

Chattering Reduction on the Control Input of a Nonholonomic Mobile Robot Using Fuzzy Logic Controller

Faten Cherni*, Yassin Bouteraa, Chokri Rekik and Nabil Derbel

University of Sfax, National Engineering School of Sfax, Tunisia, Control and Energy Management Laboratory (CEM Lab), Tunisia

Received 19 April 2016; Accepted 2 July 2016

Abstract

This paper investigates the problem of reducing chattering in the control input for a mobile robot. It is well known that the fuzzy logic controller is an effective solution to reduce chattering for the mobile robot navigation. Compared to the original method which uses the sign function in the control input, the developed method reduces chattering and presents a simple algorithm for solving the obstacle avoidance problem in unknown dynamic environment. In fact, the developed algorithm instructs the mobile robot to keep moving smoothly to the designed target without collision. Simulation results show the efficiency of the proposed control law which reduces the chattering phenomenon and illustrate that the developed algorithm can be well applied in the mobile robot navigation.

Keywords: Obstacle avoidance, mobile robot, chattering phenomenon, unknown dynamic environment, fuzzy logic controller.

1. Introduction

Several efforts have been conducted in autonomous mobile robot researches. This is due to the replacement of humans by robots in dangerous tasks or to provide better solutions for the industry. However, designing an efficient navigation procedure for mobile robots and ensuring its security are important issues in robotics. For this reason, the mobile robot should be equipped with an adequate perception system of the environment in order to give it a reactive behavior. Such a condition is provided, the robot will be able to ensure a fluid and reactive movement to the designed target without collisions with obstacles.

In fact, the entire process that enables a robot to memorize its environment and then move to reach a goal, can be divided into three interdependent phases: mapping, localization and planning. Mapping is the phase which allows the construction of a map reflecting the spatial structure of the environment from different information collected by the robot. Such a map is available, the localization is the way of determining the orientation and the position of the mobile robot in the map with respect to its surrounding. The robot needs to know the objects (target or obstacle) around it [1]. Finally, the knowledge of the environment map and the current position of the robot, planning is the phase that allows to decide the movement in order to reach a goal in the environment. Indeed, the mobile robot needs to find a collision free path between two points (from its beginning to its end). To be able to find this path, the mobile robot should run an adequate path planning

algorithm.

Several research works, for path planning of mobile robots, have been proposed in the literature [2], [3], [4], [5], [6].

The earliest approaches for robot navigation have developed reactive methods based on sensor inputs. The three most known algorithms for local navigation are: the Artificial Potential Field method (APF) [7], [8], [9], [10], the Vector Field Histogram method (VFH) [11], [12], and the Dynamic Windows Approach [13], [14]. Additionally, there are other approaches like Curvature Velocity Method [15], Lane Curvature, Method [16], the Beam Curvature Method [17], and other methods [18], [19]. Several classical approaches dedicated to static environments are extended to dynamic ones [20], [21], [22]. However, the problem of avoiding collisions in dynamic environment is much harder.

Several works have been developed for dynamic environments like velocity obstacles [23], [24], collision cones [25], the rolling window method [26], inevitable collision state [27], the prediction model for Beam Curvature Method [28], and other methods [29], [30], [31], [32].

In the other side, several researchers have been carried out using advanced techniques such as fuzzy logic systems [33], [34], [35], [36], [37], [38]. Fuzzy Logic systems have originally been advocated by Zadeh [39], Mamdani and Assilian [40]. They become a way of collecting human knowledge and experiences. Currently, fuzzy logic is becoming an interesting topic in control engineering and a successful solution to a variety of industrial systems and consumer products. Moreover, the use of fuzzy logic systems becomes very widespread to design a robust controller satisfying autonomous navigation problems.

This paper proposes a developed method for addressing chattering phenomenon with a simple and easy implementation.

* E-mail address: fatin.cherni@gmail.com

This is realized by replacing the sign function in control input used in [41] with a fuzzy logic system. This method allows chattering reduction in control input, while keeping the robustness characteristics of the robot mobile navigation. The task is to command the mobile robot in order to avoid obstacles and reach the goal while ensuring a smoothly trajectory in a static or a dynamic environment.

This paper is organized as follows. In the next section, the problem formulation is presented. Section 3 contains the navigation algorithm. Simulation results are given in Section 4. Finally, conclusions are given in Section 5.

2. Problem Formulation

Robot navigation still represents a challenge for several applications in the real world. It aims is to develop a control law in order to find the path from the starting point to the goal avoiding collisions with obstacles. Several navigation approaches can be founded in the literature. The main idea of our work is inspired by the approach developed by Savkin and Wang in [41] and based on an integrated environment representation. In fact, the information about the environment which contains close multiple stationary and moving obstacles are included in the representation of a sensing area of the mobile robot. The approach, based on an integrated environment representation, has been implemented and tested in this work [42], [43]. In the following section, we introduce the proposed algorithm dedicated to the robot path planning in presence of static and dynamic obstacles. For that, we assume that positions of the mobile robot, the obstacles and the goal are known in advance. In order to reach the target, the designed strategy is working to solve several various issues such as: how the robot can detect and separate obstacles, how to privilege situations of obstacle avoidance to overcome, how to treat the situation with more than one obstacle... To this end, the proposed algorithm uses several assumptions to solve the matter mentioned above.

