Journal of Robotics and Control (JRC)
Volume: 2, Issue: 4, July 2020
ISSN: 2715-5072, DOI: 10.18196/jrc.2483

An hybridization of global-local methods for autonomous mobile robot navigation in partially-known environments

Samia SAHLOUL[1*], Donia BEN HALIMA Abid[2] and Chokri REKIK[3]
[1,2,3]Control & Energy Management Laboratory (CEM), University of Sfax, Sfax Engineering School, BP W, 1173-3038 Sfax, Tunisia.
E-mails: [1]samiasahloul12@gmail.com, [2]donia.benhalima-abid@enis.tn, [3]chokri.rekik@enis.rnu.tn
[*]Corresponding Author

***Abstract*—This paper deals with the navigation problem of an autonomous non-holonomic mobile robot in partially-known environment. In this proposed method, the entire process of navigation is divided into two phases: an off-line phase on which a distance-optimal reference trajectory enables the mobile robot to move from an initial position to a desired target which is planned using the B-spline method and the Dijkstra algorithm. In the online phase of the navigation process, the mobile robot follows the planned trajectory using a sliding mode controller with the ability of avoiding unexpected obstacles by the use of fuzzy logic controller. Also, the fuzzy logic and fuzzy wall-following controllers are used to accomplish the reactive navigation mission (path tracking and obstacle avoidance) for a comparative purpose. Simulation results prove that the proposed path planning method (B-spline) is simple and effective. Also, they attest that the sliding mode controller track more precisely the reference trajectory than the fuzzy logic controller (in terms of time elapsed to reach the target and stability of two wheels velocity) and this last gives best results than the wall-following controller in the avoidance of unexpected obstacles. Thus, the effectiveness of our proposed approach (B-spline method combined with sliding mode and fuzzy logic controllers) is proved compared to other techniques.**

***Index Terms*—Mobile robot, unexpected obstacles, fuzzy logic controller, sliding mode controller, B-spline, Dijkstra.**

I. INTRODUCTION

In the last few years, there is a great evolution in the use of mobile robots in every day human life such as industries, hospitals, supermarkets, etc, because of their efficiency, accuracy, and time saving. For that purpose, the finding of an appropriate navigation control law has been the object of several researches.

The autonomous mobile robot's navigation methods can be generally classified into two mean types: Global and local navigation. The first type includes the methods based on path or trajectory planning where a prior knowledge of the environment is required to generate a random-safe path according to some guideline and taken into account the static obstacles positions. Among these methods, we can cite for instance, the visibility graphs method [40], the cell decomposition techniques [41], [49], the artificial potential field method [10], [57], the firefly algorithm [54], [55], [59], the Hamiltonian functions based methods [46], the RRT (Rapidly exploring Random Tree) based techniques [47], [48], the Free segments and turning points algorithm [6] and, the heuristic approaches based on artificial intelligence, such as, the neural network technique [52] which is inspired by the biological sensual systems, the fuzzy logic technique which have the benefit of continuous execution and heartiness, the genetic algorithm [2], [5] , the particle swarm optimization technique [13], [56], and, the ant colony optimization [51], [53]. Also, the hybrid fuzzy logic approaches have been used for the path planning task, such as the fuzzy-wind driven optimization algorithm [8]. In general, the local navigation methods type ensures the safety and the fastness of navigation, although, it suffers from the problem of complete knowledge requirement of the workspace, which is not the case for most realistic workspaces. Once planned, the mobile robot should be able to follow the reference path. In the literature, there are various methods used for the tracking control of non-holonomic mobile robots which are nonlinear and uncertain systems, such as, the Nonlinear Dynamic Inversion method [3], the Proportional Integral Derive (PID) controller [35] suffering from instability problem due to sensor sensitivity, the fuzzy logic controller [15], [58] characterized by its lack of precision, and, the Sliding Mode Controller [4] which has the advantage of robustness, fast response and insurance for stability.

In the second type of navigation methods, there is no need of the environment to be totally known, in fact, the mobile robot uses its various sensors to discover, in real time, the workspace, in order to navigate safely. Among the local navigation methods, we can list for example, the Voronoi Diagrams method [45], [50], the Vectors Field Histogram method [23], the Beam Curvature method [25], the methods based on Fuzzy Logic Controller [31], the Multi-Agent Systems method [7], [11], [21], the Hierarchical Fuzzy method [14], [26], the Tangent Circle algorithm [1], and the Neural Networks method [9], [16]. These navigation techniques have the advantage to be very effective in dynamic and unknown environments. However, they suffers from the problem of local minima which prevents the path from being optimal according to a chosen criterion (distance travelled, traveling time, etch). In order to overcome the drawbacks of the above methods (local and global navigation) and to bring, in same time, their advantages,

the combination of both types of methods is required.

In the literature, several hybrid techniques have been used to accomplish the autonomous navigation issue. Wang et al. [37] have combined the global path planning and the local navigation methods : Distance transform and potential field for the mobile robot navigation in indoor environment. This combination suffers from the problem of local minima. Castro et al. [42] have proposed a multi-loop modular navigation architecture composed of local and global navigation methods in order to ensure the stability and the safety of navigation. Nakhaeinia et al. [22] have proposed an hybrid control architecture for mobile robot navigation in unknown dynamic environment. Which is a combination of the deliberative and reactive navigation architectures developed based on a modeling-planning-reaction configuration. Hank et al. [17] have combined the global navigation method (RPA method and fuzzy logic controller) with the local navigation one for avoiding unexpected obstacles using the wall-following method. This approach suffers from the problem of high level of tracking errors given by the fuzzy logic controller and instability of velocities.

The aim of the proposed technique is to perform a safe autonomous navigation with minimum distance travelled in partially-known environment. The challenges are to save in memory and in energy when planning and saving planned trajectory, using a simple and effective method which does not require large space to save planned trajectory and does not require heavy calculation for the planning process. Also, to improve the track of planned trajectory (reducing the track errors level and insure stability) and to enhance the avoidance of unexpected obstacles. The navigation process presented in this paper is an hybridization of the global path planning and the local navigation methods. Thus, in the off-line navigation phase (global path planning), a set of control points is placed randomly in the environment taken into account the static obstacles positions. Once placed, each control point is connected to its neighbor using the B-spline method in order to generate a safe roadmap of the environment. Then, the start and goal points are added and using the Dijkstra algorithm, the shortest path that binds the start and goal positions is calculated and memorised like a succession of control points.

In the online navigation phase, the mobile robot should follow the planned trajectory using the sliding mode controller and, in the same time, it must be able to avoid unexpected obstacles using the fuzzy logic controller in order to accomplish the whole navigation task. As well, for a comparison reason, the online navigation task is conducted by the fuzzy logic controller for the track of planned trajectory and the wall-following method for the avoidance of unexpected obstacles.

Our first contribution is to use a simple and effective method to generate the environment roadmap using the B-spline method and the Dijkstra algorithm. This proposed technique does not require a heavy calculation. Also, the planned path is memorised like a succession of control points which allows to minimize the storage space needed. The second contribution is the use of sliding mode controller to follow the planned trajectory for guaranteeing the stability, robustness and reactivity of track compared to the fuzzy logic controller. In fact, the track approach presented by Hank et al. [29] present tracking errors that are of high level and unstable. The third contribution is the use of the fuzzy logic controller to avoid unexpected obstacles which gives more stable two wheel velocities configuration compared to the fuzzy wall-following method.

The rest of this paper is organised as follows. Section 2 is about the problem statement to be solved in this paper. Section 3 is devoted to the presentation of the proposed control strategy. Section 4 describes the off-line phase of the proposed navigation method. Section 5 presents the online phase. Section 6 presents and discusses some simulation results. Section 7 gives the conclusion and future work.

II. Problem statement

The problem to be solved in this paper is the autonomous navigation of a non-holonomic mobile robot (Khepera IV) in partially-known environment. Figure 1 shows the considered workspace in which known obstacles are shown by purple boundaries and the rest of this environment can contain unexpected obstacles during real-time navigation. Figure 2 shows the schematic model of the Khepera IV mobile robot. This mobile robot has two independent driving wheels. Thus, in order to move the Khepera IV, the speed of both wheels should be given (left and right wheel speeds). The kinematic model of Khepera IV mobile robot is given by [14], [33]:

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

where V_R and V_L are ,respectively, the mobile robot's right and left wheel velocities and α is the angle between the robot direction and X-axis.

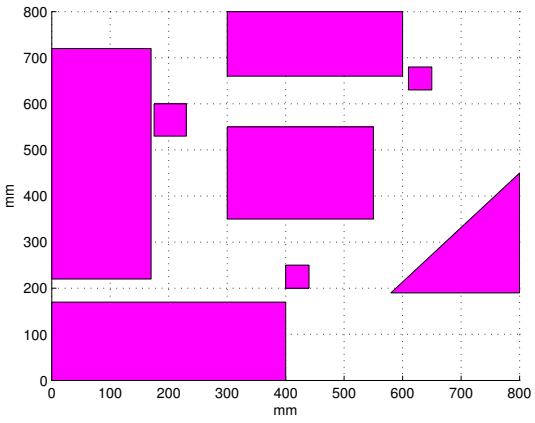

Fig. 1. The considered environment.

