

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/287718129>

Hybrid Method for the Navigation of Mobile Robot Using Fuzzy Logic and Spiking Neural Networks

Article in *International Journal of Intelligent Systems and Applications* · November 2014

DOI: 10.5815/ijisa.2014.12.01

CITATIONS

4

READS

48

3 authors, including:



[Mami Mohammed Amine](#)

Université Oran1 Ahmed Ben Bella

3 PUBLICATIONS 6 CITATIONS

SEE PROFILE

Hybrid Method for the Navigation of Mobile Robot Using Fuzzy Logic and Spiking Neural Networks

Zineb LAOUICI*, **

* RIIR Laboratory

**Faculty of Exact and Applied Sciences-University of Oran, Algeria

Email: zinebl90@gmail.com

Mohammed Amine MAMI*, **

Email: aminehome@yahoo.fr

Mohamed Fayçal KHELFI*, **

Email: mf_khelfi@yahoo.fr

Abstract— the aim of this paper is to present a strategy describing a hybrid approach for the navigation of a mobile robot in a partially known environment. The main idea is to combine between fuzzy logic approach suitable for the navigation in an unknown environment and spiking neural networks approach for solving the problem of navigation in a known environment. In the literature, many approaches exist for the navigation purpose, for solving separately the problem in both situations. Our idea is based on the fact that we consider a mixed environment, and try to exploit the known environment parts for improving the path and time of navigation between the starting point and the target. The Simulation results, which are shown on two simulated scenarios, indicate that the hybridization improves the performance of robot navigation with regard to path length and the time of navigation.

Index Terms— Mobile Robot, Navigation, Fuzzy Logic, Spiking Neural Networks, Unknown Environment

I. INTRODUCTION

Navigation is one of the most challenges in the mobile robot theory. It has several problems to be solved: how to avoid obstacles, how to find and construct a collision-free path from a starting point to the target in known, partially known or totally unknown environments. These challenges are related to the use of mobile robot, as the requirements and the uses of mobile robot extend, as the robot's navigation will be face to more challenges.

In order to solve these problems and have an autonomous mobile robot, which will be able to navigate and react in the environment properly, several methods are developed in literature [1-2-3-6-7-9-10]. Each developed method is used to solve the problem of navigation. Our aim in this study is to build a method, which improves the robot's navigation in partially known environment. To choose between the classical methods and the new methods we have done a comparison between the Fuzzy Logic (FL) method and Bug2 method [3-4]. This comparison shows the utility of FL approach.

The fuzzy logic technique is used in many works to deal the navigation task. In [6] the author use fuzzy logic to build elementary controllers, but his study focused also on how to coordinate the elementary behaviors, a comparison is done between two proposed methods to select the best. The author of the both papers [7-8] proposed an approach for the fuzzy logic based control by multi-sensor integration. According to this approach, the fuzzy reasoning performs the coordination for the avoiding obstacles, following edges and moving towards target behaviors. In addition, in [14] the author focused on developing a fuzzy rule based system, his strategy has the capability to build and update a topological map. The Fuzzy logic method has the same reaction even there are known areas, as results the FL method cannot benefit from these known areas to ameliorate the robot's navigation.

This study [15] proposed a new local navigation technique uses the avoidance obstacle technique. The author in [12] propose new mathematical approach generate a 3D path; the robot can reach and truck moving targets.

Our idea of navigation in known area is mainly inspired from the approach reported in [1], which upon on the spiking neural networks (SNN), the simulations of this method show that the robot finds the optimal path. To achieve our purpose, we develop an algorithm Fig.10 that combines the latter and the FL method. It ensures the use of methods according to the circumstances in order to construct an optimal path from the starting to the final configuration. The influence of our hybridization on the navigation of the mobile robot is proved using a set of simulations between our hybrid method and the fuzzy logic method. The simulation results include the paths of the robot's navigation and a set of curves representing the comparisons of the required parameters: distance and time of navigation. The simulations are carried out using the Python programming language.

This paper is organized as follows: The next section

discusses the navigation in unknown environment using Fuzzy Logic technique. The navigation in known environment is discussed in section 3. In section 4 we describe the hybridization procedure and the evaluation of our method using simulation. Section 5 concludes our work with a summary.

II. NAVIGATION USING FUZZY LOGIC (FL)

The proposed method of navigation in unknown environment is based on the fuzzy logic technique. It has composed of two behaviors (controllers): the first controller is used when there is no obstacle face to the robot; the second controller is launched when an obstacle is detected.

We bring the behaviors from existing methods and we make some changes to be suitable with our case. We solve the case of corners. The first controller is brought from [10]. The second controller is brought from [11]; in this controller, we change the membership functions of the input and output variables after many tests and simulations. We also give a static value to the velocity output variable for the both controllers. The steps followed by the robot are presented in Fig.1:

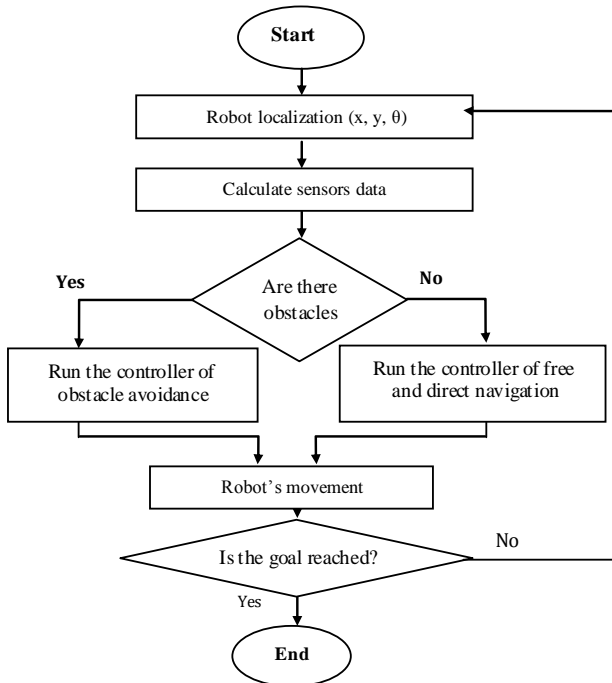


Fig. 1. The sequencing of navigation steps using FL

As mentioned previously, our fuzzy system is composed of two behaviors. Each one, works independently of the other, thus we need a mechanism that coordinate the both behaviors in order to have a good reaction with the environment. Several methods are proposed in literature to solve this problem: behavior arbitration [6-9], weighted [7-8] and the switching between behaviors method. For our system, we use the switching between behaviors, our choice is based on a comparison done between our system and another method [6] use the behaviors' arbitration method (two methods

are proposed in this work [6], we choose the best of them for the comparison), we built the same environment. As shown in the simulation results, our method gives approximately the same results Fig. 2.

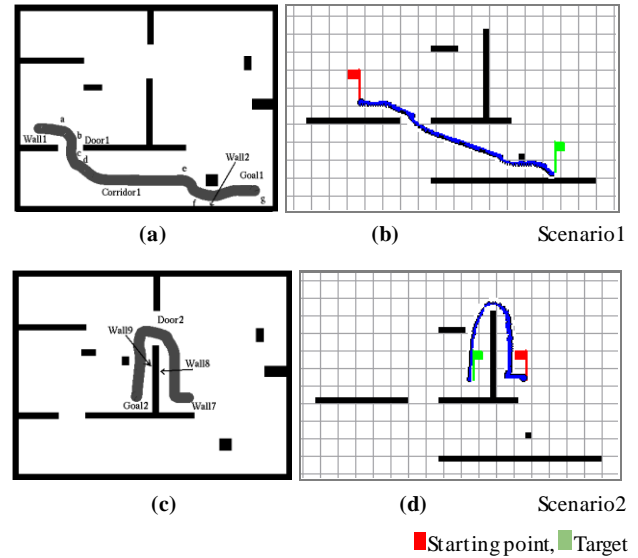


Fig. 2. Simulation results of Fuzzy Logic navigation method (b,d) Vs [6](a,c)

A. The implementation of the navigation controllers (behaviors)

The first behavior is executed when no obstacle is detected by the robot. When the robot finds an obstacle, it uses the second fuzzy controller to avoid the obstacle.

To implement these two fuzzy controllers, we need to specify input, output variables, the membership functions of these variables and the inference system which represented by rules base.

B. Input and output variables

The input variables of the free navigation controller are the distance and the angle between the robot and the goal. The output variable is the steering angle. Avoidance obstacle controller uses two input variables, which are the distance between the robot and the obstacle, as well as the angle between the current robot direction and the obstacle. The output variable of this controller is the steering angle.

According to received range information, reactive controllers of the fuzzy logic algorithm are used to control the steering angle. Initially we give a static value to the velocity.

C. Fuzzification and membership functions

According to [10], the linguistic variables Z(zero), VN(Very Near), N(Near), F(Far), VF(Very Far) are used to fuzzify the input variable distance and NB(negative big), NM(N medium), NS(N small), Z(zero), PB(positive big), PM(P medium), PS(P small) are used to fuzzify the angle between the robot and the target as well as the output variable steering angle.

The second controller, which is the obstacle avoidance behavior (OAB), uses the following fuzzy set: VN (very near), N (near), F (far), VF (very far) to fuzzify the distance input variable, which represents the distance

between the robot and the obstacle. As well as the angle between the robot and the obstacle which is represented by this fuzzy set: NB (negative big), NM (N medium), NS (N small), Z (zero), PB (positive big), PM (P medium), PS (P small).

The obstacle avoidance behavior produces an output variable which is the angle that the robot must turn with it (steering angle) in order to get the appropriate direction. The membership functions used to represent the input and output variable are represented in Fig.3.

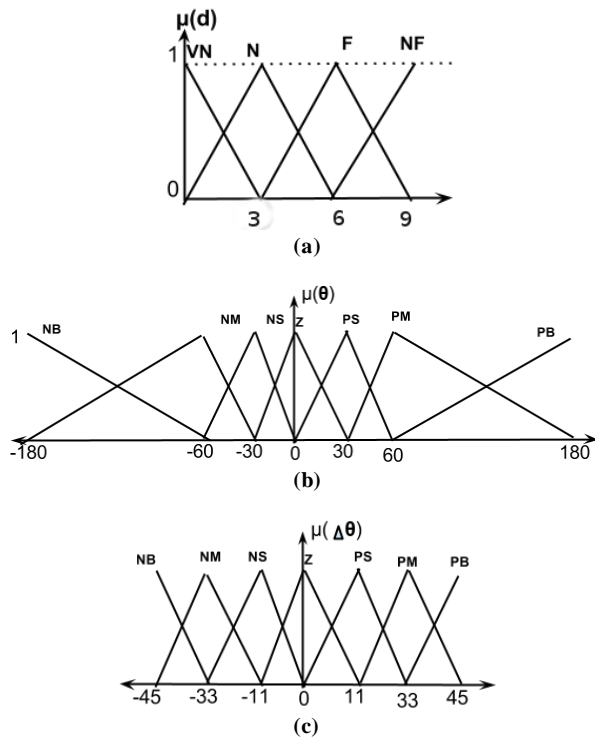


Fig. 3. The membership functions of input (a,b)/output (c) variables for obstacle avoidance behavior(OAB)

Fuzzy controllers are described using fuzzy set and inference system (fuzzy logic rules), fuzzy rules are integrated in two independent rule bases to explain how both controllers, free navigation and avoidance obstacles, are realized. We use the Mamdani (max-min) inference method for the both controllers and the centroid of area (COG) defuzzification method to defuzzify the output variables.