3. Navigation Algorithm

The navigation algorithm is divided into three general functions: the trajectory calculation, the fuzzy logic controller and the mobile robot model as shown in Fig. 1.

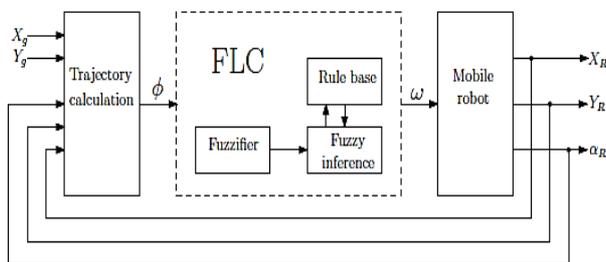


Fig. 1 Block diagram of the navigation algorithm

Notions of Fig. 1 are:

- (X_g, Y_g) represents the goal's cartesian coordinates.
- (X_R, Y_R) is the robot's cartesian coordinates and α_R is the heading direction.
- ϕ represents the new angle that changes the robot's direction.

- ω is the angular velocity of the mobile robot. (more details on ϕ and ω will be presented later)

During the actual move, the mobile robot acquires information about its environment containing static or dynamic obstacles. To achieve the goal, the robot uses the new direction angle ϕ as the main parameter. Then, the angular velocity ω is determined and given to the robot in order to be able to bypass between obstacles without being collided.

A. Modelling of the mobile robot

Several types of autonomous mobile robots have been developed to ensure a fluid and a reactive navigation. Among these types of mobile robots, we are interested by a mobile robot with two independent driving wheels which can be commanded and oriented by acting on the speed of each wheel.

We consider, as presented in Fig.2, a wheeled mobile robot for which its kinematic model is given by:

$$\begin{cases} \dot{X}_R = \frac{V_R + V_L}{2} \cos \alpha_R \\ \dot{Y}_R = \frac{V_R + V_L}{2} \sin \alpha_R \\ \dot{\alpha}_R = \frac{V_R - V_L}{L} \end{cases} \quad (1)$$

Where (X_R, Y_R) is the robot's cartesian coordinates. The heading direction α_R is taken counter-clockwise from the OX-axis. V_R and V_L are the robot right and left wheel velocities, respectively. L is the distance between the two wheels. The mobile robot is moving with a constant speed V and taking into account a maximum angular velocity equal to ω_{max} which translates the robot limits. The objective of this paper is to produce a reliable and a smooth trajectory in a static and a dynamic environment and to guide the robot towards the target direction without hitting obstacles, taking into account physical constraints of the robot.

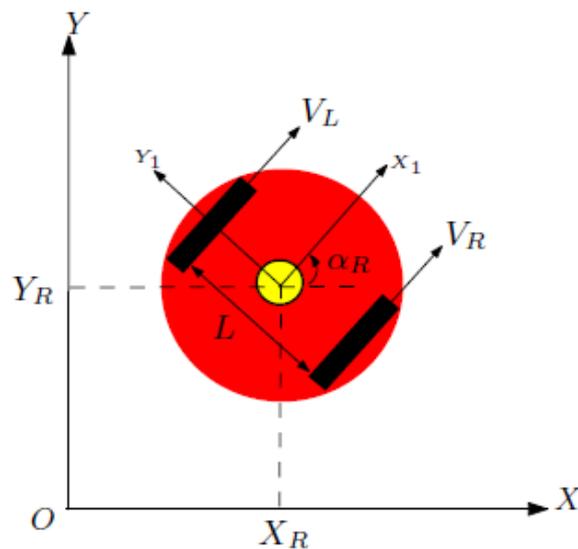


Fig. 2 The schematic model of the mobile robot

We assume that the mobile robot satisfies those initial conditions:

$$\begin{cases} X_R(0) = X_0 \\ Y_R(0) = Y_0 \\ \alpha_R(0) = \alpha_0 \end{cases} \quad (2)$$

Let τ be the sampling rate and let $\alpha_g \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ the desired goal direction assumed to be known to the robot and (X_g, Y_g) are the goal's cartesian coordinates. Then:

$$\alpha_g = \arctan\left(\frac{Y_g - Y_R}{X_g - X_R}\right) \quad (3)$$

Based on the computation angular velocity ω , we calculate the robot right and left wheel velocities as following:

$$\begin{cases} V_L = V + \frac{1}{2} L\omega \\ V_R = V - \frac{1}{2} L\omega \end{cases} \quad (4)$$

B. Trajectory calculation

In the following, we present the trajectory calculation which contains the algorithm in order to determine the angle φ . We assume that all obstacles are circles in order to facilitate the present work. We define the disc C of the radius R centred at the point Ω that is ahead of the mobile robot's position as shown in Fig. 3.

