Self-Organized Navigation in Multi-Robot System Based on the Behavioural Fuzzy Controller

TEGGAR Hamza[1]

Mustapha Stambouli, Mascara University, Route Mamounia 29000, Algeria
hamza.teggar@univ-mascara.dz

Abstract. Controlling a highly dynamic multi-robot system (DMRS) is a challenge scientific and technological in constant expansion. Our goal in this article is to control the DMRS and used the local interactions between robots to produce a form of advanced collective intelligence. For this, we have developed a reactive control architecture based on fuzzy behaviours using the concept of fuzzy logic. This architecture allows mobile robots to move in an unknown environment and, in the same time, to resolve the navigation conflicts. The simulation results obtained on the pioneer P3-DX robot clearly show the effectiveness of the proposed architecture.

Keywords: multi-robot systems · fuzzy logic · mobile robots · behavioural control architecture

1 Introduction

The main goal of using robots is to replace the man in the tiring, difficult and even dangerous activities. Nowadays, it is easier to create mobile robots that can navigate a real environment autonomously rather than controlling it remotely, which is ideal in field explorations. For example, robot vacuums works independently on all types of floors. By means of its sensors, it's able to avoid stairs and all kinds of obstacles. In addition, they can identify specific areas to return to as a base to recharge without assistance.

In order to increase the efficiency of robots in accomplishing more complex tasks, it became necessary to harness several robots to work together. This led to the development of effective control systems in collective robotics framework.

Collective robotic is synonymous with the existence of a group of robots. In DMRS, the assembly of all robotic entities must generate a coherent and an efficient configurations for the achievement of the desired tasks [1]. It doesn't only stipulate the setting up of an individual control for each robot, but also requires the use of appropriate control architecture.

The advantages of mobile robot cooperation are multiple, among them we can mention the reliability, flexibility, robustness and speed of controlled systems [1–3]. Consequently, controlling DMRS inevitably involves controlling the actions generated by each robot. In the literature of collective robotic, several approaches have been proposed to address this complex problem, as we can classify them as follows:

- **Centralized Architecture (CA).** In centralized architecture, a controller determines which robot to be assigned to perform a specific action. This requires that the controller evaluate on the both the performances of each robot, and its situation in the environment [4];
- **Distributed Architecture and Centralized Determination (DACD)**. These approaches are based on the individual reaction of the robot which is often. This reaction is locally calculated by robots and sent to the controller. When the controller receives all robot propositions action, it will determine the best action according to its situation [5];
- **Distributed Architecture (DA)**. The system becomes a distributed-MRS when there is an absence of the central control mechanism, in which, the decision-making to perform an action is usually the result of the interaction *'robots-robots'* or as a result of the interactions *'robots-environment'*. Most *'robots-robots'* interaction approaches assume that a reliable communication between robots is necessary for the selection action process. In *'robots-environment'* interaction approaches with which this paper is concerned, there is very little communication between [6].

This article focuses on navigation in DMRS. In this context, the navigation methods allow the robot to move from a start point to target point while avoiding obstacles. These methods must therefore take into account the following problems:

- Fixed Obstacle Avoidance Problem (FOAP);
- Mobile Obstacle Avoidance Problem (MOAP);
- Robots Collision Avoidance Problem (RCAP).

In the FOA Problem, the environment is considered static and contains a single robot. In the MOA Problem, the environment contains dynamic obstacles and several robots. In this case, each robot move in the environment without communication, and the robots are considered obstacles to each other. Finally, in the RCA Problem, robots could communicate with each other to find an arrangement to avoid collisions between them.

In this paper, the FOAP and RCAP will be considered. The main goal of this study is the design of a control architecture based on fuzzy behavioural rules, which exhibit both individual characteristics adapted to dynamic multi-robot systems and collective characteristics that promote a common goal. Nevertheless, the proposed architecture requires moving further away from the centralized conceptions of multi-robot system control towards fully distributed and reactive control systems. More specifically, we propose a protocol for the selection of behavioural actions that:

- Allows at the scale of robot to coordinate the activity of a set of elementary primitives (called behaviours) in a flexible way; and this to solve FOAP;

– Allows at the scale of the group of robots to achieve coordination between them promoting the common goal and this to solve RCAP.

