ISSN 2277-3061

Using Fuzzy Logic in Robot Navigation to Find Animals

Mahdi Amiri, Zeinab Abbasi, Fakhte Soltani Tafreshi
Engineering Faculty, Arak University, Arak, Iran
m-amiri@phd.araku.ac.ir
z-abbasi@phd.araku.ac.ir
f-soltani@araku.ac.ir

ABSTRACT

Fuzzy logic is a tool to use human expertise. The simplicity of fuzzy-rule based systems and its power to perform various tasks without accurate measurement and computation makes it very popular between sciences. One of these applications is using fuzzy logic in designing the controller for navigation of autonomous robots to move in various environments. This paper proposes a new method of robot navigation based on fuzzy logic. This method can be drawn upon to design robots which can find and catch different kind of animals, especially endangered species. It works based on a hierarchical set of behavior each of which acts by using a set of fuzzy rules. The proposed method is simulated and tested by MATLAB software.

Keywords

Fuzzy logic; Fuzzy algorithms; Robot navigation;Tracking animals; hound robot.

SUBJECT CLASSIFICATION

Fuzzy logic algorithms

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol. 12, No.4

editor@cirworld.com

www.ijctonline.com, www.cirworld.com

ISSN 2277-3061

INTRODUCTION

The “Fuzzy Logic” is a vague term in machine control. The word “fuzzy” in oxford dictionary means inexact, unspecific, cloudy, hazy, blurry, and confused. In fuzzy logic, linguistic variables are applied to display acting parameters with the purpose of using human-like thinking. Fuzzy logic uses a set of simple if-then rules for solving the problem instead of trying to model a system in mathematic. It has a fundamental rule in majority of the uses of fuzzy logic. This important feature of fuzzy logic is very applicable for systematic design of many systems to get the better performance, when analyzing the data with conventional mathematical model is impossible [1]. One of the applications of fuzzy logic is the robot navigation [4].

The navigation of mobile robot in unknown environments causes two problems: localization and path planning. The localization is the process of identifying position and direction of the robot with respect to its surroundings. Path planning is finding a path without collision from start to end [7]. This paper tries to investigate a new application of autonomous mobile robots. These types of robots can be used to find animals that live in various environments. One of the applications of this research is making the hound robot. However, one more important application of this robot is finding endangered species of animals.

Nowadays two common methods used to find these kinds of animals are situating cameras in various places and/or employing people who are experts in tracking animals by their trace. Both of these approaches suffer from several problems; cameras can trace animals only in angle of view and they can’t catch/follow them. As for the second method, it isn’t economical and safe. This paper is an attempt to propose a method which doesn’t suffer from the above mentioned weaknesses. However, there is a limitation on the type of environment in which the robots can be used; they can’t trace animals in the air or in the sea.

The plan of the paper is as follows: section 2 reviews the related papers. Section 3 expresses a brief discussion about concepts and principles of fuzzy logic. Sections 4, describes our new method of robot navigation. Section 5 presents simulation of this method with MATLAB software and the result of the study. Finally, section 6 concludes this paper.

RELATED WORKS

Generally, all previous works are based on one of the following approaches: the first approach is based on computational geometry and the second one draws upon heuristics method. The former approach has a richer background. The second approach contains soft computing method (Fuzzy logic, Genetic algorithm and neural network). In the middle of 90s, a new approach was introduced which combined computational geometry and probability. The purpose of proposing this approach was to decrease the complexities in configuration space and search in the methods of the first approach [4].

Many methods are presented based on these approaches. However, they can’t solve the problem in general. For example, some methods limit the work space of robot to two dimension and the obstacles to polygons. Although there are differences among these methods, they share three bases; road map, cell decomposition, and potential field approaches.

The exact cell decomposition and road map can’t solve the problem in complex environment due to many of computations. Potential field method is frequently used for local navigations because of its simplicity. But it has stopping problem in local minimum. In the last decades using soft computing methods becomes noticeable. These methods, that contain fuzzy logic, neural networks and genetic algorithm, try to decrease the complexity of path computation, modeling the obstacles and learning. So, faced with dynamic environments in which the obstacles move and the robot must find the alternative paths in a short time without collision with the obstacles, soft computing must be used [4][7]. For example [8], which is one of the first methods based on soft computing, used a combined method of genetic algorithm and fuzzy logic. In this method, it is supposed that the target is stable but obstacles are dynamic and movable. This method uses genetic algorithm for calibrating state space variables factors, and uses fuzzy logic controller rules set for navigating the robot between moving obstacles.