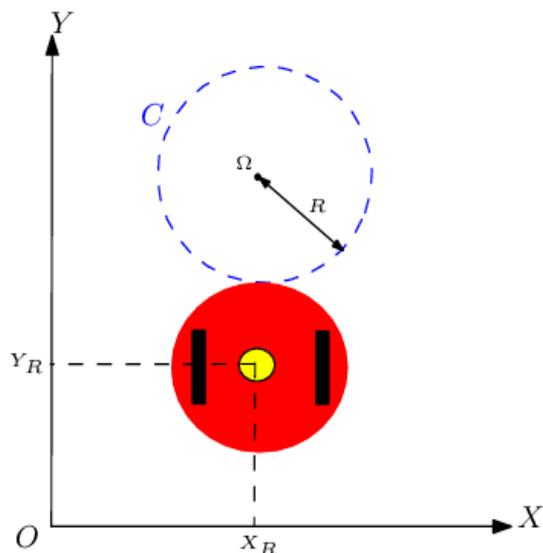


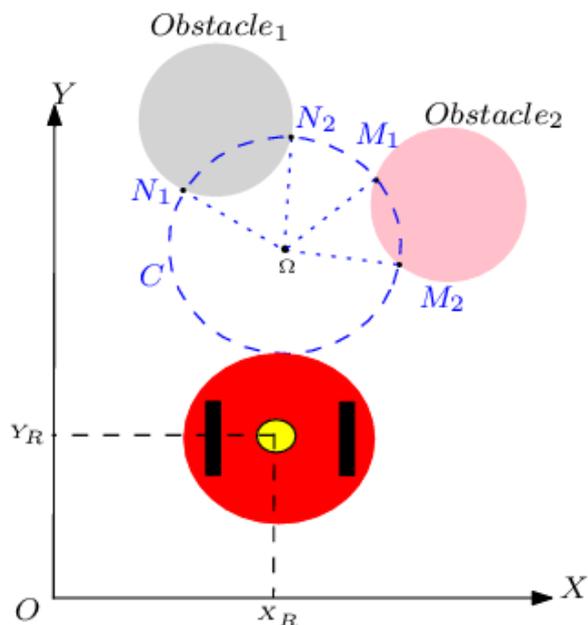
Fig. 3 Representation of the disc

The geometric sense of the disc C choice is to ensure an efficient detection of obstacles. Indeed, the geometric shape covers the entire area in front of the robot. Really, the developed obstacle avoidance approach looks for having points of the intersection between virtual disc and real obstacles (see Fig.4a). In Fig.4b we have two intersection points $N_1(X_{N1}, Y_{N1})$ and $N_2(X_{N2}, Y_{N2})$. The angle θ_1 is given by:

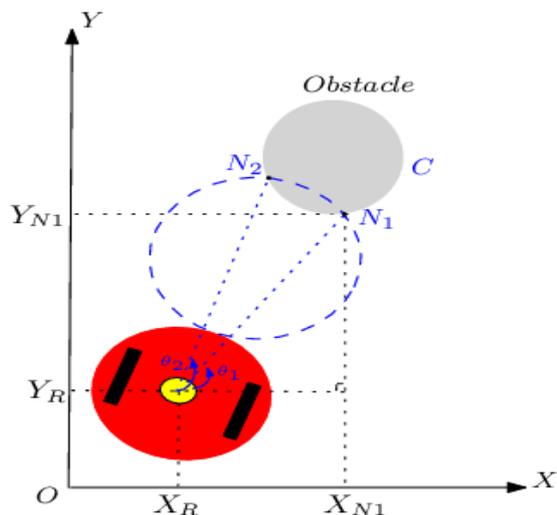
$$\theta_1 = \arctan\left(\frac{Y_{N1} - Y_R}{X_{N1} - X_R}\right) \quad (5)$$

The idea is to compute the angle that makes the disc C with two intersection points of obstacles. Based on the computation angle, a new direction φ close to α_R will be provided. The objective behind the calculation of the new direction is to change the robot's heading in order to avoid obstacles detected in front. That remains now is to move towards the goal. To this end, the angular velocity, that guides the robot to the target direction without hitting obstacles, should be determined by the fuzzy logic controller.

The flowchart algorithm presented in Fig.5, contains four steps which are explained in the following:



(a) Intersection points



(b) Definition of angle θ

Fig. 4 Obstacles detection

Step 1: Let θ_T the set which contains all values of θ calculated by Eq.5.

$\theta_T = \{\theta_1, \theta_2, \dots, \theta_{j-1}, \theta_j\}$ where $j \in \{1, 2, \dots, 2n\}$ and n is the number of obstacles. Fig.6 illustrates the distribution of different intervals.