The remainder of this paper is organized as follows. In the next section, an overview of multi-robot navigation systems is presented. In Section 3, the fuzzy controller architecture proposed for navigation will be presented and will be detailed in the Section 4. Then, in section 5, we present a navigation coordination process to solve the priority determination problem. The results of the simulation on the Pioneer 3pdx robot will be presented in Section 6. Finally, the Section 7 concludes the paper.

2 Related Works

Several architectures distributed are proposed to control the DMRS. In [7] the authors proposed a speed controller design using the combination of fuzzy logic and neural network. This system includes three neural controllers for local navigation, two neural networks for environmental recognition, and a fuzzy system for direction and velocity control. In this experiment, 24 ultrasonic range finders are used to collect the distance data and finally delivered them to the fuzzy logic system.

In [8] authors proposed an architecture controller to path planning and trajectory tracking. This approach need the generation of map and plans the suitable trajectory. Obstacle avoidance was done by the help of sonar data and modified Vector Field Histogram method.

An architecture with two reactive behaviours of the mobile robot (called: "go to goal obstacle avoidance behaviour" and "wall following behaviour") is used In [9] to solve the navigation problems. This technique allows an optimal motion planning in terms of path length and travelling time; it is meant to avoid collisions with convex and concave obstacles and to achieve the shortest path followed by the mobile robot.

In [10] authors proposed an object tracking algorithm with the help of low cost laser scanner and ultrasonic sensor. The proposed algorithm was successfully tested in follow-the-leader mode in indoor environment.

An architecture, called (NSBC) 'Null-Space based Behavioural control', has been proposed in [11]. This last, was implemented for the navigation in indoor environments and allows the collision avoidance for multi-robot distributed system. NSBC implements two navigation modes: the first is triggered when a fixed obstacle is detected. While the second, is being triggered by the robot R_1 when another moving robot R_2 is on its way. Using a laser sensor, an algorithm makes it possible to distinguish between the two types of obstacles (fixed obstacle or a robot), and it allows to trigger the appropriate mode. This architecture has been tested in various scenarios, and it has shown its effectiveness in preventing robots from colliding with each other.

In [12], the Mamdani and Sugeno Fuzzy Logic Controller (MSFLC) are developed for a mobile robot. The smoothness and efficiency that generated from these MSFLC is analyzed based on simulation of Pioneer P3-DX robot in virtual robotic software for single and multi-robot environments under static obstacles environment.

3 Robot Platform Architecture

The control architecture is strongly related to the hardware architecture of the mobile robot platform. The behaviour that we will be interested in this study is the avoidance of other mobile robots (RCAP) while trying to reach a target (motivated behaviour).

In this Article, identical robots are considered such that each of them has a set of sensors and actuators allowing it to move. The proposed controller architecture will be tested on the pioneer P3-DX robots. P3-DX robot have a circular platform and is equipped with a locomotion device consisting of two independent driving wheels, one caster-wheel, and 16 sonar sensors to cover a field perception of 380°. This platform is fairly standard and offers multiple application domains; also it is very easy to control.

Robots can communicate through messages. The proposed architecture consists of five modules and the interconnection between them is proposed in the figure Fig. 1.

