 illustrates that the coordinates of the robot and the dynamic obstacle are not equal in time collision of collision. So, it is obvious that there is no collision between the dynamic obstacles and the robot. Therefore, by using this algorithm, the robot mobile is able to navigate perfectly and without any problem in the dynamic environment.

Based on the presented simulation results, it is noted that the newly proposed technique is able to solve the path planning problem in static and dynamic environments.

On the other side, to demonstrate the performance of the sliding mode controller, some simulations results are presented. Fig. 14 (14(a), 14(b), 14(c), 14(d)) shows that the robot is able to track the desired trajectory and follow the target position without any problem. From Fig. 15 (15(a), 15(b), 15(c), 15(d)), we can see that the errors tracking always converge to zero.

According to these simulations, it is noted that the sliding mode control is a suitable solution to solve the problem of trajectory tracking and it shows good performances.

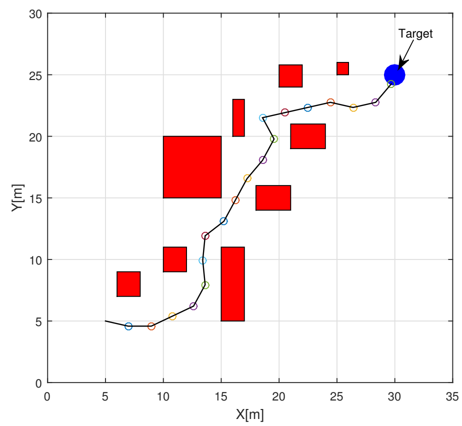


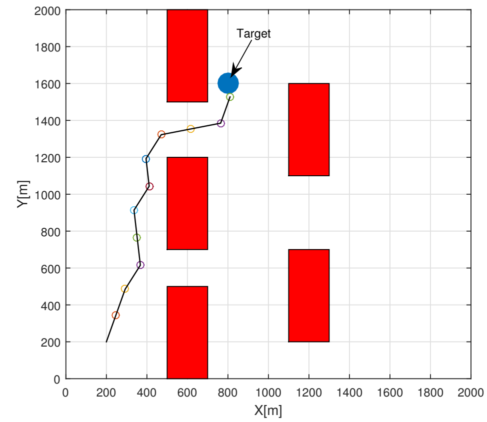
Fig. 8. Path planning of the mobile robot in an environment S_1 with the proposed strategy.

Compared to the work developed in [17], it is obvious that the simulation results presented in this research give better results than the result obtained on the research developed in [17]. First, when the mobile robot navigates in the comparing the same environments (environment S_4 and the environment S_2 using in [17] with $R_c = 80\%$, $J = 10$, $H = 30$)). It is clear that the time of simulation using our strategy ($t_p = 1.3$ s) is less that the time of simulation obtained in [17] ($t_p = 15.64$ s). The aim advantage of the developed algorithm is that the path planning using our algorithm is faster than that proposed algorithm in [17].

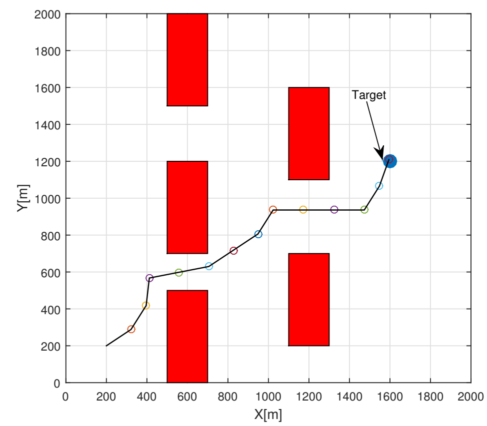
So with the proposed approach, we save for the simulation time and also the simplification of implementation.

VI. Conclusion

In the present work, a dynamic planning navigation algorithm DNPA for wheeled mobile robot Khepera II has been developed. The aim objective of the developed contribution is to solve the path planning algorithm. First, the navigation problem has been tackled when the mobile robot navigates in a static environment. Then, the DNPA algorithm has been applied to search the path of navigation in a dynamic environment. On the other hand, a sliding mode controller has been used to control the mobile robot and guarantee the reactivity, robustness, and stability response. Simulation results have proved that the



(a) u^{of} =(800,1600).



(b) u^{of}= (1600,1200).

Fig. 9. Path planning of the mobile robot in an environment S_2 with the proposed strategy.