Step 2: We note I_{nd} the index of the angle that is closest to the robot current heading α_R .

$$I_{nd} = \arg \min(|\theta_T - \alpha_R|) \quad (6)$$

Moreover we note:

$$S_1 = \alpha_R - \frac{\pi}{2}$$

$$S_2 = \theta_T (I_{nd} - 1)$$

$$S_3 = \alpha_R + \frac{\pi}{2}$$

$$S_4 = \theta_T (I_{nd} + 1)$$

$$S_5 = \theta_T (I_{nd})$$

Step 3: If the robot is in front of obstacles, there are four cases depending on the value of the index I_{nd} :

- If I_{nd} is odd:
 - Case 1: If $I_{nd} = 1$ then $\varphi = S_1$
 - Case 2: If $I_{nd} \neq 1$ then $\varphi = S_2$
- If I_{nd} is even:
 - Case 3: If $I_{nd} = 2n$ then $\varphi = S_3$
 - Case 4: If $I_{nd} \neq 2n$ then $\varphi = S_4$

Step 4: Based on the computation angle φ , we compute the new direction:

$$\gamma = \begin{cases} \frac{S_5 + \varphi}{2} & \text{if } \text{length}(\theta_T) \neq 0 \\ \alpha_g & \text{elsewhere} \end{cases} \quad (7)$$

If $\text{length}(\theta_T) = 0$, the direction γ represents the middle of the interval closest to α_R . If there isn't any obstacles in front of the mobile robot, then γ represents the desired goal direction given by Eq.3.

C. Fuzzy logic controller

The Fuzzy logic controller is simple to design, easy to implement and has several robust properties. Thus, it can be used to control the navigation behavior of the mobile robot. In fact, the fuzzy system allows the mobile robot to find the path from the starting point to the target. To increase the autonomy and the robustness of the mobile robot, we can use one or several fuzzy logic controllers.

A typical fuzzy logic controller includes a fuzzifier, membership functions, control rule base, an inference engine, and a defuzzifier.

In our case, the controller input is the angle φ calculated by the trajectory calculation and represents the new direction that makes the robot changes its direction when it senses obstacles in front.

The controller output is the angular velocity ω that will be given to the mobile robot in order to guide the robot smoothly to the target without hitting obstacles.

1) Fuzzy partition of input variables:

The angle φ is defined in $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Membership functions of the parameter φ is Gaussian (see Fig. 7) and given by the following expression:

$$\text{Gauss}(x, \sigma, a) = \exp \left[-\left(\frac{x-a}{\sigma} \right)^2 \right] \quad (8)$$

From several experiments and from different desired precision, we have associated seven linguistic values for the angle φ (NL: Negative Large; NM: Negative Medium; NS: Negative Small; Z: Zero; PS: Positive Small; PM: Positive Medium; PL: Positive Large).

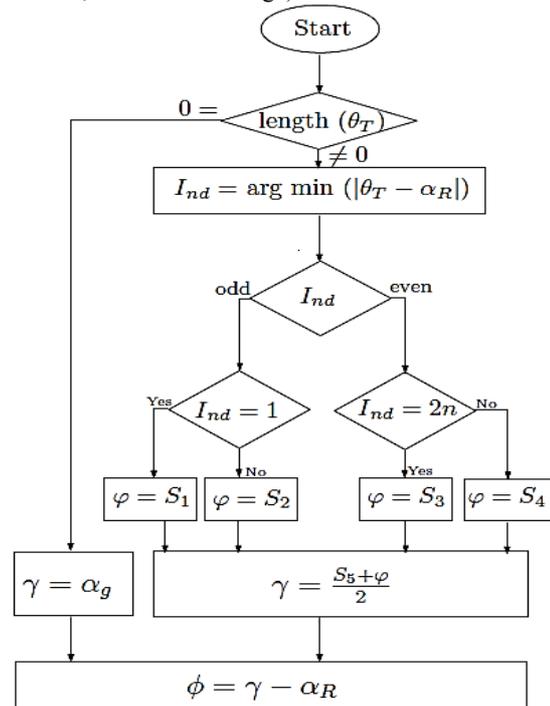


Fig. 5 Block diagram of the navigation algorithm

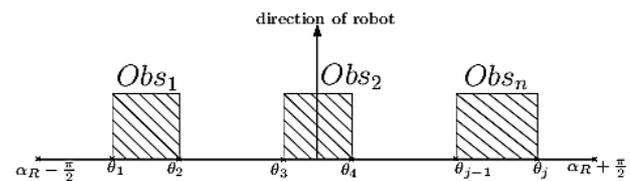


Fig. 6 Illustration of the intervals

2) Rule basis:

In this part, we determine relations between the fuzzy input variable φ and fuzzy output variable ω . These rules are given by:

If (φ is A_i) then ($\omega = y_i$)

with $i = 1, 2, \dots, m$, where m is the rule number.

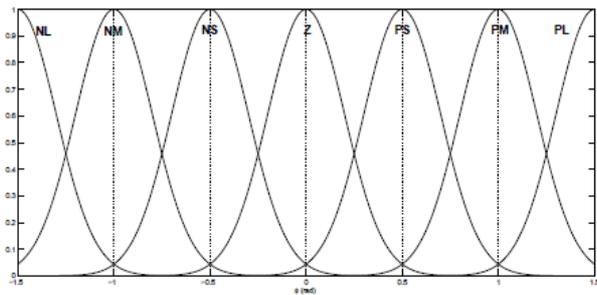


Fig. 7 Membership functions for the angle ϕ (in rad)

Following several simulations and experiment tests, we manually constructed the fuzzy inference table (situation/action). Tab.1 represents the suggested fuzzy rules.