One of the navigation problems is how the robot reacts when there are corners Fig.4. In our solution when a corner is detected, the robot creates a virtual goal behind the wall encountered it, and when the robot passes the wall, it returns to the main target (this process is described in Fig.4).

In Fig.4 (a) the robot starts from (SP) to (Tp), when the robot detects that there is a corner at p1 point, it creates a virtual goal (vg) behind the wall, and thus the robot navigates to reach it. At (p2) it passes the wall and returns to the origin target. The robot forward to reach the target till (p3) it detects a new wall, the robot do the same reaction to avoid the second and the third wall then it is reached the target.

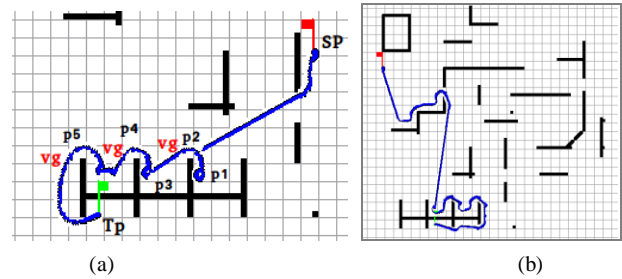


Fig. 4. Simulation show the solution of how the robot reacts when there is corners

D. Comparison with bug2 algorithm

Bug2 is one of the Bug's family, which represent a set of navigation algorithms those utilize the Boundary-Following and Motion-to-goal techniques to generate the path in unknown environment. If the robot uses these algorithms face to an unknown object, it can find a collision-free path by contouring this object.

1) The principle of bug2 algorithm

The main idea of bug2 algorithm is that the mobile robot moves toward the goal by following a constant slope computed initially between these two positions S(start point) and T(target point), unless encounters an obstacle. The robot goes around the obstacle until arrives to the initial slope again. [3]

2) Simulation results of Fuzzy Logic and Bug2

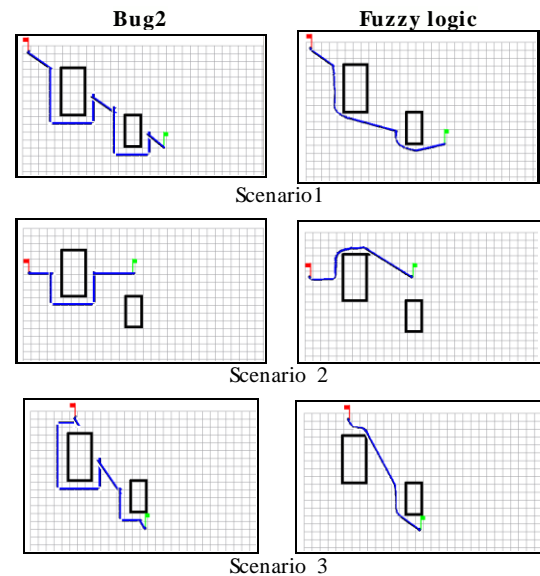


Fig. 5. Simulation results of FL Vs BUG2 navigation methods

3) Discussion

As shown above in the simulation results between the FL and Bug2 navigation methods, FL is better than Bug2, it finds an optimal path between the starting and target point. Consequently, FL has the less time of navigation.

III. NAVIGATION USING SPIKING NEURAL NETWORKS (SNN)

In order to find an optimal path to the goal in a known

environment we have improved a part of this method [1], it's based on the spiking neuron network. The general idea of the navigation is:

In the workspace, we have the starting point **S** and the target **T**. The target point **T** sends a wave that traverses the workspace, without moving through the obstacles. Each neuron receives this wave saves the first sender as it father and sends it to the directs neighbors through the workspace, the first point that sends the wave to the starting point **S** represents the father **F** of **S**, thus the robot moves toward the father **F** of each point, till reaches the

target point **T**.

A. How the wave spreads in the environments

At the first time the target point sends a wave to it direct neighbors Fig.6 (b), each neighbor sends this wave to it direct neighbors Fig. 6(c). The neuron that receives this wave and has not activated in the previous step will be active in this current step. The wave will spread in the environment using this mechanism till reaches the starting point.