III. Control strategy

In this section, the proposed control strategy of Khepera IV autonomous navigation is described. This navigation process is

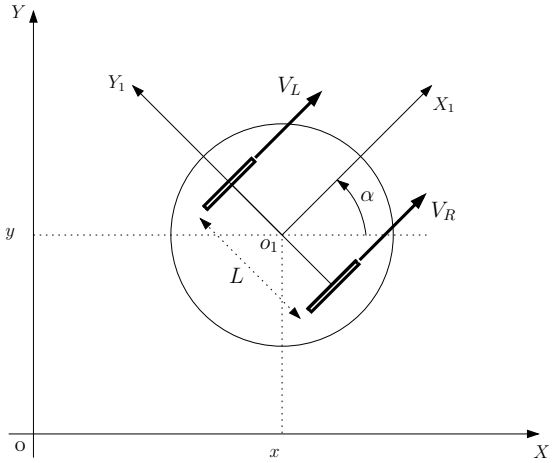

Fig. 2. Schematic model of the non-holonomic Khepera IV mobile robot.

decomposed into two modules (see figure 3). The first module (M_1) represents the off-line phase of navigation (global navigation) in which a distance-optimal reference trajectory linking the starting point to the target point is planned in the proposed environment using the B-spline method [43] and the Dijkstra algorithm [28]. Then, this reference trajectory is memorised like a succession of control points representing the reference posture of the virtual mobile robot to be tracked by the mobile robot. The second module (M_2) represents the online phase of navigation (local navigation) in which there are two possible scenarios, whether the track of reference trajectory or the avoidance of unexpected static obstacles. The choice between these two scenarios is assured by a decision module whose inputs are the current mobile robot position and the information given by its sensors. Thus, if there is no obstacles detected, the decision module activates the tracking trajectory module, else, the avoidance of unexpected static obstacles module will be activated.

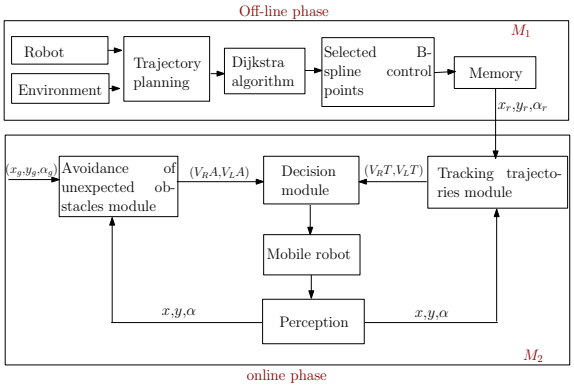

Fig. 3. Control strategy of the mobile robot navigation.

IV. Trajectory planning: Off-line phase

In the literature, there are various methods used to plan reference trajectory for the mobile robot navigation. We can cite for example, the RPA technique (Random Profile Approach) [34], [36]. The principle of this method is to randomly generate a path and a movement on this path using the B-spline and the cubic-spline functions, respectively. Then, with an optimisation procedure, the optimal positions, in terms of travelling time, are selected which allows to obtain a roadmap of near-time-optimal reference trajectories. Also, the method of cell decomposition technique is used [18], it is based on the principle of decomposing the free space of the environment into cells taking into account the positions of static obstacles and it finds in the end the optimal cell succession that leads the mobile robot from starting point to the goal point. Furthermore, the PSO (Particle Swarm Optimization) technique [12] is used for the trajectory planning task, this method is inspired by the collective behavior of social animals, such as bird, fishes, bees when searching their food, which allows to obtain optimal paths in terms of distance travelled and traveling time.

In this paper, the B-spline method [43], [44] is chosen to plan reference trajectory and the Dijkstra algorithm is used to select the shortest path. The whole planning process is decomposed in six steps:

- Step 1: To place randomly in the environment a number of nodes taking into account the location of static obstacles (see Figure 4).
- Step 2: To place between every two nodes in the environment a number of control points (see Figure 5).
- Step 3: To construct the reference trajectory using the B-spline method (see Figure 6).
  The principle of the B-spline method is to divide the trajectory length m into n sections $[m_0(=0), ..., m_k, m_{k+1}, m_n(=m_e)]$. The mobile robot's trajectory is $P(m) \equiv (x(m), y(m))^T$. Using the B-spline of order three at the section $m_k \leq m \leq m_{k+1}$ is described by control points $Q_k = [x_k, y_k](k = 0, ..., n)$ as follow:

$$P(m(l)) = \sum_{j=0}^{3} N_j(l).Q_{k+j-1} \tag{2}$$

$$m(l) = m_k + l.\Delta m_k, 0 \leq l \leq 1 \tag{3}$$

  With $\Delta s_k \equiv s_{k+1} - s_k$, $\Delta m_k \equiv m_{k+1} - m_k$ and $N_j(l)$ is a base function of B-spline defined as follow:

$$N_j(l) = \begin{cases} (1-l)^3, & \textbf{if}, j = 0 \\ 3.l.(1-l)^2, & \textbf{if}, j = 1 \\ 3.l^2.(1-l), & \textbf{if}, j = 2 \\ (1)^3, & \textbf{if}, j = 3 \end{cases} \tag{4}$$

  The B-spline of order three is chosen to ensure the continuity on curvature of the trajectory.
- Step 4: the initial and the target points (P_i and P_t) are placed in the environment and they are connected to nearest nodes using the B-spline method (see figure 7).
- Step 5: The determination of the estimated distance-optimal reference trajectory is done by the standard graph optimization algorithm developed by Dijkstra [28]. The principle of the Dijkstra algorithm is described as follow: In order to determinate the path of minimal distance between two given nodes P and Q, we use the fact that, if

R is a node on the minimal path from P to Q. Knowledge of the latter implies the knowledge of the minimal path from P to R. Then, the minimal paths from P to other nodes are constructed in order of increasing length until Q is reached. Figure 8 shows the optimal succession of nodes obtained that links the start and the goal points.

- Step 6: The reference trajectory constructed is memorised like a succession of B-spline control points. In this case, the reference trajectory is formed by 24 control points.

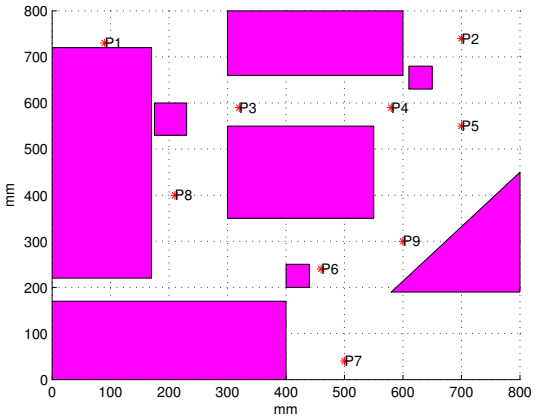

Fig. 4. The proposed environment. P_i (i=1:9) are the nodes placed randomly in the mobile robot environment.

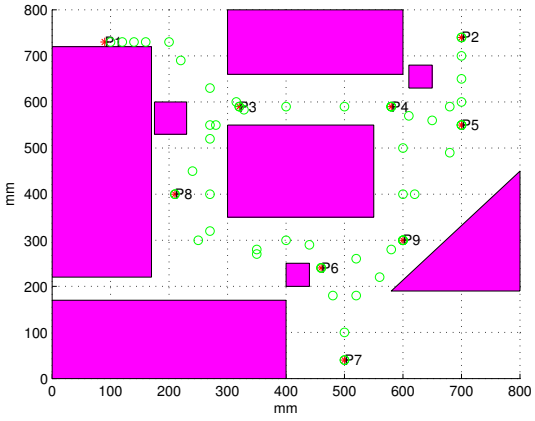

Fig. 5. The placement of control points between every two nodes in the environment.

V. Online Phase

The online phase is composed by two modules: the tracking trajectory module and the avoidance of unexpected obstacle module which are described below.

A. Tracking trajectory

The distance-optimal reference trajectory planned in the off-line phase, should be followed by the mobile robot. In the literature, there are several techniques used to accomplish this task, such as the Fuzzy Logic Controller [19], the technique

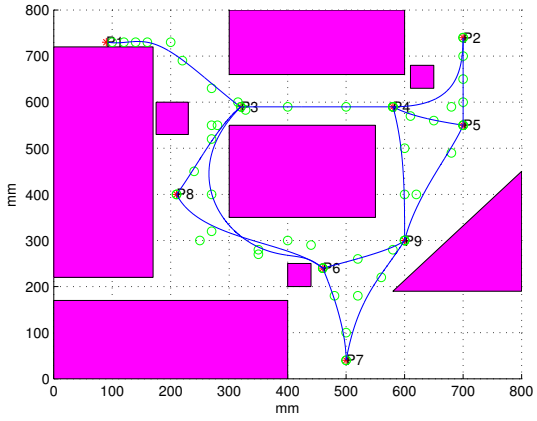

Fig. 6. The result of the trajectory planning phase between the different nodes of the environment.