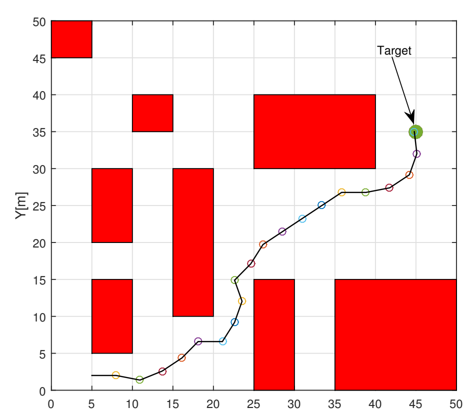


Fig. 10. Path planning of the mobile robot in an environment S_3 with the proposed strategy.

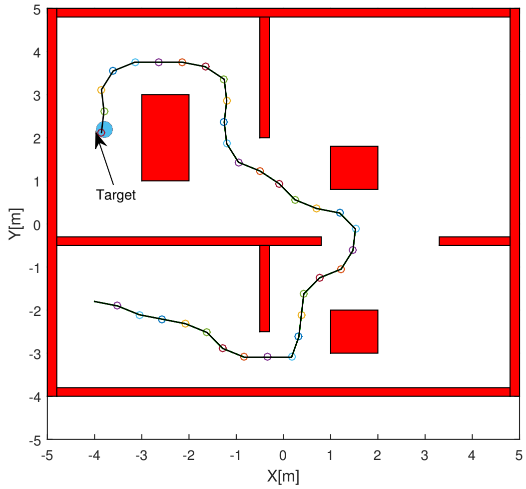


Fig. 11. Path planning of the mobile robot in a complex environment S_4 with the proposed strategy.

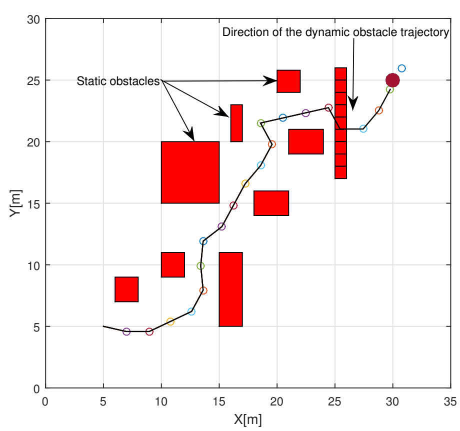


Fig. 12. Path planning of the mobile robot in an environment S_1 contains one dynamic obstacle, (x_{do}, y_{do})=$(25, 25)$, with the proposed strategy.

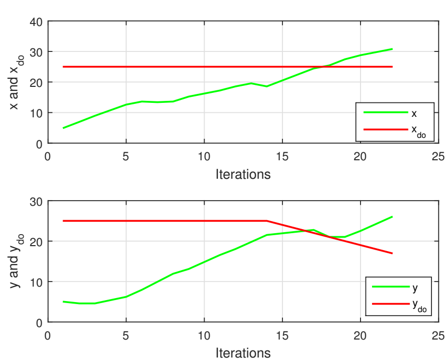
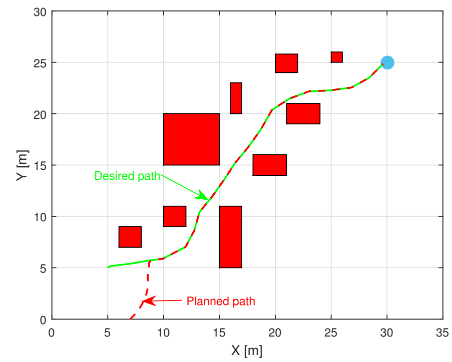
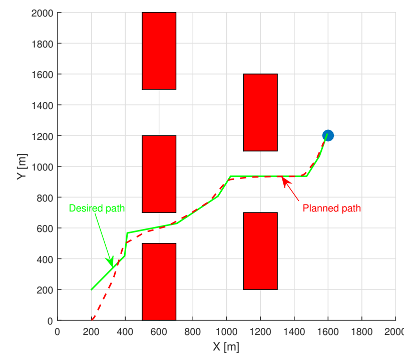


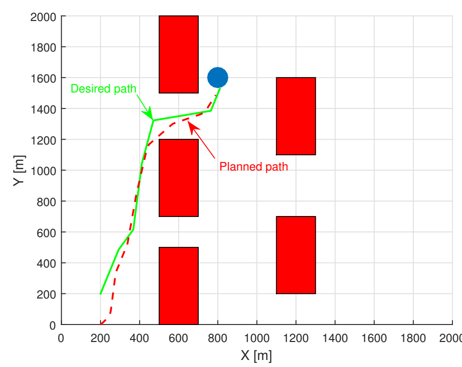
Fig. 13. Coordinates of the robot and the dynamic obstacle according to time in the case of a vertical mobile obstacle trajectory.