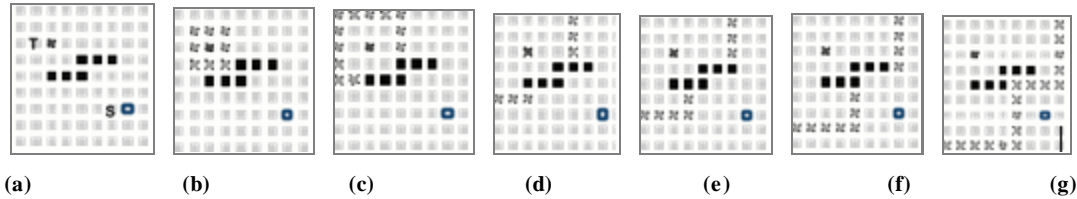


Fig. 6. The spread of the wave into a known environment

B. Neural Network Architecture

The workspace is discrete; it's represented as a grid such as a cell contains a robot. Every cell in the grid is represented by a neuron. Each neuron is connected with its eight direct neighbors (horizontal, vertical and diagonal); except the neurons situated at the corners and sides, Fig.7 (b) represent how a neuron (n_5) is connected with neighbors. The arrangement of neurons coincides with the discrete structure of the workspace. Each neuron communicates and runs its own computations only with its direct neighbors.

Neurons those represent the obstacles in the workspace haven't any connection with their direct neurons neighbors. We use the spiking neuron in order to implement the neural network.

A neuron is composed of Fig.7 (a):

- ▶ Multi connections: allow each neuron connecting with their direct neighbors.
- ▶ A connection between two neurons i and j has a weight W_{ij} , which has a static value. The horizontal (W_h) and vertical (W_v) links or connections have the same value (val_1), and have a different value (val_2) for the diagonal (W_d) connections, where $val_2 > val_1$.
- ▶ A function g add the weight W_{ij} to the value S_i of the sender neuron, the current neuron save the result of this function as S_i :

$$g_i = S_j + W_{ij}, S_i \leftarrow g_i$$

(This means that S_i get the value of the function g_i)

- ▶ j : represents the sender neuron
- ▶ S_i : represents the value of the function g for the sender neuron j
- ▶ W_{ij} : represents the weight between the current neuron i and the sender neuron j , this value differ according to the kind of link (horizontal, vertical or diagonal).

The workspace (**W**) is represented by a set of neurons; each one is defined by a couple:

$W = \{n_1, n_2 \dots n_k\}$, n_1 : neuron₁, k : the number of all neurons in the workspace. $n_i = (ID_i, st_i, NG_i)$, where:

ID_i: Identifier of the neuron i , **st_i**: represents the state of neuron i (Active or not active), this state is related to the own state of i and also the state of their neighbors.

NG_i = $\{(id_1, w_{i1}), (id_2, w_{i2}) \dots (id_j, w_{ij})\}$

NG: represents the neighbors of neuron i :

id_j: Identifier of the neuron neighbor j , j is the maximal value of a neighbor neuron of i .

W_{ij}: represents the weight of link that is wired the neuron i and their neighbor j .

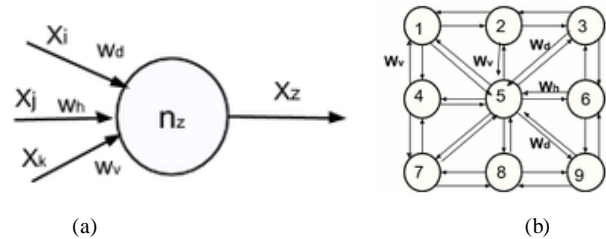


Fig. 7. Neural model (a), Connections of a neuron with the neighbors (b)

C. Path Planning Process

At the first time, not all neurons in the workspace (**W**) are active; the neuron corresponds to the target initializes and sends the wave in order to reach the starting neuron. The main functions of our method are presented in the following pseudo code, its represents how the wave is spreading in the workspace (**W**) and how the robot finds an optimal path to the goal:

1. **Proc initlz_network ()**: this function initializes the neurons network, the state of neurons and the weight of the connections between them.
2. **Proc trgt_neur_initialz_wave**: This function is private to the target neuron, which starts spreading wave.
3. To spread the wave in all network, each neuron receives the wave, executes the function **Proc neuron_receive_wave**.

4. When the wave arrives to the starting neuron, this neuron run the "**Proc neur_start_receive_wave**" function.

D. Pseudo-code of the main functions

```

Proc initlz_network():
  For all neurons:
    Statei = passive
    Si = -1
  For i in all neuron:
    For j in all neuron:
      If i and j are not obstacles then:
        If connectij is vertical or horizontal:
          Wij = val1
        Else(connectij is diagonal):
          Wij = val2
      End
    End
  End
End
note: val2 > val1

Proc neuron_receive_wave(Si):
  If neuri is the start_neuron then:
    neur_start_rcv_wave(Si)
  else:
    if neuri not activated at (t-1):
      g = Sj + Wij
      if g < Si then:
        Si = g
        Statei = active
        Fatheri = j
        Resend_wave_to_neighbors(Si)
        Statei = passive
      End
    End
  End
End

Proc trgt_neur_initialz_wave(i)
  Statei = active
  Fatheri = i
  Si = 0
  Send_wave_to_neighbors(Si)
  Statei = passive
End

Proc neur_start_receive_wave(Si):
  Fatheri = j
  While not reach_the_start_neur:
    Move_toward_fatheri()
  End
End

```

Fig. 8. Pseudo code of the main functions

E. Scenarios of robot's navigation using our method

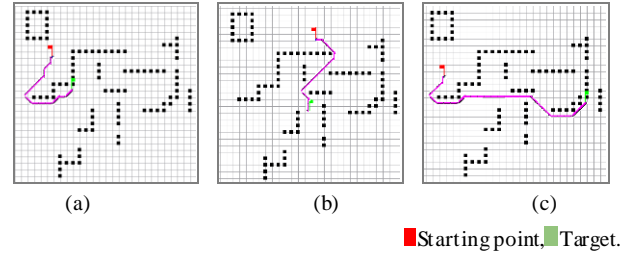


Fig. 9. Scenarios of robot navigation using our method

VI. HYBRIDIZATION PROCEDURE

The **FL** is used when the environment is unknown, it has also the same reaction when the environment is known, because it is based on the avoidance of obstacles encountered by the robot, thus it can't benefit of having known part in the environment.