In [11] fuzzy time reasoning is used to estimate the moving path of mobile things and avoid collision with these things. Reference [13] uses fuzzy logic to accurately estimate the position of the robot. First it uses multiple sensor sets to compute several approximated positions and then it applies fuzzy logic to find precise robot location with the help of approximated positions. Reference [14] investigates several problems about robot navigation, such as hole detection, choosing a rule to execute when multiple obstacles with equal distances are exist around the robot. It proposes a method using fuzzy logic that reduces robot navigation time. Also in [15], is presented a way to model a multi-agent simulation of autonomous robots that can response to behaviors of dynamic and nondeterministic environments and predicate the behaviors of the environment to make the better decision.

The method used in [4], is composed of fuzzy algorithms and reinforcement learning, for robot navigation in the semi structured building (an environment has some unknown in size and location obstacles and target and obstacles are moving). Ref [5] presents a set of fuzzy logic controller based on various behavior of the robot faced with the obstacles or the target. In this method, the robot firstly measures the distance to the obstacles or the target and then it turns or recedes from obstacles and move toward the target based on the measured distance. Also, in [5] the robot uses 6 sensors to measure the distance to the obstacles. Considering these measurements and a set of fuzzy rules, it then sends the right command to robot wheels to identifying the orientation and the speed. This method is used in indoor environments. Figure. 1 gives a brief presentation of various navigation methods.

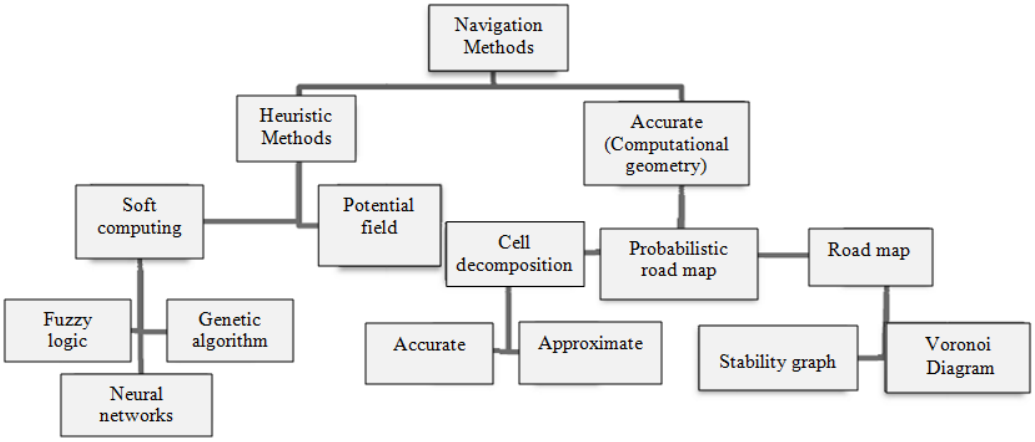


Figure 1. The various methods of robot navigation[4]

Fuzzy Logic

Fuzzy Logic is a tool that uses the expertise of the human brain. Fuzzy control systems are rule-based systems or knowledge based systems, that are a set if - then fuzzy rules are based on domain knowledge or human expertise. Simplicity of fuzzy rule-based systems and the ability to perform a variety of tasks without any explicit measurements and calculations makes it widely common among science and scientists [3].

In fuzzy sets an object can be partly owned by a set which is called grade of membership and is calculated by membership function. With this set of partial membership, controllers can be made such that manage inaccurate noisy inputs and produce an acceptable feedback. Fuzzy logic is an interface between the real world and the PC world. The system can take a decision on the membership functions called fuzzy inference system (FIS) [2].

"Fuzzy" object has the ability of transition from a state to another state by changing levels. Outline of a fuzzy controller is shown in Figure 2. Fuzzy logic controller design process includes the following steps: 1) Initialization, 2) Fuzzification and 3) Difuzzification.