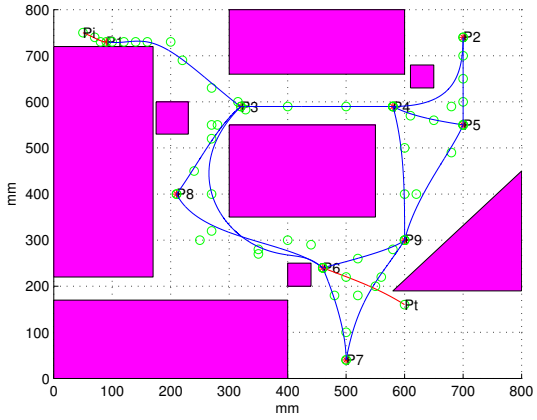

Fig. 7. Placement of the initial and end points in the environment. P_i is the start point and P_t is the target point.

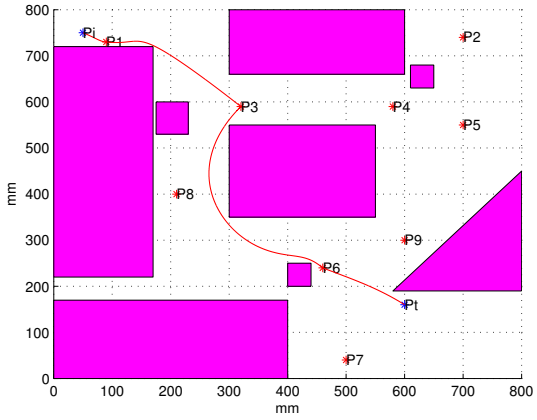

Fig. 8. The estimated optimal-distance succession of nodes.

inspired from the Potential Fields [20] and the Discrete PDC Controller [30].

In this paper, the sliding mode controller is chosen to track the planned trajectory. Also, in order to prove the effectiveness

of our proposed trajectory tracker (sliding mode controller), this mission of path tracking is done by the fuzzy logic controller.

1) Sliding Mode control: The principle of this technique is to generate a control vector $q = (v, w)^T$ representing the linear and angular velocities of the mobile robot, based on its current position vector $p = (x, y, \alpha)$ and its reference vector $p_r = (x_r, y_r, \alpha_r)$. Firstly, the relationship between the current position vector p and the control vector q is defined and is given as follows:

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & 1 \end{bmatrix} q. \tag{5}$$

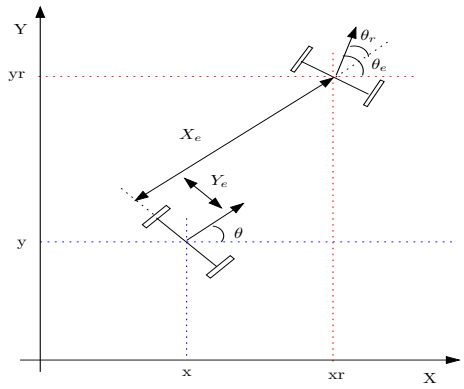

Fig. 9. Coordinates assignment for the mobile robot.

Figure 9 illustrates the error vector, $p_e = (x_e, y_e, \alpha_e)$ which is defined as the difference between the current position vector p and the posture vector p_r and is given by [32]:

$$pe = \begin{bmatrix} x_e \\ y_e \\ \alpha_e \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_r - x \\ y_r - y \\ \alpha_r - \alpha \end{bmatrix} \tag{6}$$

The linear velocity of the virtual mobile robot to be followed is chosen to be constant v_r = $1mm/s$ and the angular velocity is given by this equation:

$$w_r = \dot{\alpha}_r \tag{7}$$

with:

$$\alpha_r = \text{atan}(\frac{\dot{y}_r}{\dot{x}_r}) \tag{8}$$

The velocity error of the mobile robot is represented as follow:

$$\dot{pe} = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\alpha}_e \end{bmatrix} = \begin{bmatrix} y_e w - v + v_r \cos\alpha_e \\ -x_e w + v_r \sin\alpha_e \\ w_r - w \end{bmatrix} . \tag{9}$$

In order to solve the tracking trajectory problem, a control input $q = (v, w)^T$ should be found, so that given any initial errors, p_e can be bounded asymptotically to 0 when $t \to \infty$. Knowing that the mobile robot moves with constant linear velocity v, the problem amounts to calculate the angular velocity w. The kinematic model of the non-holonomic mobile robot represents a nonlinear system. Therefore, there are a problem in the design of a switching function. To overcome this problem, $x_e = 0$ is chosen as the first switching surface. Thus, the Lyapunov function can be defined as:

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

The time derivative of V is given by:

$$\dot{V} \quad = y_e \dot{y}_e = -y_e x_e w - v_r y_e \sin(\arctan(v_r y_e)) \tag{11}$$

With $\alpha_e = \arctan(v_r y_e)$ is chosen as a switching function, the overall stability condition requires that $\dot{V}$ be a negative function, that is to say $\dot{V} \leq 0$. Therefore, $\sin(\arctan(v_r y_e))$ must to be positive. When x_e converges to 0 and α_e converges to $-\arctan(v_r y_e)$, y_e also converges to 0.

Therefore, the switching function to be designed should have the status that $x_e = 0$ and $\alpha_e = -\arctan(v_r y_e)$ in the sliding mode.

The switching surface vector can be obtained as:

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_e \\ \alpha_e + \arctan(v_r y_e) \end{bmatrix} . \tag{12}$$

Next, a sliding mode controller should be designed, which can make s_1 and s_2 converges to zero. The reaching law approach describes the condition under which the state converge to a sliding surface. A structure called constant rate reaching law is used to reduce the chattering phenomenon, which is the big problem of the sliding mode controller, caused by the finite time delays for computations and limitations of control [29]. For that purpose, a saturation function is used and is defined as:

$$\mathbf{sat(s)} = \begin{cases} \dfrac{s}{\xi}, & \mathbf{if}, \dfrac{s}{\xi} \leq 1 \\ \mathbf{sign(s)}, & \mathbf{if}, \dfrac{s}{\xi} > 1 \end{cases} \tag{13}$$

Where, ξ is a positive constant that define the threshold width of the saturation function. The reaching law can be obtained as:

$$\dot{s} = -ksat(s). \tag{14}$$

$$\dot{s} = \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} = \begin{bmatrix} -k_1 . sat(s_1) \cos\alpha_e \\ -k_2 . sat(s_2) \sin\alpha_e \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} y_e w - v + v_r \cos\alpha_e \\ w_r - w + \dfrac{\partial\theta}{\partial v_r}\dot{v}_r + \dfrac{\partial\theta}{\partial y_e}(-x_e w + v_r \sin\alpha_e) \end{bmatrix} \tag{16}$$

Where $\theta_e = \arctan(v_r y_e)$, $\dfrac{\partial\theta}{\partial v_r} = \dfrac{y_e}{1 + (v_r y_e)^2}$ and $\dfrac{\partial\theta}{\partial y_e} = \dfrac{v_r}{1 + (v_r y_e)^2}$.

Therefore, the control law is:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} y_e w + v_r \cos\alpha_e + k_1 sat(s_1) \\ w_r + \dfrac{\partial\theta}{\partial v_r}(\dot{v}_r) + \dfrac{\partial\theta}{\partial y_e}(v_r \sin\alpha_e + k_2 sat(s_2)) \end{bmatrix} \tag{17}$$

Finally, the expressions of the left and the right velocities applied to khepera VI mobile robot are given as follow:

$$\begin{cases} V_R = v + \dfrac{L.w}{2} \\ V_L = v - \dfrac{L.w}{2} \end{cases} \tag{18}$$

2) Fuzzy logic control: In this section, the type-0 Takagi-Sugeno fuzzy logic controller [17] is used to follow the reference path. Its inputs are the error in position (distance between the real mobile robot position and its reference position: e_p) and the error in orientation (the angle between the mobile robot orientation and its reference orientation: e_φ). The outputs are the linear and angular velocities (v and w). The predefined parameters are the tracking and stopping distances (d_T and $dstop$) and the tracking and stopping angles ($\Delta\varphi_T$ and $\Delta\varphi stop$). Figures 10 and 11 show, respectively, the membership functions of e_p and e_φ. Table I contains the inference fuzzy system for both v and w. For both linear and angular velocities three values are chosen as consequences of fuzzy rules given respectively: Z_v: Zero; S_v: Small; B_v: Big and Z_w: Zero; S_w: Small; B_w: Big. Based on physical characteristics of the mobile robot, to each linguistic variable is attributed a numerical value as shown in figures 12 and 13.

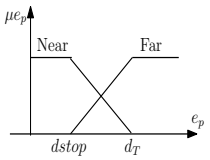

Fig. 10. Membership functions of position error e_p.