Table 1. Inference table for the angle ω

ϕ	NL	NM	NS	Z	PS	PM	PL
ω	NB	NM	NS	Z	PS	PM	PB

3) Fuzzy controller outputs:

The fuzzy controller output is the angular velocity ω . It is defined by:

$$\omega = \frac{\sum_{i=1}^m \alpha_i y_i}{\sum_{j=1}^m \alpha_j} \tag{9}$$

with α_i is the level activation of rule i .

The elaborated fuzzy controller is a Sugeno fuzzy logic system of order zero. Thus, fuzzy rule consequences are constants. We have chosen seven values as linguistic variables of the fuzzy rule consequences as following: NL: Negative Large; NM: Negative Medium; NB: Negative Big; Z: Zero; PS: Positive Small; PM: Positive Medium; PB: Positive Big. Based on simulations and experimentation tests, we have attributed to each linguistic variable a numerical value as shown in Fig. 8.

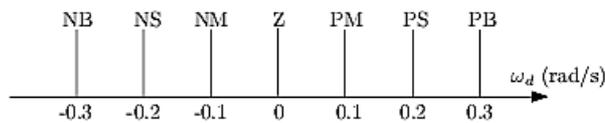


Fig. 8 Numerical values of fuzzy rules output

4. Simulation Results

To conclude the performances of the developed method using the principle of the fuzzy logic controller, we will present simulations of an arbitrarily environment including static obstacles. In all simulations, the linear velocity and the maximum angular velocity have been chosen respectively as $V = 0.2 \text{ ms}^{-1}$, $\omega_{\max} = 0.3 \text{ rad.s}^{-1}$. The sampling rate is set to be $\tau = 0.1\text{s}$.

A. Navigation with static obstacles

In order to prove that the developed approach is robust and

efficient in partially known environment, we have constructed a virtual environment containing static obstacles. In this scene, five static obstacles, with different shapes, are placed with an arbitrary way in front of the robot path and between the robot initial position and its goal. We assume that the robot start its motion from the initial position $[X_R(0), Y_R(0), \alpha_R(0)] = [0, 0, \frac{\pi}{2}]$. The robot should begin at point S (0, 0) and finish at point G (0, 12). In such a crowd environment the developed algorithm is used to guide the mobile robot to fulfill the task.

Fig.9 and Fig.10 illustrate the mobile robot trajectory depicted with small circles. Once the mobile robot senses the obstacle, it changes its heading towards a safe direction in order to avoid collision with obstacles. As we can see in both figures, the robot is able to move around static obstacles or to pass through the middle of the gap between them.

B. Navigation with moving obstacles

In this case, the robot is navigating with 3 dynamic obstacles starting its motion as mentioned in Tab.2. We assume that the robot start its motion from the initial position $[X_R(0), Y_R(0), \alpha_R(0)] = [0, 0, \frac{\pi}{4}]$. The robot should begin at point S (0, 0) and finish at point G (8,10).

In such a crowd environment the developed algorithm is used to guide the mobile robot to fulfill the task. In the following simulations, we show different scenarios illustrating the mobile robot moving towards the goal in a dynamic environment. As it can be seen in Fig. 11 and Fig.12, the robot tries to detour the moving obstacles from its front and changes its direction when it detects the obstacle. Finally, the mobile robot accomplishes successfully the navigation mission and reaches the stationary goal.

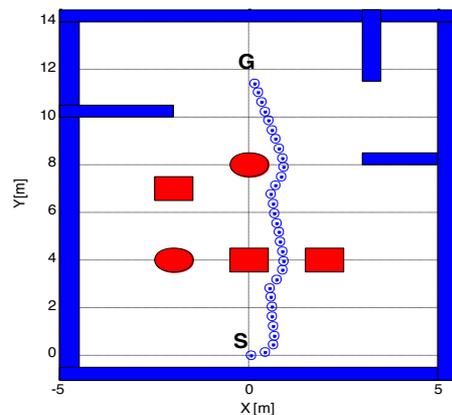


Fig. 9 Path planning with original method [41] in static environment starting at (0,0) and ending at (0,12).

Moreover, we define $\cdot S = s_2 - s_1$ the covered distance between the starting and the ending points. In fact, in the original method, the robot spends a long time to attain the target which results in a much longer path $\cdot S = 17.68\text{m}$.

While in the developed method, the travelled distance to the goal is $\cdot S = 14.82\text{m}$. Indeed, due to the developed algorithm, the path is optimized and the robot seeks with the shortest path to the destination and the lowest elapsed time

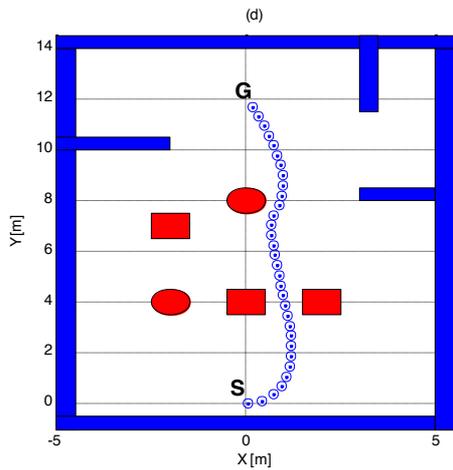


Fig. 10 Path planning with developed method using fuzzy logic controller in static environment starting at (0,0) and ending at (0,12).