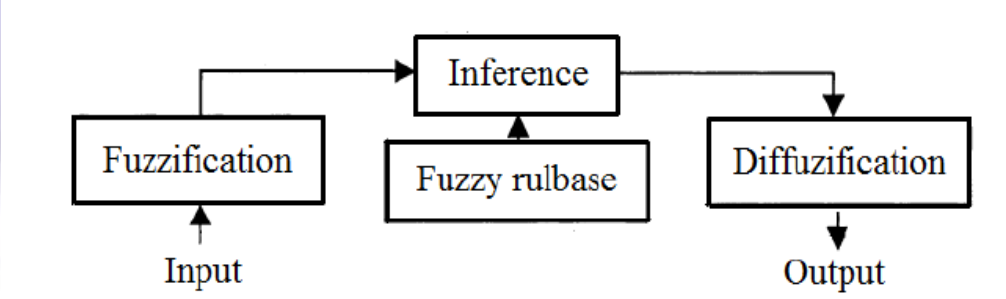


Figure 2. The fuzzy controller structure [3]

The first step in determining the linguistic input and output variables is to define fuzzy sets. Fuzzification or fuzzy classification is the process of converting a set of accurate and crisp data into a set of fuzzy variables using membership functions (fuzzy sets) [3]. For example, in Figure 3, the grade of membership to a given data is 0.6 of membership function depending on the input data can be a triangle, a piece of linear, Gaussian, trapezoidal or unique.

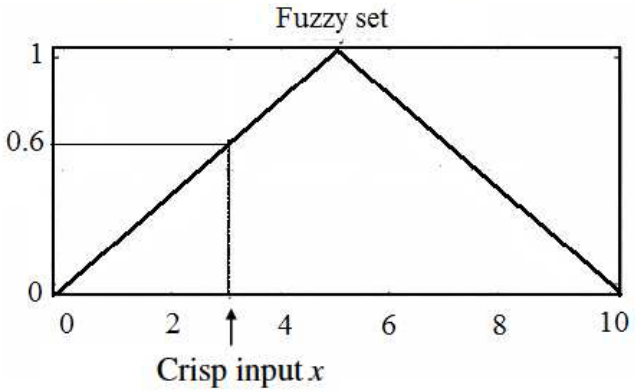


Figure 3. Membership degree of a crisp input x in the fuzzy set [3]

Fuzzy rules are set of if-then rules that are necessary for controllers and stores the rules dominating the proposed controlling input and output. The inference mechanism is responsible for deciding in fuzzy controller which uses approximate reasoning. The procedure of Difuzzification maps fuzzy output of inference mechanism into a clear signal [6].

Robot Navigation Method

So far, many methods have been proposed based on fuzzy logic for routing robots based on the kind of environment that the robot is in operation; for example, methods provided in [5] and [6]. This section describes the methods used in this research for robot navigation. The proposed method is based on the algorithm described in [5] along with the changes and modifications in the obstacle avoidance behavior and path reminder behavior design. The structure of robot in [6] is used for routing. The robot used is as shown in Figure 4.

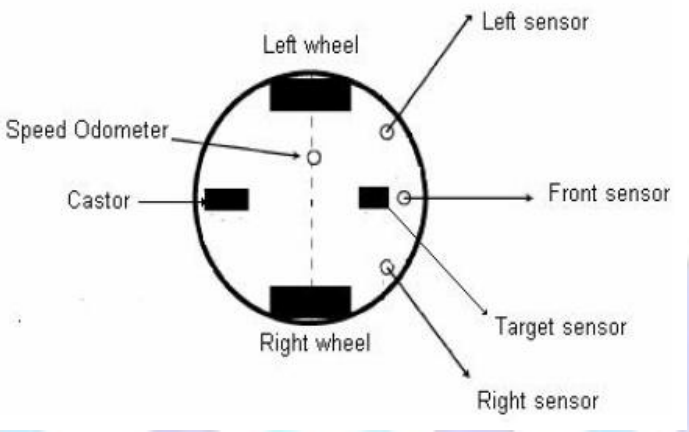


Figure 4. The structure of robot [6]

Fuzzy controller inputs are outputs obtained from the sensors. distance from obstacles d_l,d_r,d_f is obtained from front , left and right sensors , target sensor that actually could be a camera with the capability of detecting the regarding animal with fuzzy methods used in [2] or other methods gives us the target coordinates (d_x, d_y) using the distance measured by the target sensor and the current position of the robot. The method used in [2] is such that firstly, taken image is processed and changed into black and white, then divided into sections, later on image is analyzed, and then using fuzzy logic the object movement is to be detected.