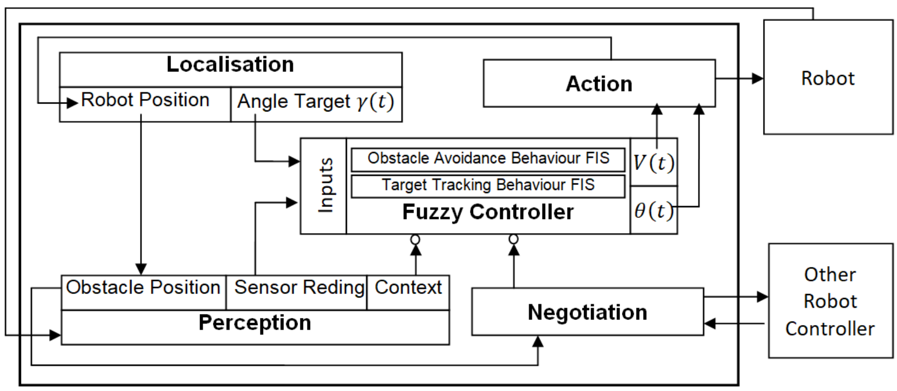
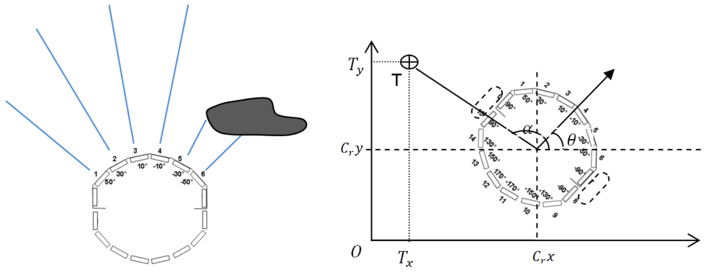


Fig. 1. Modular control architecture for multi-robot navigation.

3.1 Perception module

The perception module essentially ensures the detection of possible obstacles and carries out a local modelling of the environment to provide. This module format the information necessary for the decision on the movement to be made. For reasons of simplification, the controller uses only the six front sonar sensors



(a) (b)

Fig. 2. Platform Configuration:(a)-Orientation and range of the front sonar sensors; (b)-Relative robot and target positions.

identified from left to right (from 1 to 6). Each sensor is oriented with a specific angle according to the figure Fig. 2-(a).

The perception process requires two successive steps: the reading of the sensors, then the interpretation of this reading in raw information which makes it possible to deliver a clear data to the "Fuzzy controller". Indeed, the message sent by the perception module to the controller contains two pieces data:

- Sensor Reading (SR): The side on which the dynamic obstacle is located. In our case, this step does not perform any calculation because it is related to direct translation of the information transmitted by sensors;
- Obstacle Position (OP): An approximate position of the obstacle detected. This data is calculated from the opening angle of the sensor detecting the obstacle and the distance of this latter.

The first data (SR) is used for resolve Robots Collision Avoidance Problem (RCAP). The second (OP) is transmitted to the negotiation module to determine the detected obstacle state and the priority-of-way. This priority will be detailed later in the Section 5.

3.2 Localisation module

According to the chosen platform, the robots are considered as points in the configuration space with the absolute origin O in the environment. Each robot r is located by its centre point denoted by C_r. The different configurations of the robot in the environment E at any time are determined by the following kinematic equations:

$$\begin{cases} C_r x(t) = C_r x(t-1) + |V_r(t)| cos\theta(t) \\ C_r y(t) = C_r y(t-1) + |V_r(t)| sin\theta(t) \end{cases} \tag{1}$$

with

$$\begin{cases} \theta(t) = \theta(t-1) + \Delta\theta(t) \\ |V_r(t)| = |V_r(t-1)| + \Delta|V_r(t)| \end{cases} \tag{2}$$

such as $\theta(t)$ is the robot heading angle in the configuration space and $|V_r(t)|$ is the translation velocity of robot r at time t. The relative angle that the robot makes with the target T denoted by $\gamma(t)$ is presented in the figure Fig 2-(b).

$$\gamma(t) = \alpha(t) - \theta(t) \tag{3}$$

α is the angle of the target in configuration space such that :

$$\alpha(t) = arctan\left(\frac{T_y - C_r y}{T_x - C_r x}\right) \tag{4}$$

3.3 Negotiation Module

The negotiation between the different robots plays a fundamental role in the actions planning. The negotiation module allows robots to resolve conflicts that could put it at risk. For example, when two robots must determine the priority-of-way on the path intersection points, or in the case of leave the narrow passage as the corridors. To overcome this problem, the negotiation module ensures the coordination between the action modules of each robot through the establishment of contract. To avoid any type of collision, these contracts serve as rules for determining priority-of-way when two or more robots are at the same path intersection point.