Our contribution represents a hybrid method, it combines two navigation methods: fuzzy logic (FL) and the method based on spiking neuron network (SNN), and look to get the advantages of each one in order to ameliorate the global navigation when the environment is partially known. Our method has the following properties:

1. Less calculations in the processing algorithm
2. An optimal path between the starting and target point.
3. Improvement of the time of navigation.

A simulation done proves all this advantages. The results are represented in the next section.

A. How to build our hybrid method

The robot has two behaviors FL and SNN, the robot runs the behavior adequately with it localization, if it's situated in an unknown area of the environment it calls the FL method, else the robot runs the SNN method.

When the robot runs the FL method, it will re-execute the previous steps after each movement until reach the target. If the robot is located in a known environment, it runs the SNN method, there are two cases: the robot executes SNN method directly and finishes the navigation, when the robot runs the FL method; it will re-execute the previous steps after each movement until reach the target. If the robot locate in a known environment, it runs the SNN method, there is two sub cases: executes SNN method directly and finish the navigation, when the target is situated in this known part, the second sub case is when the target is out of this area, thus the robot creates a virtual target (VT) on the limits of this known area then apply the SNN method, the VT is the nearest point to the original target. When the robot reaches the virtual target it re-executes the previous steps until reaches the original target. The main algorithm is presented in Fig.10:

B. Simulation & discussion

To show the influence of the combination done between the FL and the method of SNN on the global navigation of the robot, we have built complex workspace which contains different cases those the robot may encounter, as well as the realization of several simulations.

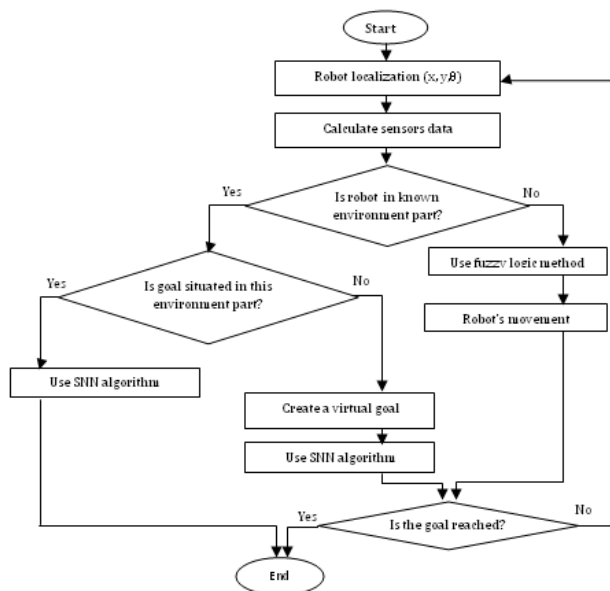
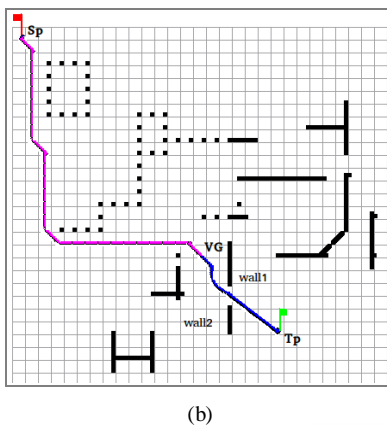
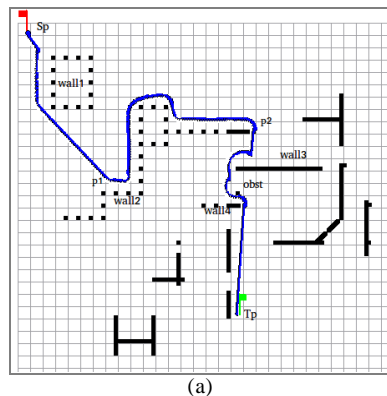


Fig. 10. The main algorithm that our hybrid method is based on.

The environment is partially known, thus the robot has a plan about some areas, which are represented in both figures Fig.11 and Fig.14 by discrete lines, and the unknown parts are represented by continuous lines. The path of navigation using our method is blue when the robot uses the FL and pink when it uses the method based on the SNN, this path highlights the robot's reactions with the state of environment (known, unknown) using each method.