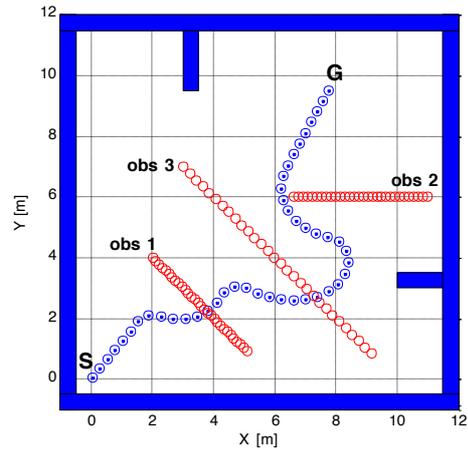


Fig. 11 Mobile robot navigation with originally method [41] in dynamic environment (17.68 m path length)

C. Chattering problem

The chattering phenomenon is an undesirable phenomenon that generates oscillations in the control input which can result the deterioration anticipated of the control system.

Table 2 Initial conditions of mobile obstacles

	x_{obs}	y_{obs}	θ_{obs}	V_{obs}
<i>obstacle</i> ₁	2	4	$-\frac{\pi}{4}$	0.05
<i>obstacle</i> ₂	13	6	$-\pi$	0.05
<i>obstacle</i> ₃	4	8	$-\frac{\pi}{4}$	0.1

In order to prove that there are no collision between the robot and obstacles, we illustrate in Fig. 13 the curves of the robot's cartesian coordinates (X_R, Y_R) depicted with continuous line and the curves of cartesian coordinates of different obstacles represented by dotted lines. If there is a collision between the robot and obstacles, they will have the same cartesian coordinates at the same time. Observing Fig.13, it is easy to conclude that there is no collision between the robot curves and obstacles. This proves that the robot moves away from mobile obstacles and doesn't collide them.

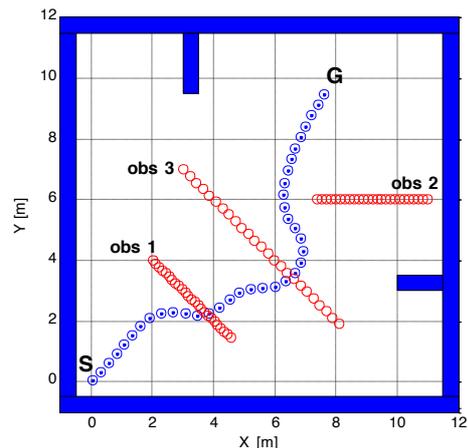
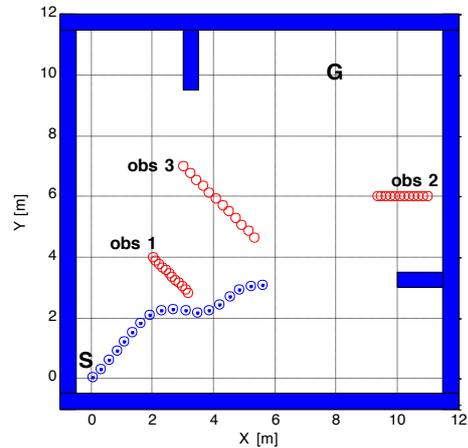
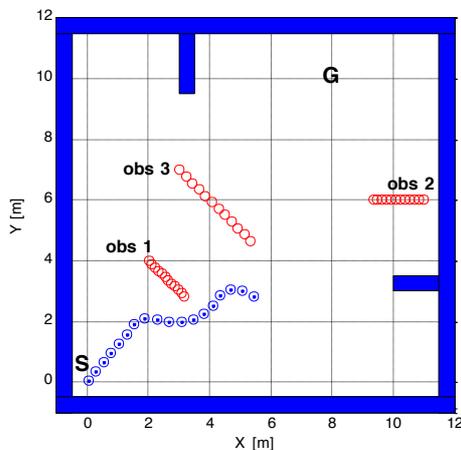


Fig. 12 Mobile robot navigation using fuzzy logic controller in dynamic environment (14.82 m path length)

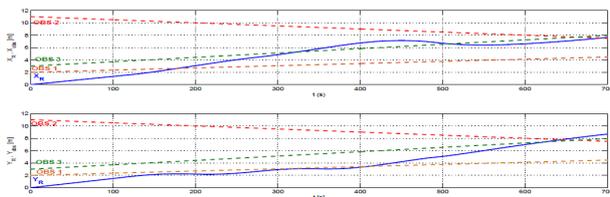


Fig. 13 Cartesian coordinates curves of the robot and obstacles

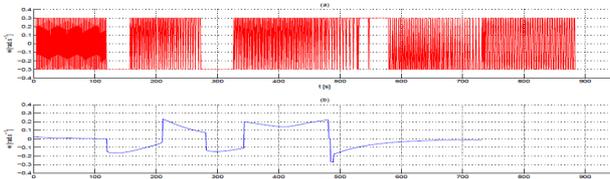


Fig. 14 Control input in static environment: (a) original method (b) developed approach