Building behavior-based fuzzy controllers for robot navigation is depicted in Figure 5. Behavior controller is designed based on the distance from obstacles dl,dr,df and target coordinates (dx, dy) and the steering angle of the robot(φ). Meanwhile, behavior controller decides on the behavior with higher priority, and moving robot is controlled by higher priority behavior during motion to the destination.

ISSN 2277-3061

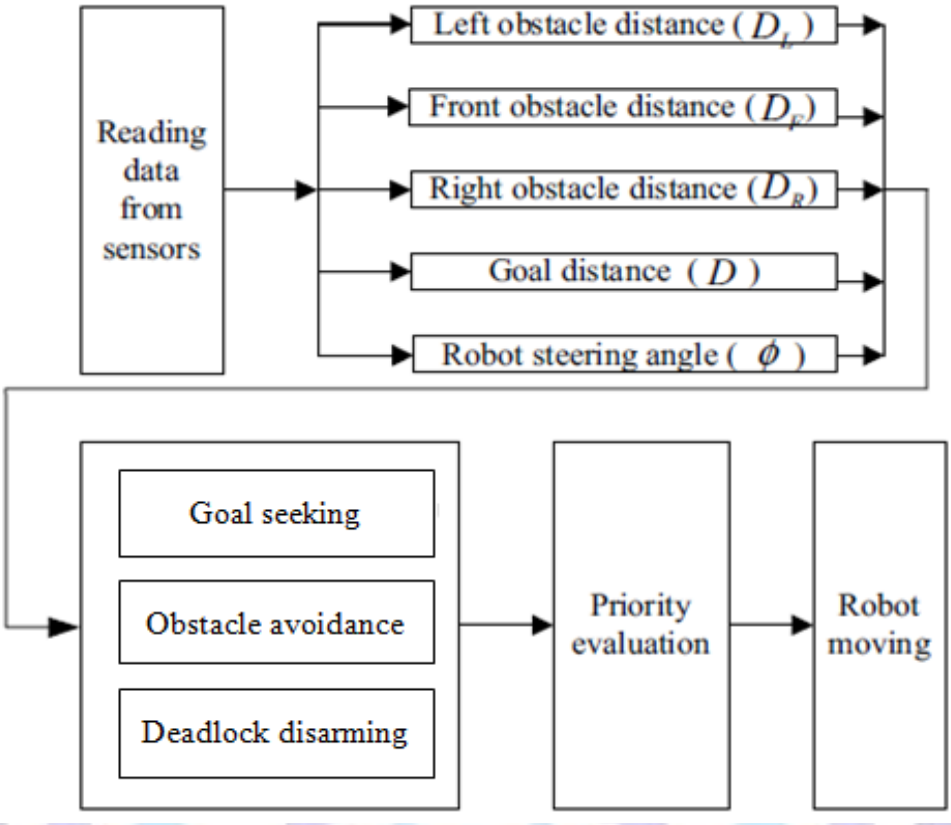


Figure 5. Behavior-based hierarchical fuzzy controller architecture based on [5]

DESIGN OF ROBOT BEHAVIOR

The Goal Seeking Behavior

When there is no obstacle around a moving robot, Goal seeking behavior causes the robot to move towards the target. For this behavior, robot receives the coordinates of the target and the robot steering angle φ as fuzzy input controller values of Goal seeking. As previously mentioned, the target coordinates d_x, d_y are obtained based on the distance measured to target by robot's target sensor and the current coordinates of the robot. The robot steering angle is obtained from relation "m= tan φ" where m is the slope of the line drawn between the current coordinates of the robot (d_{xr}, d_{yr}) and target coordinates d_x, d_y .

Rotation angle of the robot is also output values of WO controller and linear velocity, VO. The input value membership function D (distance from obstacle) φ, dx, and dy is shown in Figure 6(a), 6(b), 6(i) and 6(h). In this figures, the abbreviated terms of membership functions are: VL: very left, LE: left, MI: middle, RI: right, VR: very right and VB: very bottom, BU: bottom, MI: middle, TO: Top, VT: very high NB: negative big, NM: negative medium, NS: negative small, ZE: zero, PS: positive small, PM: positive medium, BM: big middle. (positive and negative means that target is to the left or right of the robot) and membership functions of output values of WO and VO are shown in Figure 6(e) and 6(f), respectively, with abbreviated terms meanings of LB: big Left , LM: left middle, LS: left small , ZE : zero , RS: right small , RM: right middle, RB: right big, (here, the left and right means the robot turning left and right , respectively). It should be noted that the definitions of membership functions and their value giving are experimentally. Goal seeking fuzzy rules are summarized as follows:

A) If the target is on the left (or the right) to the robot, the robot turns to the left (or the right), and if the target is on the top (or the bottom) to the robot, the robot moves upward (or downward).