4 Fuzzy Controller

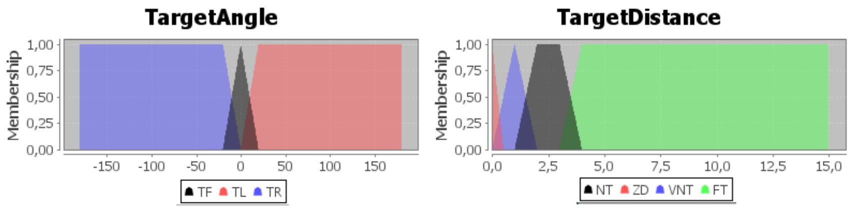
Behaviours are processing units that transform sensor data into information that can be used by the action module through a reactive process [4].

To implement behaviour, we use a series of fuzzy decision rules to resolve conflict situations. Because it is possible that two opposing behaviours, or more, can be triggered at the same time (ex: turn right to reach the target and turn left to avoid the obstacle). To solve this problem, we introduced the notion of context. This latter allows the determination of the behaviour should be followed by the robot based on a given environmental situation. All behaviours are not applicable in the same way. For example, the behaviour going towards the target will be more appropriate when the path is free.

In this paper, we propose describing the context through terms in fuzzy logic [13]. Specifically, we formulate the knowledge of robots about its environment as fuzzy variables with linguistic membership functions. Note that the robot has the following knowledge:

- $\gamma(t)$ represents the relative angle target. This data is delivered by the localisation module.
- Sonar sensors data allowing the detection of obstacles.

These variables can be used as inputs variables of the fuzzy controller to effectively estimate the value of the desired heading angle $\theta(t)$ of robot and the



(a) (b)

Fig. 3. Membership function of FIS input variables:(a) represents Membership function of Target angle variable, (b) represents Membership function of Target Distance variable

desired translation velocity $v(t)$. The value output variable $\theta(t)$ allows to robot the navigation in the environment with avoiding obstacles and in the same time the access to the target. The value of $\theta(t)$ and $v(t)$ is obtained by means of the Fuzzy Inference System (FIS) where we define two contexts: the first is Target Tracking and the second is Obstacle Avoidance.

4.1 Target Tracking Behaviour

To implement the target tracking behaviour, the first step, called fuzzification, consists of transforming the real values of the input variables into fuzzy variables. For this, the controller uses the following universe of discourse:

- T_A Target Angle: { TL: 'Target on the Left', TF: 'Target on front', TR: 'Target on the right' }
- T_D Target Distance: { ZD:' Zero Distance', VNT: ' Very Near Target' , 'NT: ' Near Target', FT: ' Far Target' }

The membership functions of these input variables are shown in figure Fig. 3.

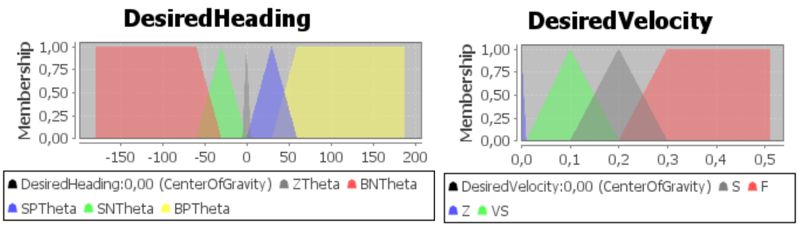
The two output variables are the desired heading and the desired translation velocity represented in figures Fig. 4.

- D_θ Desired θ : { $Z\theta$: ' Zero θ ', $BN\theta$: ' Big Negative θ ', $SN\theta$: ' Small Negative θ ', $BP\theta$: ' Big Positive θ ', $SP\theta$: ' Small Positive θ '}
- D_V Desired velocity :{ Z: ' Zero ', VS: ' Very Slow', S: ' Slow ', F: ' Fast '}

The proposed model uses fuzzy inference system proposed in [14]. The linguistic variable value of the desired θ and the velocity is determinate using 24 fuzzy rules proposed in table 1.