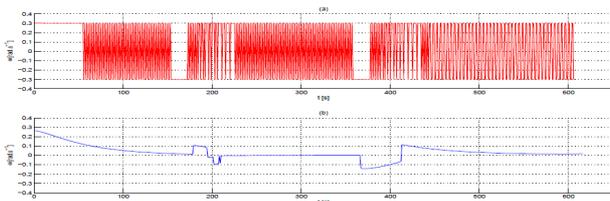


Fig. 15 Control input in dynamic environment: (a) original method (b) developed approach

To limit this phenomenon, the fuzzy logic controller is introduced to replace the sign function and to solve the chattering problem in order to provide the stability and the robustness of the system.

In fact, the switching caused by the sign function presented in Eq.10 [41] involves the appearance of the chattering phenomenon which is characterized by high

frequencies (see Fig. 14.a and Fig. 15.a).

$$\begin{aligned} \text{if } m(k\delta) = 0 \text{ then } u(t) &= U_{max} \text{ sign} [\theta_0(t) - \theta(t)] \\ \text{if } m(k\delta) = 1 \text{ then } u(t) &= U_{max} \text{ sign} [C(t) - \theta(t)] \end{aligned} \quad (10)$$

$$\forall t \in [k\delta, (k+1)\delta)$$

Where:

$m(k\delta)$ is a function taking 0 when the mobile robot does not sense the environment in front itself or 1.

when the mobile robot senses the environment. $u(t)$ the angular velocity, U_{max} the maximum angular velocity. θ_0 represents the desired goal direction and $C(t)$ the new direction. δ is the sampling period.

Simulations results, given by Fig.14.b and Fig.15.b, prove that the chattering phenomenon has been eliminated from the control command of the angular velocity and show clearly that the use of the fuzzy logic controller gives good performances and reduces the chattering phenomenon.

5. Conclusion

In this paper, we have proposed a developed algorithm based on an integrated representation of the environment.

The aim has been to reduce chattering phenomenon in the control law using the fuzzy logic controller compared to the original method which uses the sign function. Based on the developed algorithm, the mobile robot is able to find a collision-free path between the starting point and the goal in cluttered environment containing static and moving obstacles.

Simulation results prove that the developed algorithm proves a high effectiveness in obstacle avoidance and for the convergence to the target.

References

1. D. Filliat, J.A. Meyer, Map based navigation in mobile robots: I. A review of localization strategies, *Cognitive System Research*, 2003, 243–82.
2. Z. Cai, Z. Peng, Cooperative coevolutionary adaptive genetic algorithm in path planning of cooperative multi-mobile robot systems, *Journal of Intelligent and Robotic Systems*, 33(1), 2002, 61–71.
3. J. Liang, Ch. Lee, Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm, *Advances in Engineering Software*, 79, 2015, 47–56
4. I. Nishitani, T. Matsumura, M. Ozawa, A. Yorozu, M. Takahashib, Human-centered X-Y-T space path planning for mobile robot in dynamic environments, *Robotics and Autonomous Systems*, 66, 2015, 18–26.
5. YH. Liu, S.Arimoto, Path planning using a tangent graph for mobile robots among polygonal and curved obstacles, *The International Journal of Robotics Research*, 11, 1992, 76–382.
6. X. Zhong, X. Zhong, X. Peng, Velocity-Change-Space-based dynamic motion planning for mobile robots navigation, *Neurocomputing*, 143, 2014, 153–163.
7. O.Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *The international journal of robotics research*, 1985, 500–505.
8. SS. Ge, YJ. Cui, Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3), 2002, 207–221.
9. E. Masehian, MR. Amin-Naseri, A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning, *Journal of Robotic Systems*, 21(6), 2004, 275–300.
10. SR. Amada, PR. Vundavilli,DK. Pratihar, Adaptive vs. Conventional potential field approaches for solving navigation problems of a real car-like wheeled robot, *International Journal of Intelligent Defence Support Systems*, 2(4), 2009, 290–318.
11. J. Borenstein, Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation*, 7(3), 1991 278–288.
12. YJ. Kim, JH. Kim, DS. Kwon, Evolutionary programming-based uni-vector field navigation method for fast mobile robots, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(3), 2001, 450–458.
13. D. Fox, W. Burgard, S. Thrun, The Dynamic Window Approach to Collision Avoidance, *IEEE Robotics and Automation Magazine*, 4(1), 1997, 23–33.
14. O. Brock, O. Khatib, High-speed navigation using the global dynamic window approach, *IEEE International Conference on Robotics and Automation*, 1, 1999, 341–346.
15. R. Simmons, The Curvature Velocity Method for Local Obstacle Avoidance, *IEEE International Conference on Robotics and Automation Magazine*, 1996, 2275–2282.
16. NY. Ko, RG. Simmons, The Lane Curvature Method for Local Obstacle Avoidance, *Intelligent Robots and Systems*, 1998.
17. JL. Fernandez, R. Sanz, JA. Benayas, Improving collision avoidance for mobile robots in partially known environments: The Beam Curvature Method, *Robotics and Autonomous Systems*, 46(4), 2004, 205–219.
18. PP. Jamdagni, K. Patra, Obstacle avoidance for mobile robot navigation in unknown environment using geometrical information of mobile camera images, *International Journal of Computational Vision and Robotics*, 4, 2014, 39–54.