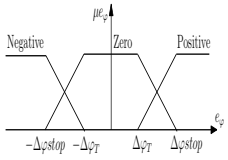

Fig. 11. Membership functions of orientation error e_φ.

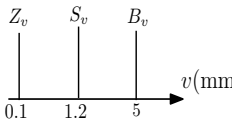

Fig. 12. Numerical values of fuzzy rules output of v.

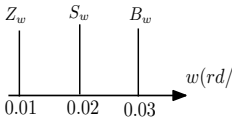

Fig. 13. Numerical values of fuzzy rules output of w.

TABLE I
LINGUISTIC INFERENCE TABLE FOR POSITION AND ORIENTATION ERRORS

e_p e_φ		Negative	Zero	Positive
$Near$	v	Z_v	Z_v	Z_v
	w	S_w	Z_w	$-S_w$
Far	v	S_v	B_v	S_v
	w	B_w	Z_w	$-B_w$

3) Simulation results: This simulation is presented to prove the effectiveness of the sliding mode controller for the path tracking compared to the fuzzy logic controller. During this simulation, the linear velocity of the non-holonomic mobile robot v is chosen to be constant, v_r = $1mm/s$. The mobile robot is assumed to start from the position (x_i,y_i) = (50,750) and to reach the target position (x_t,y_t) = (600,160). Figures 14, 15 and 16 show, respectively, the trajectory tracking result using the sliding mode controller, the tracking errors and the two wheel velocities configuration. Figures 17, 18 and 19 illustrate, respectively, the mobile robot trajectory tracking result using the fuzzy logic controller, the tracking errors and the two wheel velocities.

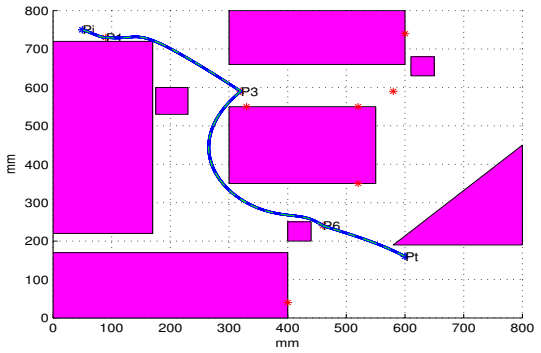

Fig. 14. Tracking reference trajectory by the sliding mode controller.

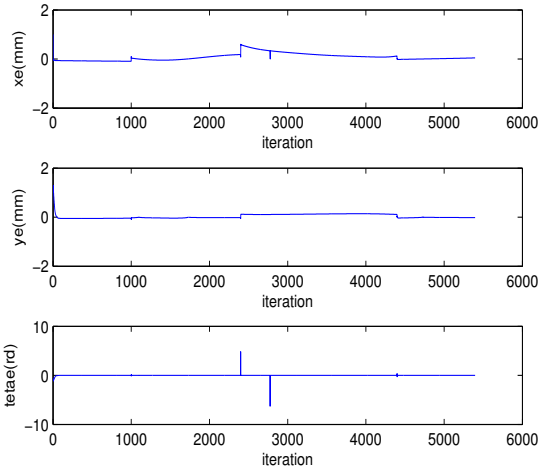

Fig. 15. Tracking trajectoriy errors with the use of sliding mode controller.

From these results, we can conclude that the sliding mode controller track more precisely the planned trajectory than the fuzzy logic controller. Thus, as shown in figure 15, the tracking errors are smallest and more stable than those shown in figure 18. Also, the configuration of the two wheel velocities left and right of the mobile robot is more stable using the sliding mode controller. as well, the sliding mode controller requires less time to reach the target than the fuzzy logic controller ($1.2s$ compared to $8.2s$). So, this simulation result proves the

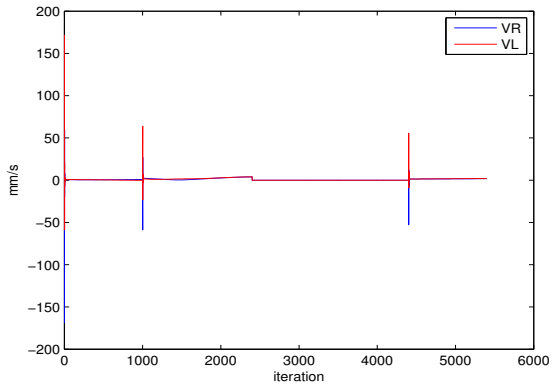

Fig. 16. The two wheel velocities left and right by the use of the sliding mode controller.

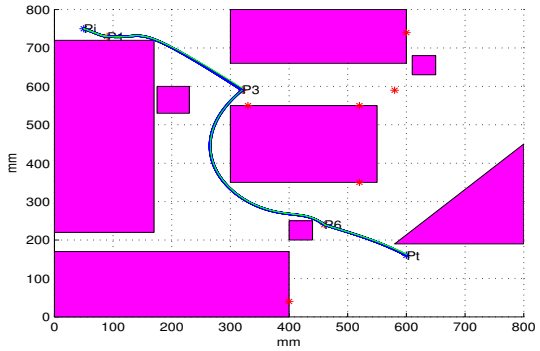

Fig. 17. Tracking reference trajectory by the fuzzy logic controller.

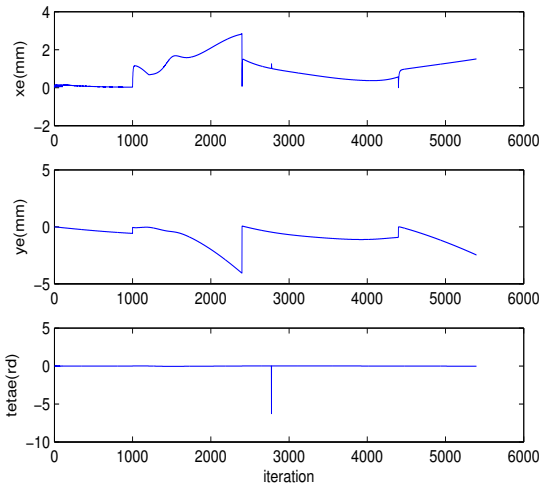

Fig. 18. Tracking trajectory errors with the use of the fuzzy logic controller.

effectiveness of our proposed tracker controller (sliding mode controller) compared to the fuzzy logic controller. Whereas, During the reference trajectory tracking, the mobile robot may encounter unexpected obstacles, since, the environment is supposed to be partially known in priori. Therefore, it must be able to leave the tracking trajectory and switch to the avoidance of unexpected obstacles which is described in the next section.

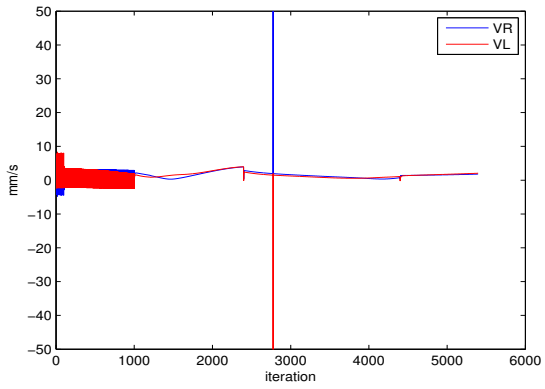

Fig. 19. The two wheel velocities left and right by the use of the fuzzy logic controller.

B. *Avoidance of unexpected obstacles*

A great amount of research in the literature was interested in the avoidance of unexpected obstacles problem. Among the methods used, we can cite for example, the fuzzy logic controller method [38], the neural networks method [16] and the hierarchical fuzzy method with multi-agent architecture [21]. In this paper, the fuzzy logic controller is chosen to avoid unexpected obstacles. In order to prove the effectiveness of our proposed controller, the avoidance of unexpected obstacles mission is also conducted by the wall-following method.

1) Wall-following method: The wall-following method [27] is based on the fuzzy logic controller [37] which is Takagi-Sugeno type of order 0 [39]. Its inputs are the lateral and frontal distances (dL, dF) as shown in figure 20 and its outputs are the linear and angular velocities (v, w). The predefined parameters are ,respectively, the minimal lateral distance, the maximum lateral distance and the minimal frontal distance ($dLmin$, $dLmax$, $dFmin$, $dFmax$). Figures 21 and 22 show the membership functions of the lateral distance (dL) and the frontal distance (dF), respectively. Table II show the inference fuzzy system for both v and w. For the linear velocity three values are chosen as fuzzy consequences presented as follow: Z_v: Zero; S_v: Small; B_v: Big and for the angular velocity two values are attributed : Z_w: Zero; B_w: Big. Based on technical tests, to each linguistic variable is attributed a numerical value as shown in figures 23 and 24.