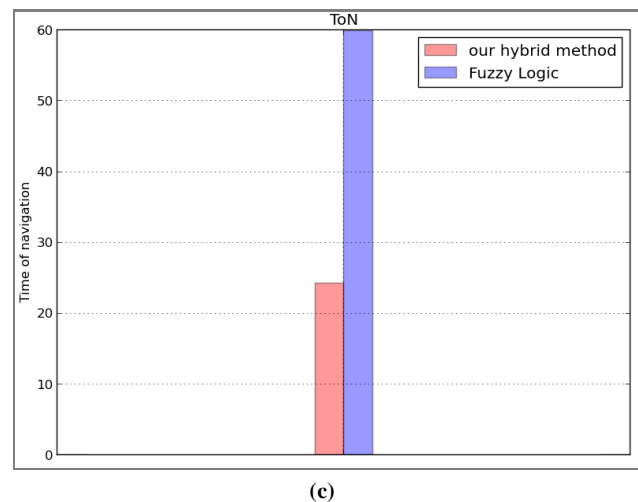
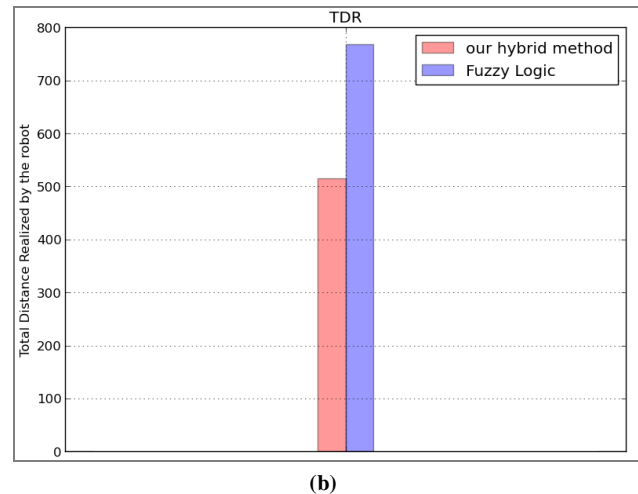
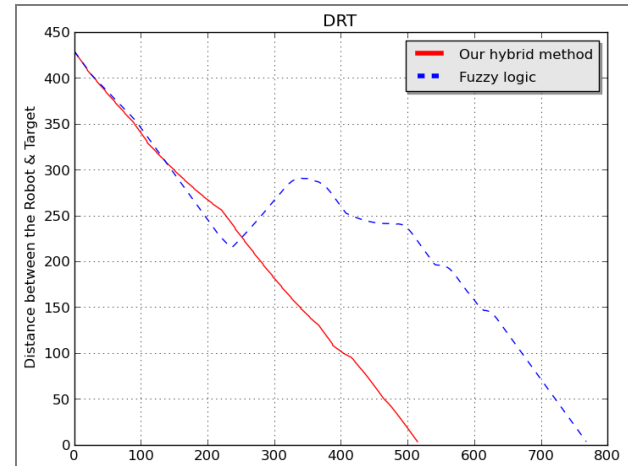


■ Starting point, ■ Target

Fig. 11. The paths of the robot navigation using FL (a), our hybrid method (b) (scenario 1)

The results of each kind of simulation doing for the both methods are organized as follows to facilitate the comparison: the path realized by each method, as well as the diagrams of the following parameters: **D**istance between the **R**obot and the **T**arget during navigation (**DRT**), the **T**otal **D**istance **R**ealized by the robot (**TDR**) and the **T**ime of **N**avigation (**ToN**).

1) Scenario 1

Fig. 12. Comparison between our hybrid method and Fuzzy logic of **DRT** (a), **TDR** (b) and **ToN** (c) parameters (scenario 1)

In this scenario, the robot moves from a known to unknown part in the environment. A detailed description of the both simulation is given as follow:

1. Fig. 11(a) shows the simulation of the FL method, the robot moves towards the goal till encountered an obstacle (wall₁); the robot uses the avoiding obstacle behavior to move beside it. Then no obstacle is detected, the robot moves on a straight line towards the goal. At this point (p1), the robot is front of an obstacle, the robot moves beside it till reaches this configuration (p2), at this point no obstacle is detected, thus the robot moves using the free navigation behavior till encounter the obstacle signed by (wall₃), followed by (obst) and (wall₄). Finally, the robot moves on a straight path till reaches the target (Tp).

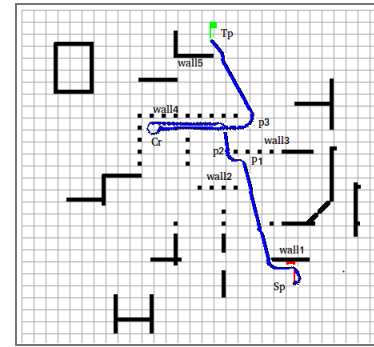
2. Fig.11(b) shows the simulation of our hybrid method, the starting point of the robot is in the known part, thus the robot use the method based on SNN to move (the path has a pink color). In the first step, the robot creates a virtual goal (VG) because the original goal isn't located in this known part, then the path to the virtual goal is built, and the robot follows it till reaches the virtual goal (the path is created from SP to VG). In the second step, when the robot gets the virtual goal, it moves in the unknown environment, thus it switches to fuzzy logic method automatically (the path has a blue color), in this time the robot is face to a wall (wall₁), the robot uses the avoiding obstacle behavior of the FL method to move beside it, till find a way to the goal and avoid the last obstacle (Wall₂) then the robot reaches the Target (Tp).

2) Scenario 2

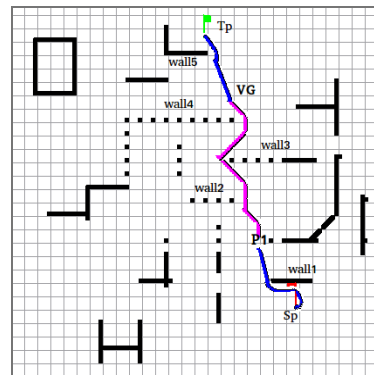
In the second scenario the starting and target points are located in two different unknown environment parts, and between them there is a known part. The description of the both simulations results is given as follows:

1. Fig.13 (a) shows the simulation of the FL method, the robot avoids the obstacle encountered (wall₁), and moves beside it using the avoiding obstacle behavior. In addition, because no obstacle is detected, the robot moves on a straight line towards the goal. At this point (p1), the robot is front of an obstacle, it moves beside this obstacle on the left side till end. On (p2) the robot moves using the free navigation behavior till encountered another obstacle (wall₄), the robot follows it, and when it's front the corner (Cr) it rotates and moves till reaches the point (p3). At the final step, the robot moves on a straight line, it reaches the target (Tp) after avoiding the (wall₅).