19. K. Park, N. Zhang, Duel fuzzy logic controllers design for autonomous robot navigation, *International Journal of Systems, Control and Communications*, 3(3), 2011, 322–329.
20. SS. Ge, YJ. Cui, Dynamic motion planning for mobile robots using potential field method, *Autonomous Robots*, 13(3), 2002, 207–233.
21. D. Hsu, R. Kindel, JC. Latombe, S. Rock, Randomized kinodynamic motion planning with moving obstacles, *The International Journal of Robotics Research*, 21(3), 2012, 233–255.
22. JP. Van Den Berg, MH. Overmars, Roadmap-based motion planning in dynamic environments, *IEEE Transactions on Robotics*, 21(5), 2005, 885–897.
23. P. Fiorini, Z. Shiller, Motion planning in dynamic environments using velocity obstacles, *The International Journal of Robotics Research*, 17, 1998, 760–772.
24. F. Large, C. Lauger, Z. Shiller, Navigation among moving obstacles using the NLVO: principles and applications to intelligent vehicles, *Autonomous Robots*, 19, 2005, 159–171.
25. A. Chakravarthy, D. Ghose, Obstacle avoidance in a dynamic environment: a collision cone approach, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28(5), 1998, 562–574.
26. CG. Zhang, YG. Xi, Rolling path planning and safety analysis of mobile robot in dynamic uncertain environment, *Control Theory Application*, 20(1), 2003, 37–44
27. [27] T. Fraichard, H. Asama, Inevitable collision states. A step towards safer robots, *Advanced Robotics*, 2004, 388–393.
28. Ch. Shi, Y. Wang, J. Yang, A local obstacle avoidance method for mobile robots in partially known environment, *Robotics and Autonomous Systems*, 58, 2010, 425–434.
29. MAK. Jaradata, M.Al-Rousan, L. Quadan, Reinforcement based mobile robot navigation in dynamic environment, *Robotics and Computer-Integrated Manufacturing*, 27, 2011, 135–149.
30. W. Yaonan, Y. Yimin, Y. Xiaofang, Z. Yi, Z. Yuanli, Y. Feng, Autonomous mobile robot navigation system designed in dynamic environment based on transferable belief model, *Measurement*, 44, 2011, 1389–1405.
31. Q. Zhu, J. Hu, W. Cai, L. Henschen, A new robot navigation algorithm for dynamic unknown environments based on dynamic path re-computation and an improved scout ant algorithm, *Applied Soft Computing*, 11, 2011, 4667–4676.
32. R. Bis, H. Peng, AG. Ulsoy, Velocity occupancy space: autonomous navigation in an uncertain, dynamic environment, *International Journal of Vehicle Autonomous Systems*, 10, 2012, 41–66.
33. A. Fahmi, L. Khriji, A. Al-Yahmadi, N. Masmoudi, Contribution of fuzzy logic based autonomous robot navigation in unknown environment, *Proceeding of IEEE conference on Signals System and Decision*, 2007.
34. H. Maaref and C. Barret, Sensors-based fuzzy navigation of an autonomous mobile robot in an indoor environment, *Control Engineering Practice*, 38, 2002, 1–18.
35. M. Sugeno, M. Nishida, Fuzzy control of a model car, *Fuzzy Sets and systems*, 16, 1985, 103–113.
36. M. Boujelben, Ch. Rekik and N. Derbel, Hierarchical fuzzy controller for a nonholonomic mobile robot, *Mediterranean Conference on Control and Automation*, 2012, 341–347.
37. M. Amoozgar, S. Sadati, and Kh. Alipour, Trajectory tracking of wheeled mobile robots using a kinematical fuzzy controller, *International Journal of Robotics and Automation*, 2012.
38. F. Abdessemed, Kh. Benmahammed, E. Monacelli, A fuzzy-based reactive controller for a non-holonomic mobile robot, *Robotics and Autonomous Systems*, 47, 2004, 31–46.
39. LA. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy sets and systems*, 8, 1965, 338–353.
40. EH. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, *Proceedings of the Institute of Electrical Engineers*, 121, 1974, 1585–1588.
41. AV. Savkin, Ch. Wang, Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation, *Robotics and Autonomous Systems*, 62, 2014, 1568–1580.
42. F. Cherni, Y. Boutereaa, Ch. Rekik, N. Derbel, Autonomous mobile robot navigation algorithm for planning collision-free path designed in dynamic environments, 9th Jordian Int. Electrical and Electronics Engineering Conference (JIEEEEC), 2015.
43. F. Cherni, M. Boujelben, L. Jaiem, Y. Boutereaa, Ch. Rekik, N. Derbel, Autonomous mobile robot navigation based on an integrated environment representation designed in dynamic environments, *Int. Journal of Automation and Control*, 2015, (under review).