B) If the robot steering angle φ, is large (medium, small), the rotation angle of the robot WO will be large (medium, small).

C) If the distance between the current location and the target is close (middle, far), the linear velocity, VO, will be low (medium, fast) [5].

The Obstacle Avoidance Behavior

This behavior is activated when the obstacles are in front, right - front and left – front to the robot or the obstacles are on one side or both sides of the robot (the side obstacles such as a wall), so this behavior receives distance front obstacles, DF, and the side obstacles, DS, as the controlling fuzzy input values of distance obstacle. Output values are, Rotation angle control, WO, and the linear velocity, VO.

Input membership application of DF and DS are depicted in Figure 6(c) and 6(d). (RL: Right Large, RS: Right Small, ZE: zero, LS: Left Small, LI: Left Large), when the distance to the left obstacle (right) is greater than 1 meter, distance is converted to side obstacle: DS =-DR (DS = DL)

Distance from left (DL) and distance from right (DR) show controlling fuzzy input values for side obstacle avoidance behavior. It needs to be noted that there are various way to avoid the side obstacles or to follow the wall. For example, in [12] a fuzzy controller using genetic algorithms for implementing a wall following behavior in a mobile robot is presented. This algorithm is based on iterative learning rule approach.

Membership functions application of output values of WO and VO in stay away from the obstacle are shown in figures 6(e) and 6(f), respectively.