4.2 Obstacle Avoidance behaviour

Obstacle avoidance behaviour is triggered when one of the sensors detects an obstacle. The six sensors used by the robots have a range of 5 meters. The values transmitted by these sensors are normalized in the interval [0..1]. The



(a) (b)

Fig. 4. Membership function of FIS output variables:(a) represents Membership function of Desired θ, (b) represents Membership function of desired v

Table 1. List of fuzzy rules used for implementation of target tracking behaviour.

Target Distance T_D	Target Angle T_A		
	TL	TF	TR
Z	D_θ is $Z\theta$	D_θ is $Z\theta$	D_θ is $Z\theta$
	D_V is Z	D_V is Z	D_V is Z
VNT	D_θ is $SP\theta$	D_θ is $Z\theta$	D_θ is $SN\theta$
	D_V is VS	D_V is VS	D_V is VS
NT	D_θ is $BP\theta$	D_θ is $Z\theta$	D_θ is $BN\theta$
	D_V is S	D_V is S	D_V is S
FT	D_θ is $BP\theta$	D_θ is $Z\theta$	D_θ is $BN\theta$
	D_F is S	D_V is F	D_V is F

fuzzy input variables and their membership functions for each sensor are shown in figure Fig. 5. The outputs value of the desired θ and the velocity is determinate using 18 fuzzy rules proposed in table 2.

4.3 Defuzzification of outputs variables

When all the rules are put, we define the method of defuzzification. This step makes it possible to transform the linguistic variable values used by the fuzzy controller from fuzzy domain to the domain of the real values. The choice of the defuzzification method is usually conditioned by a compromise between the ease of implementation and performance of calculates. In this paper we used the centre of gravity method [12], which can be determined using the general relationship:

$$\theta^{*COG} = \frac{\int \theta \psi(\theta) d_\theta}{\int \psi(\theta) d_\theta} \tag{5}$$

$$v^{*COG} = \frac{\int v \psi(v) d_v}{\int \psi(v) d_v} \tag{6}$$

Table 2. List of fuzzy rules used for implementation of obstacle avoidance behaviour.

Sensor Range	Input Variable	Output Variables	
	Fuzzy Value	Desired θ	Desired Velocity
Sensor 1	VN	$BN\theta$	VS
	N	$SN\theta$	VS
	F	$Z\theta$	S
Sensor 2	VN	$BN\theta$	VS
	N	$N\theta$	S
	F	$SN\theta$	S
Sensor3	VN	$BN\theta$	Z
	N	$SN\theta$	VS
	F	$SN\theta$	F
Sensor 4	VN	$BP\theta$	Z
	N	$SP\theta$	VS
	F	$SP\theta$	F
Sensor 5	VN	$BP\theta$	VS
	N	$P\theta$	S
	F	$SP\theta$	S
Sensor 6	VN	$BP\theta$	VS
	N	$SP\theta$	VS
	F	$Z\theta$	S

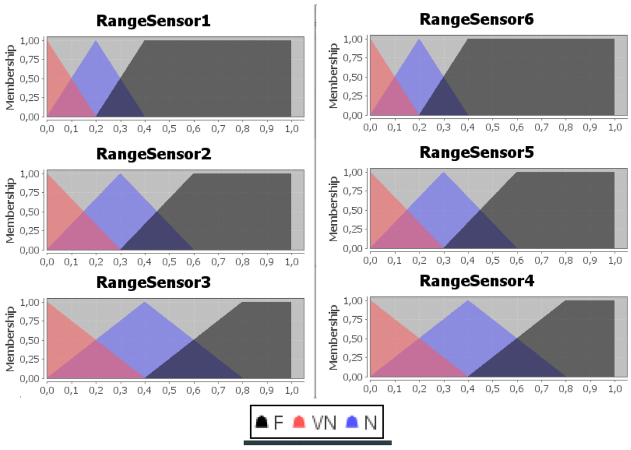


Fig. 5. Membership function of input variables Sonar Range

were θ^{*COG} is the real value of desired angle and v^{*COG} is the desired velocity obtained by Centre of Gravity method.