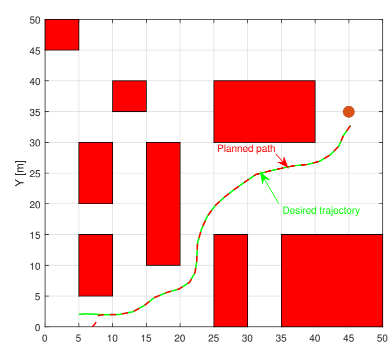
(a) Tracking planned of Fig. 8.



(b) Tracking planned of Fig. 9(b).

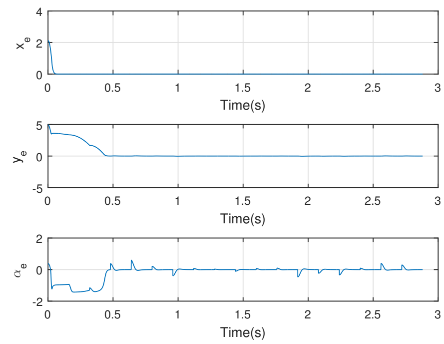


(c) Tracking planned path of Fig. 9(a).

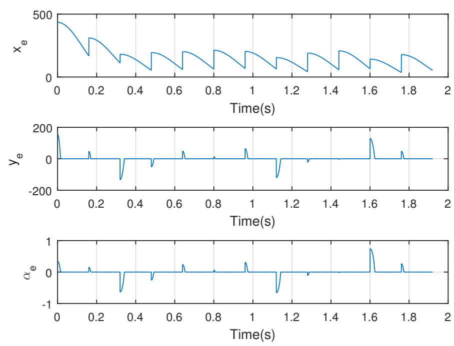


(d) Tracking planned path of Fig. 10.

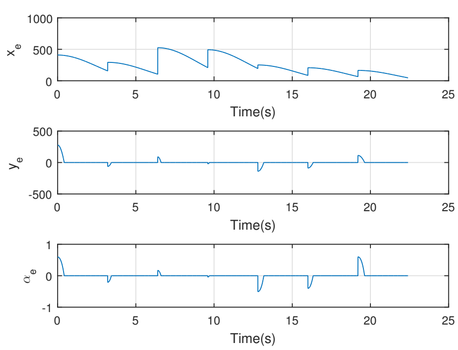
Fig. 14. Tracking planned path.



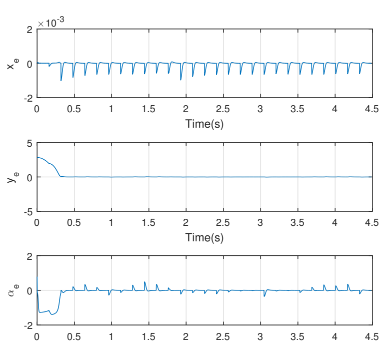
(a) Case of Fig. 8.



(b) Case of Fig. 9(b).



(c) Case of Fig. 9(a) .



(d) Case of Fig. 10.

Fig. 15. Tracking Errors (x_e ,y_e ,α_e).

developed algorithm is a suitable solution to solve the path planning problem in various types of environments (simple or complex and static or dynamic). Also, simulation results have mentioned the good performances of the sliding mode controller. It has been found that the proposed controller gives good results. As future work, we want to apply the developed algorithm on a real prototype Khepera IV mobile robot.

REFERENCES

[1] W. Rahmaniar and A. Wicaksono, “Design and Implementation of A Mobile Robot for Carbon Monoxide Monitoring,” *Journal of Robotics and Control (JRC)*, vol. 2, no. 1, pp. 1-6, 2021.

[2] H. Zhang, W. Lin, and A. Chen, “Path Planning for the Mobile Robot: A Review,” *Symmetry*, vol. 10, no. 10, p. 450, Oct. 2018,

[3] 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*, vol. 2, no. 1, pp. 172-186, 2022.

[4] S. Sahloul, D. Ben Halima and C. Rekik, “Tracking trajectory of a mobile robot using sliding mode control,” *International Multi-conference on Systems, Signals and Devices (SSD’18)*, 2018, pp. 1351-1355.

[5] U. O. Rosas, O. Montiel and R. Sepulveda, “Mobile robot path planning using membrane evolutionary artificial potential field,” *Applied Soft Computing*, vol. 77, pp. 236-251, 2019.

[6] 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)*, vol. 2, no. 4, pp. 351-355, 2021.

[7] M. Boujelben, C. Rekik and N. Derbel, “A multi-agent architecture with hierarchical fuzzy controller for a mobile robot,” *International journal of Robotics and Automation*, vol. 30, no. 3, pp. 289-298, 2015.