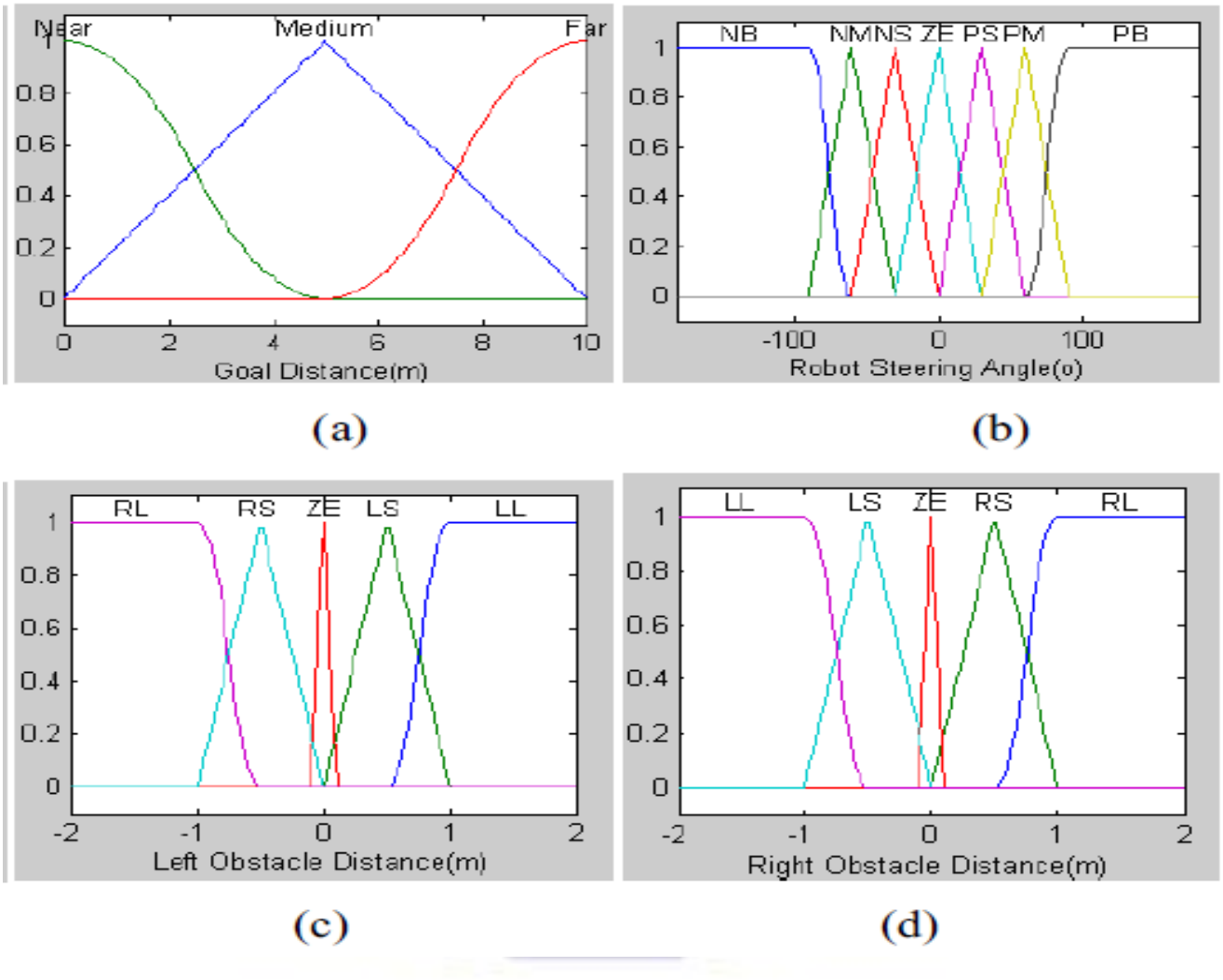
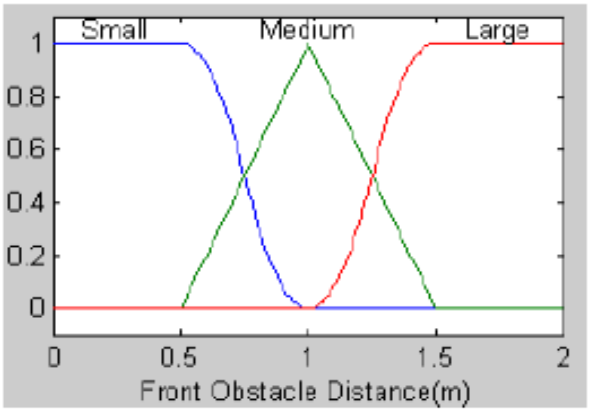
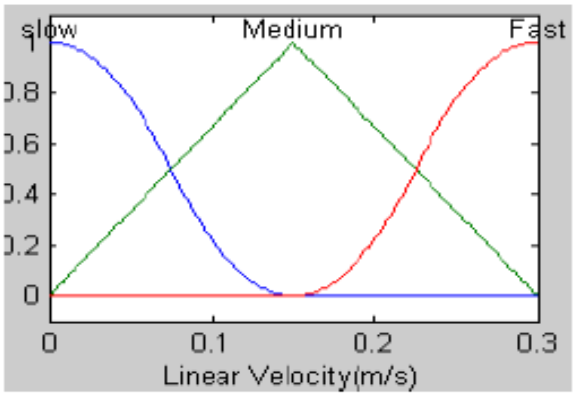


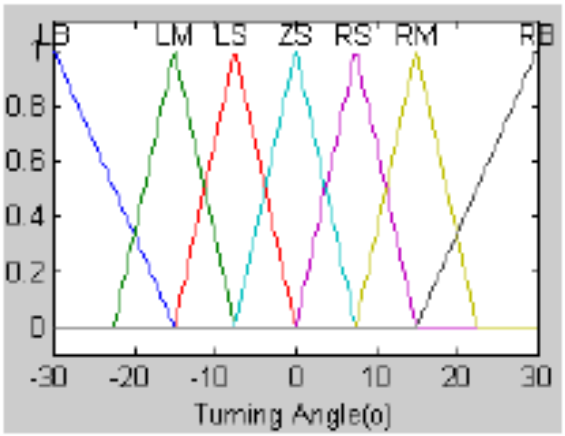
Figure 6. The defined membership functions for solving the problem based on [5]



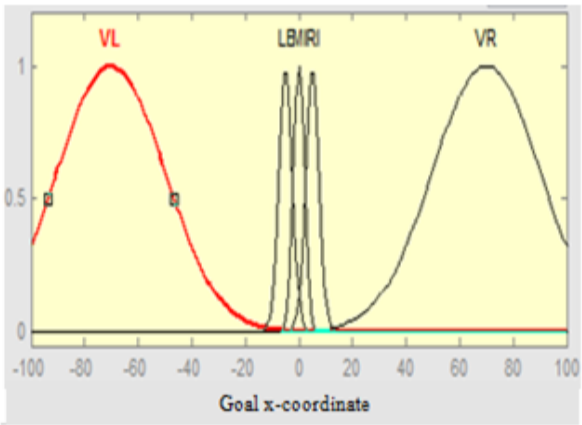
(e)