4.4 Behaviors Combination

The fuzzy controller architecture proposed in this article allows a robot to change its behaviour according to the following two contexts:

- if one of the sensors detects an obstacle then active the Obstacle Avoidance behaviour FIS;
- if no sensor has detected an obstacle then active the Target Tracking Behaviour FIS.

To activate the appropriate FIS, the behaviour combination process uses a function $Context(R)$ which returns 1 if an obstacle is detected, otherwise it returns 0; such that $R = \{r_1, r_2, .., r_6\}$ represent the values returned by the sensors s_1 respectively $(s_2, .., s_6)$.

$$Context(R) = \begin{cases} 1 \text{ if } (\min_{\forall r_i \in R} r_i) < RangeMax \\ 0 \text{ if } (\min_{\forall r_i \in R} r_i) = RangeMax \end{cases} \quad (7)$$

This function is implemented by the perception module, such that $RangeMax$ is the maximum range value of sensors fixed at 5 meters.

The proposed method is effective in the case of static environment without any dynamic obstacle.However, in our case, several robots can meet in the intersection of paths. Therefore, a mechanism for resolving conflict needs to be determined by computing the priority-of-way.

5 Coordination Process in the Navigation System

To overcome the priority-of-way determination problem (i.e when several robot paths intersect), the negotiation module coordinates the movement of the robots through the establishment of contract to resolve the conflict.

A robot that detects an object in its close environment will be considered as a manager robot. The manager broadcasts the detected coordinates of the obstacle point in the robot's network. Whichever robot receives this message, it must verify that the obstacle point (submitted by the manager) is not the coordinates of an obstacle, but it is indeed its own position point coordinates. In the event that this is true, the robot receiving the message of manager robot will become the contractor robot. That way, the manager and the contractor robots know that they are about to pass on a point of intersection. Once the manager and the contractor have been identified, they will enter into negotiations process to determine the priority-of-way.

To determine the priority-of-way, the multi-robot system uses the only shared information on the environment that is the position of the target denoted by T. Each manager and contractor calculates the relative position with respect to the target denoted respectively γ_{man} and γ_{cont}. We denote by $P(C_{man}, C_{cont})$ the priority function with C_{man} and C_{cont} are the centre point of the robot represented by the manager respectively the contractor robot. The function $P(C_{man}, C_{cont})$ is defined as follows:

$$P(C_{man}, C_{cont} = \begin{cases} 2 \text{ if } D_{man} < D_{cont} \\ 1 \text{ if } D_{man} = D_{cont} \; \& \; \gamma_{man} < \gamma_{cont} \\ 0 \text{ elsewere} \end{cases} \quad (8)$$

With D_{man} and D_{cont} are the distance represented by the manager respectively the contractor robots relative to the target T.

$$\begin{cases} D_{man} = \sqrt{(T_x - C_{man}x)^2 - T_y - C_{man}y)^2} \\ D_{cont} = \sqrt{(T_x - C_{cont}x)^2 - T_y - C_{cont}y)^2} \end{cases} \quad (9)$$

Once the priority function is computed locally by each robot, the robot with the highest value has the exclusive right of passage.

5.1 Communication Protocol

This protocol allows robots to exchange data, information and negotiation knowledge so they can navigate safely. In the SMR system studied in this paper, sending and receiving messages requires synchronous communication. For example, if the manager requests the submission of contractors, it remains on standby for a specified period defined in the time window $[BTS..ETS]$ (with BTS is the ' Beginning Time Submission ' and the ETS is the ' End Time Submission '). In the case where $t > ETS$, the manager considers the detected obstacle as a failed robot or a fixed obstacle.

With the proposed robot model, one of the main elements of communication is the transmission of a message that represents the position and orientation of the robot. When the number of robots increases, the communication protocol must be established in order to be able robot to address an individual or collective message. Each message sent by a robot is broken down into several fields; each field carries specific information for the receiver:

- **Sender** : The sender field contains the identifier of the robot sending the message;
- **Receiver**: The receiver field contains the receiver identifier of the message. In the case where the message is addressed to all robots, this field takes the word ALL.
- **Message ID**: Field contains the message identifier
- **Message Type**: This field tells the receiver what type and how it will be retrieve or process to the information.
- **DATA**: Field contains information, for example the coordinates of the obstacle point detected.