TABLE II
LINGUISTIC INFERENCE TABLE FOR THE ERRORS IN POSITION AND ORIENTATION

d_F d_L		Near	Medium	Far
$Near$	v	Z_v	S_v	S_v
	w	$-B_w$	Z_w	$-B_w$
Far	v	B_v	B_v	B_v
	w	Z_w	B_w	B_w

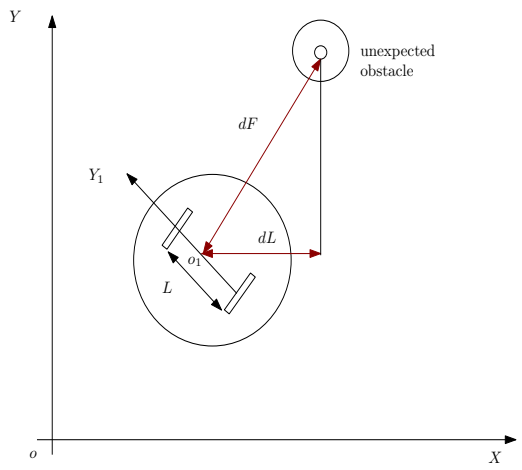

Fig. 20. The lateral and frontal distances between the mobile robot and the unexpected obstacle.

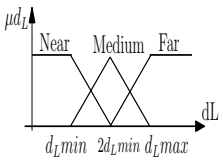

Fig. 21. Membership functions of the lateral distance.

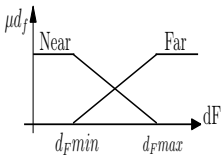

Fig. 22. Membership functions of the frontal distance.

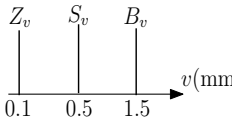

Fig. 23. Numerical values of fuzzy rules output of linear velocity v.

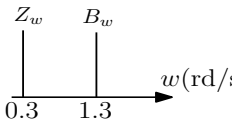

Fig. 24. Numerical values of fuzzy rules output of angular velocity w.

2) Fuzzy logic controller: In this section, our proposed controller used to avoid unexpected obstacles is presented (the fuzzy logic controller). This controller has as inputs the euclidian distance between the mobile robot and the unexpected obstacle (d) and the angle between the mobile robot direction and the obstacle direction (φ). Its outputs are the two wheel velocities left (V_L) and right (V_R) (see figure 25).

Given that (x_o,y_o) is the coordinates of the unexpected obstacle, (x,y) the current position of the mobile robot and θ_a the orientation of the unexpected obstacle with respect to

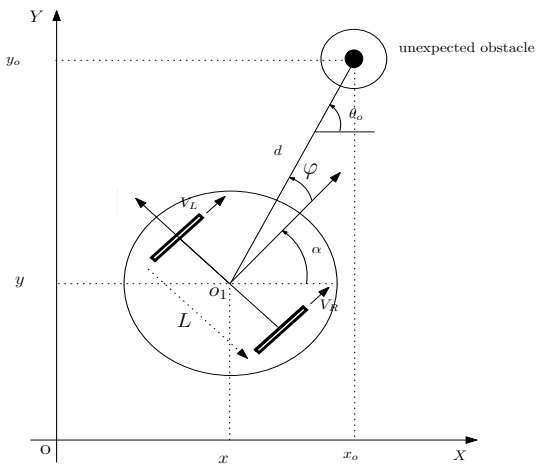

Fig. 25. Mobile robot configuration according to the unexpected obstacle.

$X-axis$, the equations of distance d and angle φ are given as follow:

$$d = \sqrt{(x_o - x)^2 + (y_o - y)^2} \quad (19)$$

$$\varphi = \theta_o - \alpha \quad (20)$$

with:

$$\theta_o = \tan^{-1}\frac{(y_o - y)}{(x_o - x)} \quad (21)$$

Tables III and IV represent the fuzzy inference system.

TABLE III
LINGUISTIC INFERENCE TABLE FOR THE RIGHT VELOCITY

V_R		φ						
		NB	NM	NS	EZ	PS	PM	PB
d	VS	B	M	S	Z	Z	S	S
	S	VB	B	M	S	S	S	M
	M	VB	VB	B	M	S	M	B
	L	VB	VB	VB	B	M	B	B
	VL	VB	VB	VB	B	B	B	B

TABLE IV
LINGUISTIC INFERENCE TABLE FOR THE LEFT VELOCITY

V_L		φ						
		NB	NM	NS	EZ	PS	PM	PB
d	VS	S	S	Z	Z	S	M	B
	S	M	S	S	S	M	B	VB
	M	B	M	S	M	B	VB	VB
	L	B	B	M	B	VB	B	VB
	VL	B	B	VB	VB	VB	VB	VB

After several simulation tests, numerical values are attributed to the consequences variables (see figure 26).

VI. SIMULATION RESULTS AND DISCUSSION

In these simulation scenarios, the linear velocity of mobile robot v is chosen to be constant, $v_r = 1m/s$.

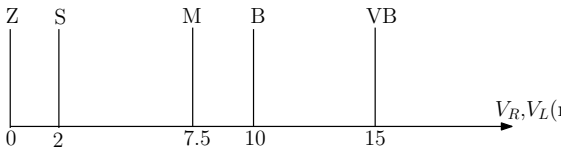

Fig. 26. Numerical values of fuzzy rules outputs.

A. Scenario 1

In this scenario, the mobile robot starts from the position (x_i, y_i) = $(50, 750)$ and reach the target position (x_f, y_f) = $(600, 160)$. Figure 27 shows the navigation mission result using the sliding mode controller for the path tracking and the wall-following method for the avoidance of unexpected obstacles with a traveling time $T = 1699.823s$ and figure 28 shows the two wheel velocities configuration using this method. Figure 29 illustrates the navigation of mobile robot using the sliding mode controller for the track and the fuzzy logic controller for the avoidance of unexpected obstacles with a traveling time $T = 1447.399s$ and figure 30 shows the two wheel velocities configuration. From these results, we can conclude that the two wheels velocities configuration of the two methods are similar, though, we can notice that in terms of traveling time, the results given by the fuzzy logic controller are the best, because it requires less time to reach the goal position.

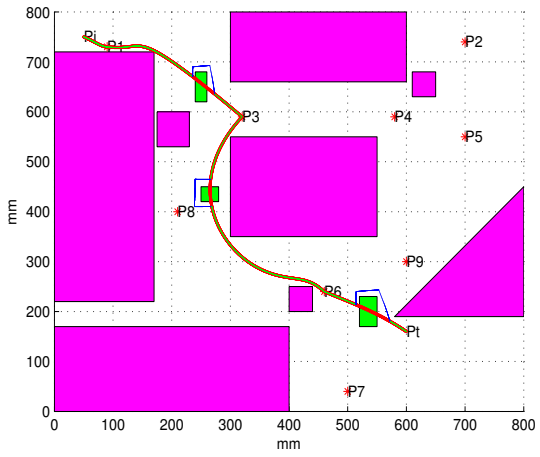

Fig. 27. Avoidance of unexpected obstacles using the wall-following method. Reference trajectory is shown in red. The tracking trajectory is shown in green. Unexpected obstacles are shown in green and the avoidance of these obstacles is shown in blue.

B. Scenario 2

In this scenario, the mobile robot is estimated to navigate in the inverse direction as that of first scenario. In fact, it starts from the point (x_i, y_i) = $(600, 160)$ and the target point is (x_f, y_f) = $(50, 750)$. Figures 31 and 32 show, respectively, the avoidance of unexpected obstacles using the wall-following method and the configuration of left and right mobile robot's wheels velocities. The elapsed time is $T = 2080.6s$. Figures 33 and 34 represent, respectively, the avoidance of unexpected

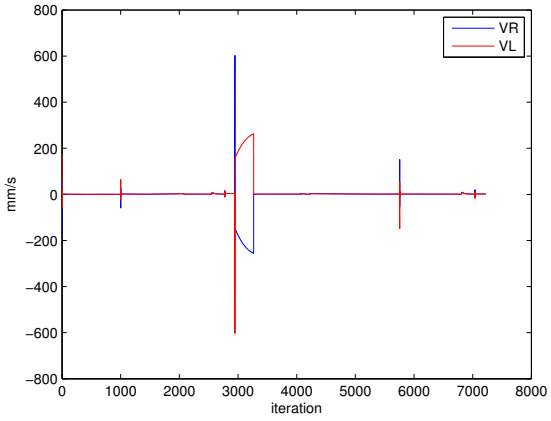

Fig. 28. The two wheels velocities : left and right.

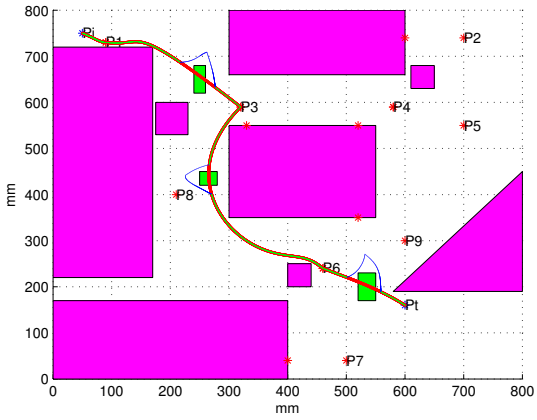

Fig. 29. Avoidance of unexpected obstacles using the fuzzy logic controller method. Reference trajectory is shown in red. The tracking trajectory is shown in green. Unexpected obstacles are shown in green and the avoidance of these obstacles is shown in blue.