2. Fig.13 (b) shows the simulation of our hybrid method, the starting point of the robot is in the unknown part, the robot move using FL method. At the first time it's avoid the obstacle (wall₁) then move directly towards the goal. The robot moves in the known environment part at (P1), it switches to (SNN) the path represented by pink color, creates a virtual goal at (VG) and follows this path till reaches this virtual goal (VG). The robot moves in the unknown environment after reaching the virtual goal(VG), thus it switches to fuzzy logic navigation method (blue path), the robot moves on a straight line till encountered an obstacle (the corner of wall₅) near the goal, the robot reaches the target after avoiding (wall₅).



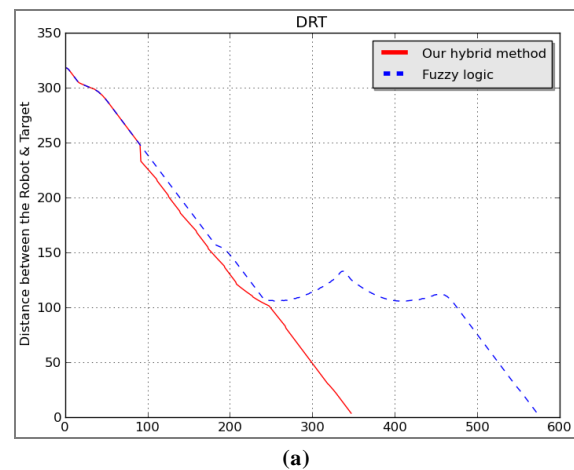
(a)



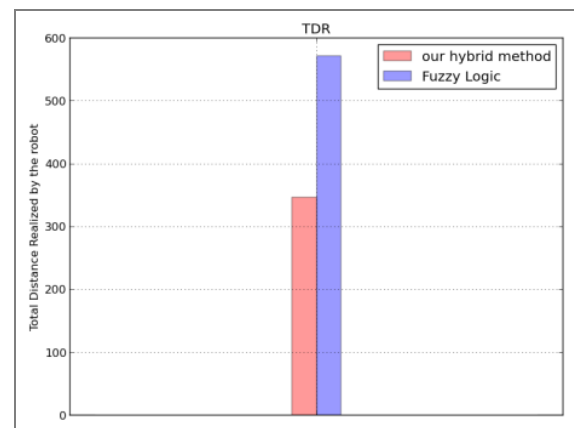
(b)

■ Starting point, ■ Target.

Fig. 13. The paths of the robot navigation using FL (a), our hybrid method (b)(scenario 2)



(a)



(b)

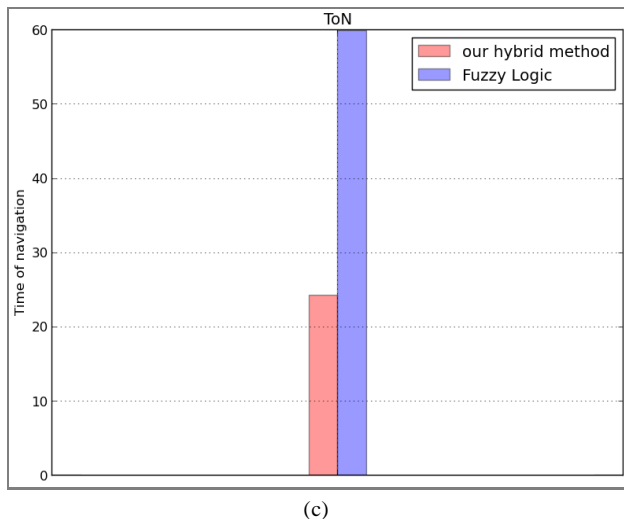


Fig. 14. Comparison between our hybrid method and Fuzzy logic for **DRT (a), TDR (b) and ToN (c)** parameters (scenario 2)

As shown in the above results Fig.11 and Fig.13, our method generates an optimal path better than the fuzzy logic method, as well as the diagrams of the both methods. When the robot uses our method, in order to reach the target, it's realized less distance Fig.12 (b), Fig.14 (b) and also less time of navigation Fig.12(c), Fig.14(c) compared to the fuzzy logic method. The Fig.12 (a), Fig.14(a) represent the distance between the robot and target during the navigation process, its show that our method allows to the robot to be nearer to the target than the fuzzy logic method.

V. CONCLUSION

In this work we have presented a hybrid method which is composed of two methods: the first is a new proposed method based on the spiking neuron networks, which is used when we have a known environment part, the second is a fuzzy logic method, it is composed of two behaviors, each one it implemented by the means of membership function and a fuzzy rules base.

The switching mechanism offered by fuzzy logic can effectively coordinate the two fuzzy behaviors; it is proved in the simulation results.

This architecture allows the robot to use the behavior that coincides with the state of the environment (the presence of obstacles or not, and also the environment is known or unknown).

In addition to the ability of avoiding obstacles, and reaching the target in totally unknown environment the simulation results prove that the robot can benefits of the known environments parts to improve the navigation (the fuzzy logic used when the environment is unknown): the robot has less time of navigation, as well as it save computation effort doing between the current and the destination configuration, and also the robot has an optimal path in the most cases.