In the example of figure Fig 6, the $R01$ robot broadcasts a message to all robots. The type of a message is OD (Obstacle Detection) and the Data field contains the coordinates of the obstacle point in the absolute coordinate system (here 412 cm on the abscissa axis and -351 cm on the ordinate axis).

<table><tr><td>R01</td><td>ALL</td><td>@MAC-001</td><td>OD</td><td>412|-351</td></tr></table>

Fig. 6. Example of encapsulating a message

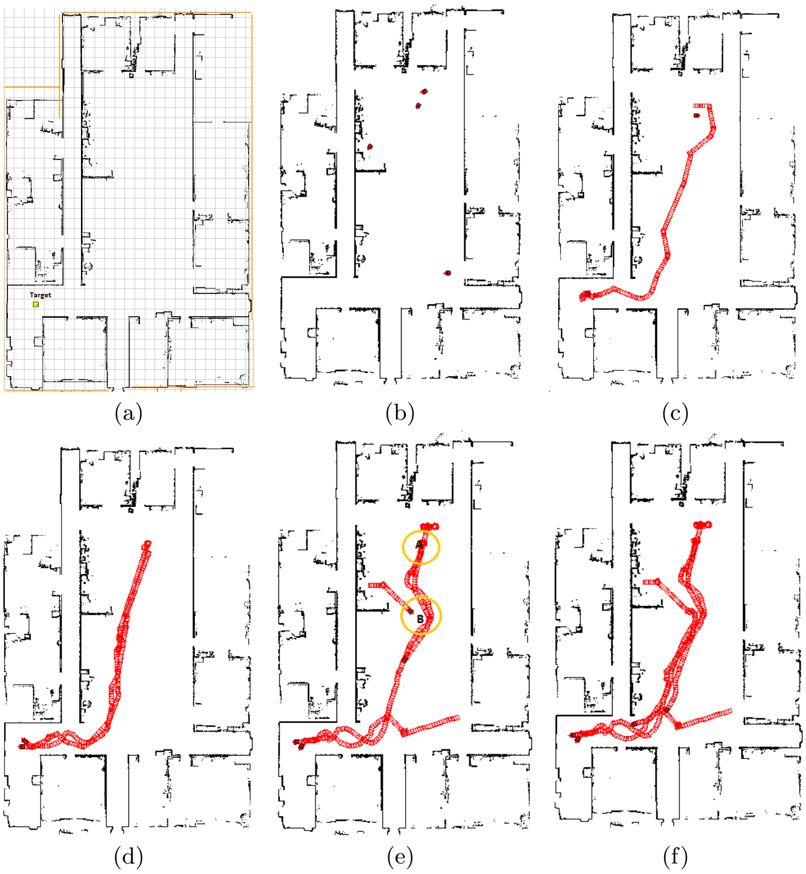


Fig. 7. Simulation results:(a)- Test environment topology; (b)-Start point of the robots; (c)-Navigation of a single robot with another failed; (d)-Navigation with two robots; (e)-Navigation scenario shows the role of priority-of-way at A and B intersection points; (f)-The paths generated at the end of the navigation.

6 Simulation result

In order to experimentally demonstrate the role of navigation controller, we have used a realistic simulation environment. Experimentation were performed using MobileSim Simulation Environment [15]. MobileSim simulates the physically-realistic mobile robots and their environments. We have used ARIA Java packages (Omron, which provided us with various tools to programming robot actions in high-level language and to developing a client application under Java for the description of complex tasks. Also, we have used Mapper3Basic to build realistic environment maps. The experiments were conducted by using a team of Pioneer P3DX robots. Each robot can localize itself based in odometric position estimation. The simulations were conducted in an enclosed space with dimension 40 x 25 meters included fixed obstacles.