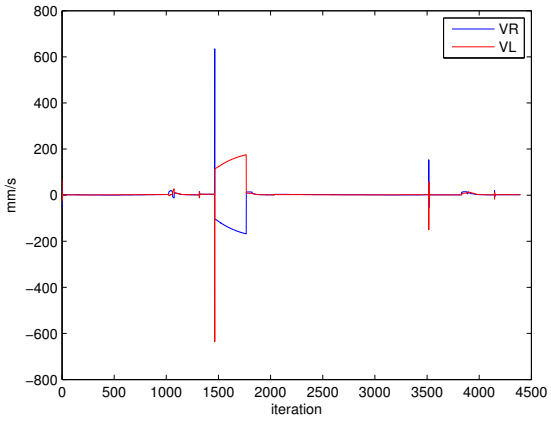

Fig. 30. The two wheel velocities : left and right.

obstacles using the fuzzy logic controller and the evolution of the two wheels velocities. The traveling time is $T = 1970.7s$. The simulation results show that the configuration of the two wheels velocities given by the two methods are almost

similar, however, the fuzzy logic controller present the smallest traveling time.

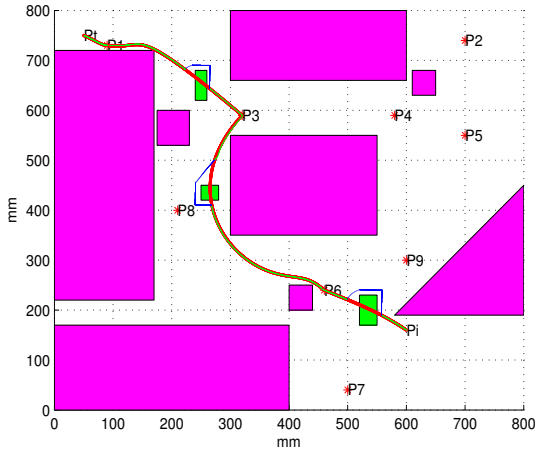

Fig. 31. Avoidance of unexpected obstacles using the wall-following method. Reference trajectory is shown in red. The tracking trajectory and the avoidance of unexpected obstacles are shown in green.

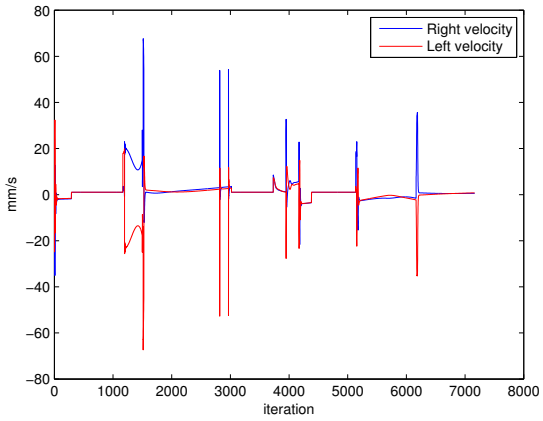

Fig. 32. The two wheel velocities : left and right.

C. Scenario 3

The task proposed in this scenario is to move the mobile robot in a cluttered environment from the start position (x_i, y_i) = $(75.12, 732)$ to the target position (x_f, y_f) = $(678.6, 101)$. Figure 35 shows the trajectory performed by the mobile robot between the initial and goal points using the sliding mode controller for the track and the wall-following method for the avoidance of unexpected obstacles. In this figure, the red stars represents the nodes, the green stars represents the control points, the red line represents the planned trajectory using the B-spline method and the green line shows the followed trajectory with avoidance of unexpected obstacles shown in blue. Figure 36 represents the tracking errors which are stable and converge to 0. Figure 37 shows the two wheels velocities configuration which is also stable. The traveling time is $T = 41.88s$. Figure 38 shows the path following using

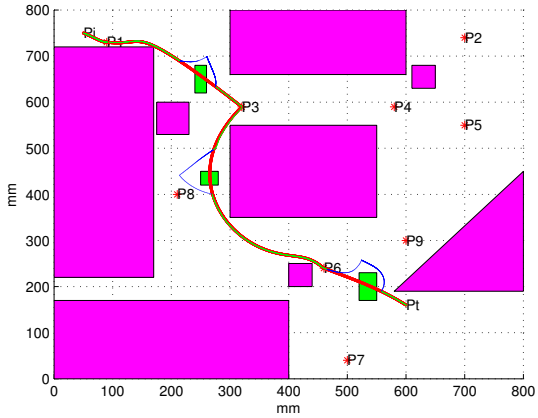

Fig. 33. Avoidance of unexpected obstacles using the fuzzy logic controller method. Reference trajectory is shown in red. The tracking trajectory and the avoidance of unexpected obstacles are shown in green.

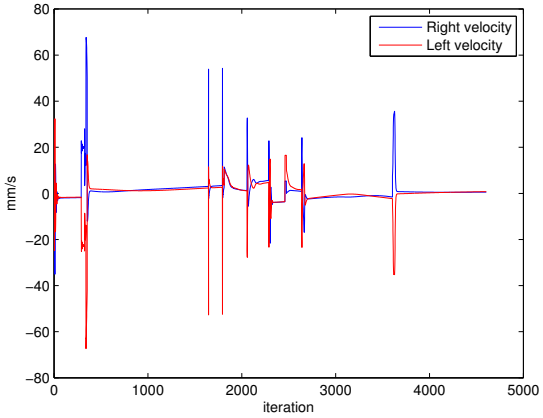

Fig. 34. The two wheel velocities : left and right.

the sliding mode controller with the avoidance of unexpected obstacles using the fuzzy logic controller. Figure 39 shows the tracking errors and the two wheels velocities are shown in figure 40. The traveling time is $T = 40.8s$. These simulation results prove the effectiveness of the proposed hybrid algorithm to perform the autonomous navigation task in cluttered environment.

D. Scenario 4

In this scenario, the obstacle is supposed to be dynamic with the speed $0.3m/s$. Figure 41 shows the dynamic obstacle avoidance and figure 42 represents the return of the robot to the track of reference trajectory. Figure 43 shows the two wheels velocities configuration which is stable. Consequently, this simulation task proves the flexibility of the proposed approach shown by its ability to react to any type of unexpected obstacle (static or dynamic).

VII. Conclusions and Future Work

In this paper, an hybrid approach for the autonomous navigation of non-holonomic mobile robot in partially-known

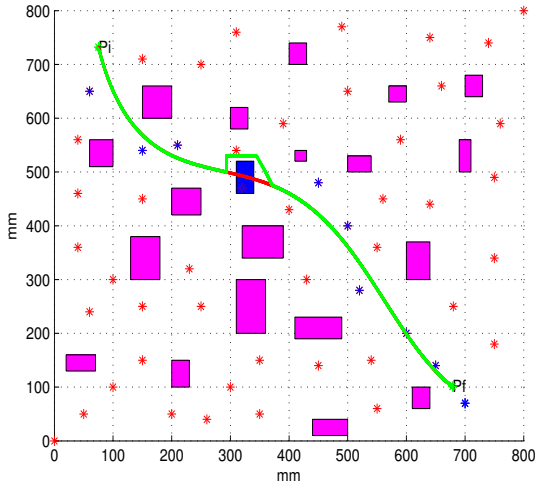

Fig. 35. Tracking reference trajectories by the sliding mode controller and avoidance of unexpected obstacle by the wall-following method.

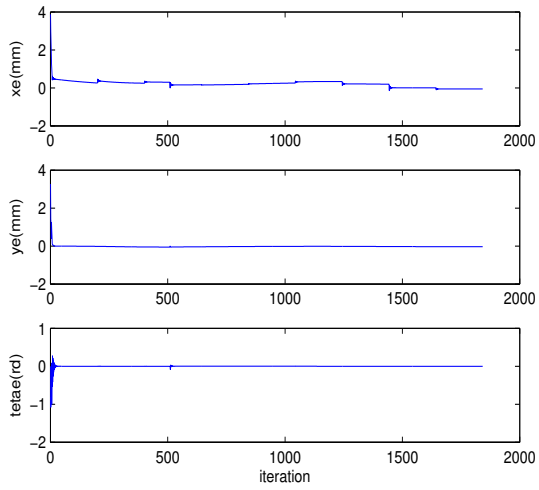

Fig. 36. Tracking trajectories errors.