[8] I. Hassani, I. Maalej and C. Rekik, “Robot path planning with avoiding obstacles using free segments and turning points algorithms,” *Mathematical problem in Engineering*, vol. 2018, no. 6, pp. 1-13, 2018.

[9] I. Hassani, I. Maalej and C. Rekik, “Mobile robot navigation using turning point algorithm and sliding mode controller,” *International Multi-conference on Systems, Signals and Devices (SSD’18)*, pp. 1345-1350, 2018.

[10] A. N. Ouda, A. Mohamed, “Autonomous fuzzy heading control for a multi-wheeled combat vehicle,” *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 90-101, 2021.

[11] 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)*, vol. 3, no. 1, pp. 62-72, 2022.

[12] C. Rekik, M. Jallouli, N. Derbel, “Optimal trajectory of a mobile robot using hierarchical fuzzy logic controller,” *International Journal of Computer Applications in Technology*, vol. 53, no. 4, pp. 348-357, 2016.

[13] I. Hassani, I. Maalej and C. Rekik, “Control points searching algorithm for autonomous mobile robot navigation with obstacle avoidance,” *19th international conference on Sciences and Techniques of Automatic control and computer engineering (STA’19)*, 2019, pp. 158-163.

[14] Y. K. Ever, “Using simplified swarm optimization on path planning for intelligent mobile robot,” *International Conference on Theory and Application of Soft Computing with Words and Perception*, ICSCCW, 2017, pp. 83-90.

[15] K. Hao, J. Zhao , B. Wang , Y. Liu and C. Wang, “The application of an adaptive genetic algorithm based on collision detection in path planning of mobile robots,” *International Journal of Computational Intelligence and Neuroscience*, vol. 2021, pp. 1-20, 2021.

[16] W. Rahmaniar and A. E. Rakhmania, “Mobile Robot Path Planning in a Trajectory with Multiple Obstacles Using Genetic Algorithms,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 3, pp.1-7, 2021.

[17] A. V. F. M. de Oliveira and M. A. C. Fernandes, “Dynamic planning navigation strategy for mobile terrestrial robots,” *Robotica*, vol. 34, pp. 568-583, 2016.

[18] H. Cen and B. K. Singh, "Nonholonomic wheeled mobile robot trajectory tracking control based on improved sliding mode variable structure," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1-9, 2021.

[19] T. T. Pham and M. T. Le, "Omnidirectional mobile robot trajectory tracking control with diversity of inputs," *International Journal of Mechanical Engineering and Robotics Research*, vol. 10, no. 11, pp. 639-644, 2021.

[20] 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*, vol. 2, no. 1, pp. 37-56, 2022.

[21] Y. Koubaa, M. Boukattaya and T. Damak, "Intelligent control for an uncertain mobile robot with external disturbances estimator," *New Trends in Robot Control*, Springer, 2020, pp. 155-176.

[22] L. T. Anh, T. T. T. Nga and Vu Van Hoc b, "PID-type iterative learning control for output tracking gearing transmission systems," *International Journal of Robotics and Control Systems*, vol. 1, no. 3, pp. 256-268, 2021.

[23] A. Maarif and N. R. Setiawan, "Control of DC motor using integral state feedback and comparison with PID: simulation and arduino implementation," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 256-268, 2021.

[24] Iswanto Iswanto and Irfan Ahmad, "Second Order Integral Fuzzy Logic Control Based Rocket Tracking Control," *Journal of Robotics and Control (JRC)*, 2021, vol. 2, pp. 594-604 , 2021.

[25] F. Cherni, Y. Boutereaa, C. Rekik and N. Derbel, "Chattering reduction on the control input of a nonholonomic mobile robot using fuzzy logic controller," *Journal of Engineering Science and Technology Review*, vol. 9, no. 3, pp. 41-48, 2016.

[26] A. Bavarsad, A. Fakharian and M. B. Menhaj, "Optimal sliding mode controller for an active transfemoral prosthesis using state-dependent riccati equation approach," *Arabian Journal for Science and Engineering*, vol. 45, no. 8, pp. 6559-6572, 2020.

[27] M. S. Mahmoud, A. Alameer and M. M. Hamdan," An adaptive sliding mode control for single machine infinite bus system under unknown uncertainties," *International Journal of Robotics and Control Systems*, vol. 1, no. 3, pp. 226-243, 2021.

[28] H. Sira-Ramírez, M. A. Aguilar-Orduna and E. W. Zurita-Bustamante, "On the sliding mode control of MIMO nonlinear systems: An input-output approach," *International Journal of Robust Nonlinear Control*, vol. 29, no. 3, pp. 715-735, 2019.