The figure Fig. 7-(a) shows the environment that we have used to tested the proposed architecture and the figure Fig. 7-(b) shows the start points of 4 robots. We have realized several simulations with different configurations. In figure Fig. 7-(c) we have simulated a robot that has broken down, the traces of movement of the active robot shows that the latter considered the failed robot as a fixed obstacle. In figure Fig.. 7-(d) we have realized the same simulation, but we have turned on two robots simultaneously. The trace show that the two robots have not been intersects and the determination priority-of-way processes has not been made. In another test, figures Fig. 7-(e) and Fig. 7-(f) show that the robot paths cross in the two points A and B. The mechanism of determination priority-of-way makes it possible to avoid the collisions. Finally, the figure Fig. 7-(f) clearly shows that robots always arrive at the target without collision with any obstacles or with another robot.

7 Conclusion

Fuzzy logic can be a useful tool for the navigation process in the distributed MRS and more particularly in a dynamic and uncertain environment. It can be used to formulate the different information received and observed by robots in the form of linguistic terms similar to those used by human. From this point of view, we can design robots with reasoning abilities which are similar to those found in humans. This may be possible if we choose a set of rules that allow the use these linguistic variables to conclude other information in order to make a particular decision. For this, it is necessary to use an appropriate architecture, especially when the functionalities of the robots are effectively separated. The modular architecture model proposed in this article shows a great flexibility to implement the different navigation behaviours in multi-robot mobile systems.

References

1. Martinoli, A.: Modeling Swarm Robotic Systems: a Case Study in Collaborative Distributed Manipulation. The International Journal of Robotics Research **23**(5), 415–436 (2004)
2. Buffet, O., Dutech, A., Charpillet, F: Adaptive combination of behaviors in an agent. In: 9th European Conference on Artificial Intelligence, pp. 48–52. (2010)
3. Jones, C., Mataric, M.: Sequential task execution in a minimalist distributed robotic system. In: From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior, pp. 395. MIT Press, (2002)

4. Liu, C., Kroll, A. : A centralized multi-robot task allocation for industrial plant inspection by using a* and genetic algorithms. In: International Conference on Artificial Intelligence and Soft Computing, pp. 466–474. Springer, Berlin, Heidelberg (2012)
5. Liu, F., Liang, S., Xian, X: Multi-robot task allocation based on utility and distributed computing and centralized determination. In: The 27th Chinese Control and Decision Conference (2015 CCDC), pp. 3259–3264. IEEE, (2015)
6. Parker, L. E.: ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In: Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS'94), pp. 776–783, IEEE, (1994)
7. Chi, K. H., Lee, M. F. R.: Obstacle avoidance in mobile robot using neural network. In: International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 5082–5085. IEEE, (2011)
8. Hoang, T. T., Hiep, D. T., Duong, P. M., Van, N. T. T., Duong, B. G., Vinh, T. Q.: Proposal of algorithms for navigation and obstacles avoidance of autonomous mobile robot. In: IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), pp. 1308–1313, IEEE, (2013)
9. Aouf, A., Boussaid, L., Sakly, A.: Same fuzzy logic controller for two-wheeled mobile robot navigation in strange environments. Journal of Robotics **2019**, (2019)
10. Cheam, S. R., Saman, A. B. S. : Developing algorithm for object tracking using passive sensors. In:5th International Conference on Intelligent and Advanced Systems (ICIAS), pp. 1–5, IEEE, (2014)
11. Arrichiello, F., Chiaverini, S., Mehta, V. K.: Experiments of obstacles and collision avoidance with a distributed multi-robot system. In:IEEE International Conference on Information and Automation, pp. 727–732, IEEE, (2012)
12. Apandi, N. I. A., Martin, A. : The Integration of Fuzzy Logic System for Obstacle Avoidance Behavior of Mobile Robot. International Journal of Electrical Engineering and Applied Sciences (IJEEAS) **2**(1), 31–38 (2019)
13. Zadeh, L. A.: Fuzzy sets. Information and control **8**(3), 338–353 (1965)
14. Mamdani, E. H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International journal of man-machine studies **7**(1), 1–13 (1975)
15. Omron ('MobileSim'), http://robots.mobilerobots.com/wiki/MobileSim.. Last accessed 17 Jan 2019