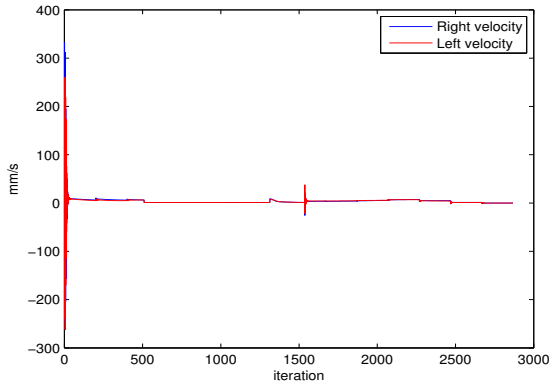

Fig. 37. The two wheels velocities left and right.

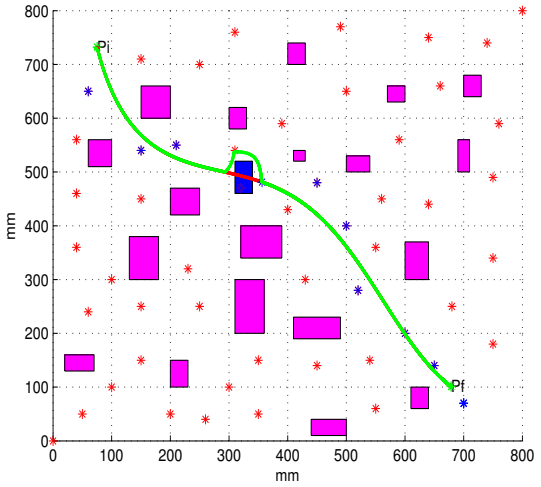

Fig. 38. Tracking reference trajectories by the sliding mode controller and avoidance of unexpected obstacle by the fuzzy logic controller.

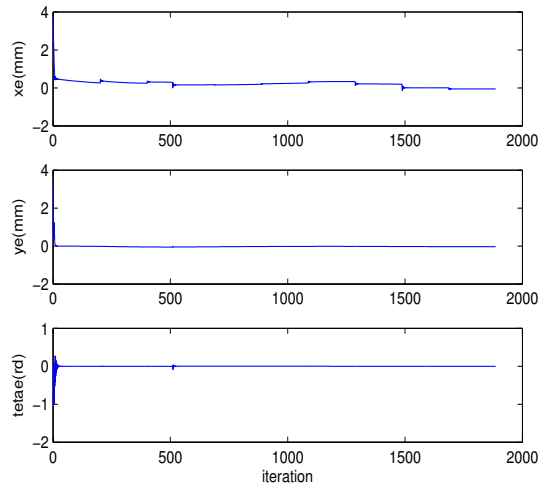

Fig. 39. Tracking trajectories errors.

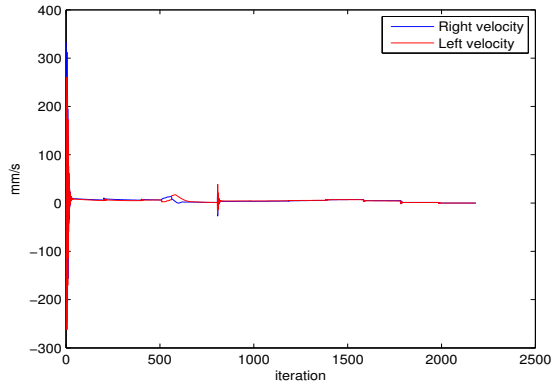

Fig. 40. The two wheels velocities left and right.

environment have been developed. The whole navigation mission have been decomposed into two main modules. The

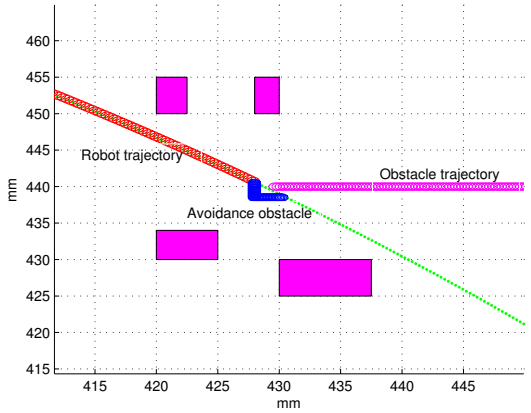

Fig. 41. The avoidance of dynamic obstacle.

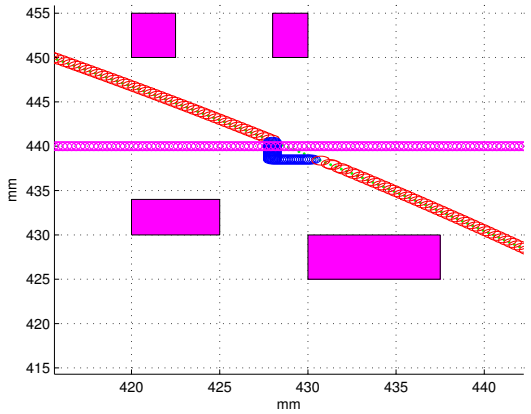

Fig. 42. The return of the mobile robot to follow planned trajectory.

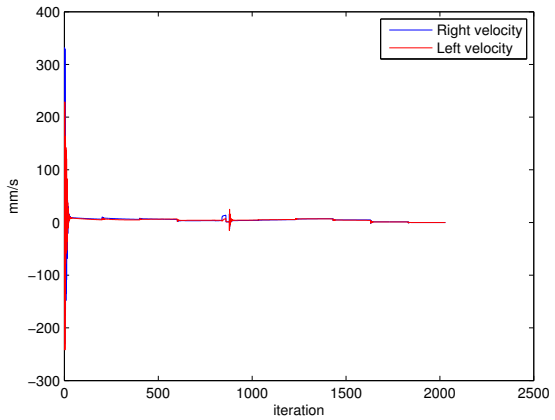

Fig. 43. The two wheel velocities : left and right.

first module has been designed to plan the distance-optimal reference trajectory, while the second one has been dedicated to the follow-up of this planned trajectory and the avoidance of unexpected obstacles. For the planning of distance-optimal reference trajectory, the B-spline method and the Dijkstra algorithm have been used. For the track of planned trajectory, the sliding mode controller has been proposed and for the avoidance of unexpected obstacles, the fuzzy logic controller has been used. Several simulation tests have been executed to evaluate the performance of the proposed control strategy by comparing the performances of the controllers proposed in our strategy with the fuzzy logic and the fuzzy wall-following controllers. Simulation results have proved the efficiency of B-spline method and Dijkstra algorithm in the generation of safe and distance-optimal reference trajectory, also, they have demonstrated that the sliding mode controller track more precisely the planned trajectory than the fuzzy logic controller. As well, the fuzzy logic control have given better results than the wall-following method in the avoidance of unexpected obstacles. Thus, the hybrid control strategy proposed composed by the B-spline, the Dijkstra, the sliding mode controller and the fuzzy logic controller, have proved its simplicity and efficiency in the management of autonomous mobile robot navigation in partially known environment.

As future work, we will focus on testing optimization algorithms in the adjustment of the manually constructed fuzzy logic controller parameters and studying cases when the mobile robot be found in critical situations when navigating, such as, when the robot is kidnapped. Also, we want to study the autonomous navigation issue in completely unknown environment.

REFERENCES

[1] F. Cherni, Ch. Rekik and N. Derbel, Mobile robot navigation based on tangent circle algorithm, International Journal of Computer Applications in Technology, 2019.
[2] Z. B. Garip, G. Atali, Path planning for multiple mobile robots in static environment using hybrid algorithm, 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2018.
[3] B. Anoohya B and R. Padhi, Trajectory Tracking of Autonomous Mobile Robots Using Nonlinear Dynamic Inversion, International Federation of Automatic Control , 2018, pp. 202-207.
[4] N. K. Goswami and P. K. Padhy, Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance, International Journal of Computers and Electrical Engineering , 2018, pp. 307-323.
[5] C. Lamini, S. Benhlima, A. Elbekri, Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning, Procedia Computer Science, 2018, pp. 180-189.
[6] I. Hassani, I. Maalej and Ch. Rekik, Robot Path Planning with Avoiding Obstacles in Known Environment Using Free Segments and Turning Points Algorithm, Mathematical Problems in Engineering, 2018.
[7] D. Ayedi, M. Boujelben and Ch. Rekik, A Multiagent Architecture for Mobile Robot Navigation Using Hierarchical Fuzzy and Sliding Mode Controllers, Mathematical Problems in Engineering, 2018.
[8] A. Pandey, D. R. Parhi, Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm, Journal of Defence Technology, 2017, pp. 47-58.
[9] M. Boukens and A. Boukabou, Design of an intelligent optimal neural network-based tracking controller for nonholonomic mobile robot systems, Neurocomputing, 2017, pp. 46-57.
[10] F. Matoui, B. Boussaid, B. Metoui, G.B. Frej and M.N. Abdelkrim, Path Planning of group of robots with potential field approach:decentralized architecture, International Federation of Automatic Control, 2017, vol. 50, pp. 11473-11478.
[11] L. Liu, C. Luo and F. Shen, Multi-agent formation control with target tracking and navigation, Information and Automation (ICIA), 2017.
[12] 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.