We concentrate in this study on how to benefit in partially known environment to prepare to the next work that will study how to apply this method when the

environment is changed and also how to apply it for a set of robots.

REFERENCES

- [1] U.Roth, M.Walker, AHillmann, H.Klar, "Dynamic Path Planning with Spiking Neural Networks", Biological and Artificial Computation: From Neuroscience to Technology, Springer, Volume 1240, 1997, pp 1355-1363,1997.
- [2] Lebedev DV, Steil JJ, Ritter H. "A New Wave Neural Network Dynamics for Planning Safe Paths of Autonomous Objects in a Dynamically Changing World". In: Grmela A, Mastorakis NE, eds. *Advances in Neural Networks World*. WSEAS Press; 2002: 141–146.
- [3] Lumelsky, V.J. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment", IEEE Trans. Automat. Contr. 31, 1058–1063, 1986.
- [4] A. Yufka, O.Parlaktuna, "Performance comparison of bug algorithms for mobile robots, proceeding of the 5th International Advanced Technologies Symposium (IATS'09), Karabuk, Turkey, May 13-15, 2009.
- [5] Ch. Chiang, J.Liu, Y.Chou, "Comparing path length by boundary following fast matching method and bug algorithms for path planning", Springer-Verlag Berlin Heidelberg, opportunities and Challenges, SCI 214, pp. 303–309.2009
- [6] E. Aguirre, A. Gonzalez,"Fuzzy behaviors for mobile robot navigation: design, coordination and fusion", International Journal of Approximate Reasoning, pp. 255-289, 2000.
- [7] W.LI,"Fuzzy-Logic-Based Reactive Behavior Control of an Autonomous Mobile System in Unknown Environments". EngngAppl. Artif. Intell. Vol. 7, No. 5, pp. 521-531, 1994.
- [8] W.LI,"Fuzzy Logic Based Robot Navigation in Uncertain Environments by Multisensor Integration".Procs. Of the 1994 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MF1'94) Las Vega, NV Oct. 2-5, 1994.
- [9] A. Saffiotti "Fuzzy logic in autonomous robotics: behavior coordination".Procs. of the int. Conf. on Fuzzy Systems , pp. 573-578, Barcelona, SP,july 1997.
- [10] N. Ouadah, O.Azouaoui, M. Hamerlain "Implémentation d'un controleur flou pour la navigation d'un robot mobile de type voiture" *MajecSTIC*, pp. 237-244 .2005.
- [11] A. Hassam, B. Kasmi, M. Boubezoula "Planning of a Trajectory Along with Avoiding the Obstacle Based on Fuzzy Logic Method for a Unicycle Mobile Robot",The 13 IACIT'2012 Dec.10-13 ISSN:1812-0857
- [12] V. Aenugu,P. Woo" Mobile Robot Path Planning with Randomly Moving Obstacles and Goal ", International Journal of Intelligent Systems and Applications, Vol.4, No.2, March 2012 DOI: 10.5815/ijisa.2012.02.01
- [13] M.Faisal, R.Hedjar, M.Al Sulaiman, K.Al-Mutib" Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment", International Journal of Advanced Robotic Systems, 2013, Vol. 10, 37:2013
- [14] H. Surmann, J. Huser , L.Peters, " A Fuzzy System for Indoor Mobile Robot Navigation", Proc. of the Fourth IEEE Int. Conf. on Fuzzy Systems, pp. 83 - 86, 20 - 24.03.1995, Yokohama, Japan.
- [15] A. Fujimori, P.N.Nikiforuk, M.M. Gupta "Adaptive navigation of mobile robots with obstacle avoidance ", IEEE Transaction on robotics and automation, vol. 13, no. 4, august 1997.

Authors' Profiles

Zineb LAOUICI: Received her Master degree in Computer Science in 2012 at the University of Jijel - Algeria, on the thematic of AdHoc Networks. She is currently preparing the Ph.D. degree in Computer Science at the University of Oran - Algeria. She is also a research member at the Laboratory of Research in Industrial Computing and Networks at the University of Oran. Her main research interests include the control of Mobile Robots using the techniques of the Artificial Intelligence.

Mohammed Amine MAMI: Received the Engineer Degree in Soft Engineering in 1999 and the Master Degree in Soft Computing and Automatic Control in 2006 at the University of Oran - Algeria. He is currently preparing the Ph.D. degree in Soft Computing and Automatic Control at the same university. He is also a research member at the Laboratory of Research in Industrial Computing and Networks. His main research interests include Automatic Control, Industrial Computing, Mobile Robots and Networks.

Mohamed Fayçal KHELFI: Received Ph.D. degree in Automatic Control and Robotics from Nancy University, France, in 1995. He is currently Professor at the Computer Science Department - Faculty of Exact and Applied Sciences - University of Oran - Algeria. He is also a permanent research member at the Laboratory of Research in Industrial Computing and Networks. His main research interests include Automatic Control, Industrial Computing, Robotics and Networks.

How to cite this paper: Zineb LAOUICI, Mohammed Amine MAMI, Mohamed Fayçal KHELFI, "Hybrid Method for the Navigation of Mobile Robot Using Fuzzy Logic and Spiking Neural Networks", International Journal of Intelligent Systems and Applications(IJISA), vol.6, no.12, pp.1-9, 2014. DOI: 10.5815/ijisa.2014.12.01