[13] T. T. Mac, C. Copot, D.T. Tran, R. D. Keyser, Heuristic approaches in robot path planning: A survey, Journal of Robotics and Autonomous Systems, 2016, pp. 13-28.
[14] C. Rekik, M. Jallouli and N. Derbel, Optimal Trajectory of a mobile robot using hierarchical fuzzy logic controller, International Journal Computer Applications in Technology, 2016, Vol. 53.
[15] F. Cherni, Y. Boutereaa, Ch. Rekik and N. Derbel, Path Planning for mobile robots using fuzzy logic controller in the presence of static and moving obstacles, Proceedings of Engineering and Technology (PET), 2016, pp. 503-509.
[16] D. Duguleana and G. Mogan, Neural networks based reinforcement learning for mobile robots obstacle avoidance, Expert Systems With Applications 62, 2016, pp. 104-115.
[17] M. Hank and M. Haddad, A hybrid approach for autonomous navigation of mobile robots in partially-known environments, Robotics and Autonomous Systems, 2016, pp. 113-127.
[18] S. Brown and S. L. Waslander, The constriction Decomposition Method for Coverage Path Planning, International Conference on Intelligent Robots and Systems (IROS), 2016.
[19] M. Benzaoui, H. Chekireb, M. Tadjine and A. Boulkroune, Trajectory tracking with obstacle avoidance of redundant manipulator based on fuzzy inference systems, Neurocomputing, 2016, pp. 23-30.
[20] B. Kovacs, G. Szayer, F. Tajti, M. Burdelis and P. Korondi, A novel potential field method for path planning of mobile robots by adapting animal motion attributes, Robotics and Autonomous Systems, 2016, pp. 24-34.
[21] 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, 2015, vol. 30, pp. 289-298.
[22] D. Nakhaeinia, P. Payeur, T. S. Hong and B. Karafsi, A hybrid Control architecture for autonomous mobile robot navigation in unknown dynamic environment, IEEE International Conference on Automation Science and Engineering (CASE), 2015.
[23] W.J. Yim and J.B. Park, Analysis of mobile robot navigation using vector field histogram according to the number of sectors, the robot speed and the width of the path, Control, Automation and Systems(ICCAS), 2014.
[24] A. Pandey, R. K. Sonkar, K. K. Pandey and D. R. Parhi, Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller, IEEE International Conference on Intelligent Systems and Control (ISCO), 2014.
[25] S. Chaoxia, L. Tianheng and T. Zhenmin, Motion planning by adding geometric constraint of roadside to beam curvature method, IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, 2013.
[26] M. Boujelben, C. Rekik and N. Derbel, Hierarchical fuzzy controller to avoid mobile obstacle for a mobile robot, International Multi-Conference on Systems, Signals and Devices (SSD), 2013.
[27] Fahmizal and C-H. Kuo, Development of a Fuzzy Logic Wall Following Controller for Steering Mobile Robots, International Conference on Fuzzy Theory and Its Application, 2013.
[28] Y. Deng, Y. Chen, Y. Zhang and S. Mahadevan, Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, Applied Soft Computing, 2012, pp. 1231-1237.
[29] J. Yang, R. Ma, Y. Zhang and C. Zhao, Sliding Mode Control for Trajectory Tracking of Intelligent Vehicule, International Conference on Medical Physics and Biomedical Engineering, 2012, pp. 1160-1167.
[30] El-H. Ghechi, J. Lauber, M. Dambrine, G. Klancar and S. Blazic, Discrete PDC control design with observer for trajectory tracking of non-holonomic mobile robot, IFAC Proceeding Volumes, 2009, pp. 170-175.
[31] S. K. Pradhan, D. R. Parhi and A. K. Panda, Fuzzy logic techniques for navigation of several mobile robots, Applied Soft Computing 9, 2009, pp. 290-304.
[32] 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, 2009, pp. 429-435.
[33] N. Yousfi, C. Rekik, M. Jallouli and N. Derbel, Optimized Fuzzy controller for mobile robot navigation a cluttered environment, International Multi-Conference on Systems, Signals and Devices, 2010.
[34] M. Haddad, T. Chettibi, S. Hanchi and H. E. Lehtihet, A random-profile Approach for trajectory planning of wheeled mobile robots, European Journal of Mechanics A/Solids, 2007, pp. 519-540.
[35] R. Rojas and A. G. Forster, Holonomic Control of a Robot with an omnidirectional drive, IEEE Latin American Robotics Symposium, 2006, pp. 41-46.
[36] M. Haddad, T. Chettibi, S. Hanchi and H. E. Lehtihet, A new approach for minimum time motion-planning problem of wheeled mobile robots, Triennial World Congress, Prague, 2005.
[37] L.C. Wang, L.S. Yong, M.H. Ang Jr., Hybrid of global path planning and local navigation implemented on a mobie robot in indoor environment, IEEE International Symposium on Intelligent Control, 2002, pp. 821-826.
[38] C-H. Lin and L-L. Wang, Intelligent collision avoidance by fuzzy logic control, Robotics and Autonomous Systems, 1997, pp. 61-83.
[39] T. Takagi and M. Sugeno, Fuzzy identification of Systems and its applications to modeling and control, IEEE trans. Syst. Man Cybern. 1, 1985, pp. 116-132.
[40] C. Alexopoulos and P. M. Griffin, Path Planning for a Mobile Robot, IEEE transactions on Systems, Man and Cybernetics, 1992, vol. 22, pp. 318-322.
[41] S. Kambhampati and L. S. Davis, Multiresolution Path Planning for Mobile Robots, IEEE Journal of Robotics and Automation, 1986.
[42] J. Castro, V. Santos and M. I. Ribeiro, A Multi-Loop Robust Navigation Architecture for Mobile Robots, IEEE International Conference on Robotics and Automation, 1998.
[43] M. Yamamoto, M. Iwamura and A. Mohri, Time-Optimal Motion Planning of Skeed-Steer Mobile Robots in the Presence of Obstacles, International Conference on Intelligent Robots and Systems, 1998.
[44] M. Yamamoto, M. Iwamura and A. Mohri, Quasi-Time-Optimal Motion Planning of Mobile Platforms in the Presence of Obstacles, International Conference on Robots and Automation, 1999.
[45] J-C. Latombe, Robot Motion Planning, Kluwer Academic Publishers, 1991.
[46] D. Casagrande, G. Fenu and F.A. Pellegrino, Hamiltonian Path Planning in Constrained Workspace, European Journal of Control, 2017, pp. 1-10.
[47] I. Ko, B. Kim and F.C. Park, Randomized Path Planning on Vector Fields, International Journal of Robotics Research, 2014, pp. 1664-1682.
[48] F. Janjos, R. Reichart, P. Niermeyer, Smooth Path-Generation Around Obstacles Using Quartic Splines and RRTs, International Federation of Automatic Control, 2017, pp. 9108-9113.
[49] B. Dugarjav, S.G. Lee,D. Kim, J.H. Kin and N.Y. Chong, Scan Matching online Cell Decomposition for Coverage path planning in an unknown environment, International Journal of Precision Engineering and Manufacturing, 2013, pp. 1551-1558.
[50] H. Choset, J. Burdick, Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph, International Journal of Robotics Research, 2000, pp. 96-125.
[51] C.T. Yen, M.F. Cheng, A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance, Microsystem Technologies, 2018, pp. 125-135.
[52] U.A. Syed, F. Kumvar and M. Iqbal, Guided Autowave pulse coupled Neural Network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots, International Journal of Robotics and Autonomous Systems, 2014, pp. 474-486.
[53] J. Liu, J. Yang, H. Liu, X. Tian and M. Gao, An improved ant colony algorithm for robot path planning , International Journal of Soft Computing, 2017, pp. 5829-5839.
[54] A.H. Paniagua, M.A.V. Rodriguez and J.F.N. Pavon, Solving the Multi-Objective Path Planning problem in Mobile Robotics with a Firefly-based Approach, Methodologies and Application, 2015, pp. 949-964.
[55] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, A modified Firefly Algorithm for UCAV Path Planning, International Journal of Hybrid Information Technology, 2012, pp. 123-144.
[56] G. Li and W. Chou, Path Planning for mobile robot using Self-Adaptive Learning Particle Swarm Optimization, Science China Information Sciences, 2017.
[57] L. Huang, Velocity Planning for a mobile robot to track a moving target-A Potential Field Approach, Robotics and Autonomous Systems, 2009, pp. 55-63.
[58] P.G. Zavlangas, S.G. Tzafestas, Motion Control for mobile robot obstacle Avoidance and Navigation: a fuzzy logic-based approach, Systems Analysis Modelling Simulation, 2003, pp. 1625-1637.
[59] B.K. Patle, A. Pandey, J. Anne, D.R. Parhi, Path Planning in uncertain environment by using firefly algorithm, World Journal of Engineering, 2012, pp. 123-144.