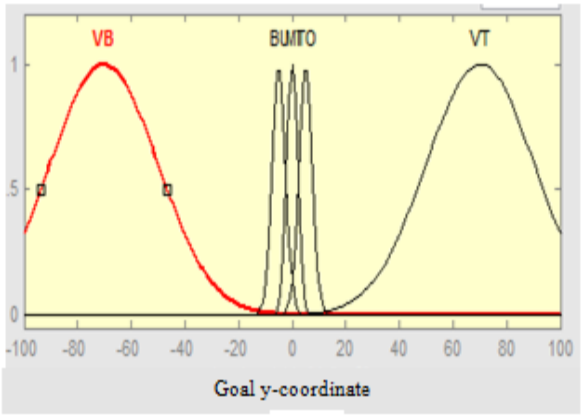
(f)



(g)



(i)



(h)

Continue of Figure 6

The obstacle avoidance fuzzy rules can be summarized as follows:

A) If the obstacle is to the left (right) of robot, the robot rotates to the right (left).

B) If the front obstacle distance is small (medium, large), the rotation angle will be large (medium, small) and linear velocity will be low (medium, fast).

C) If the obstacle is to the right (left) side of robot and the obstacle type is the side obstacle and target is to the left, the robot moves forward toward the target.

D) If the obstacle is on the left (right) side of robot and the obstacle type is the side obstacle and target is to the right, the robot moves toward the target.

F) If the robot moves towards the target, the linear velocity speeds up, otherwise linear velocity is average.

The deadlock disarming behavior

In some cases when the robot moves in V and U-shaped obstacles, it is very difficult for the robot to get rid of such situations. For the robot to get rid of such a predicament, the robot's path reminder behavior is activated to help coming out. To avoid the repletion of re-entry of the walking robot to the same obstacle, Virtual obstacle construction behavior is activated.

The path reminder behavior

In order to minimize the number of iterations obtained by the robot to avoid obstacles, the path reminder behavior is enabled. This behavior is enabled when the distance of obstacles from all sides is less than one (i.e., the robot is entrapped in a dead end), path reminder enabling algorithm is shown below.

1) If the robot distance from front, left and right obstacles is greater than one, go to Step 3; otherwise, if the number of iterations of the program is less than 10 add one to program iterations and go to Step 9.

2) If the current position of sensor is a stored one, and the number of program iterations is more than 10, go to step 6.

3) Save the current position.

4) Make number of program iterations zero.

5) Go to step 9.

6) Enable The path reminder behavior.

7) Put the obstacle type of side one.

8) Enable obstacle avoidance behavior.

9) End

Fuzzy rules to remind path and help out of the deadlock (when the robot distance from front, left and right obstacles is less than one) are summarized as follows:

A)If the robot moves toward the target and target is to the bottom (top) and left (right), the robot moves upward (downward) and the robot should rotate to the left (right) with medium rotation angle .

B)If the robot moves toward the target and the target is to the bottom (top) and robot steering angle φ, is zero, the robot moves upward (downward) and the robot should rotate to the left with small rotation angle.

C) If a robot is moving towards the target and the target is on the top (down) and left (right), the robots moves to the right (left), and the robot should rotate to the right (left) with the average rotation angle.

D) If a robot moves towards the target and the target is to the left (right) and robot steering angle φ, is zero, the robot moves to the right (left) and the robot should rotate to the left with small rotation angle.

E) If the robot is not moving toward the goal, then do the goal seeking behavior.

Virtual obstacle construction behavior

With the introduction of the path reminder behavior, the number of occurrences for the robot to evade the obstacles are much reduced. But in some cases, after escaping from deadlock, robot must move back and therefore robot tangles in the same deadlock, again. This problem is solved with constructing the virtual obstacle. When a deadlock is detected, virtual barrier allows the robot to the area around the barrier to entry is prohibited. Only the robot is able to detect the obstacle that is virtually made. Therefore, the robot will not enter back into the prohibited area.

The virtual barrier is created when the side obstacle avoidance behavior and path reminder behavior are enabled. The virtual barrier is created only when the robot is far away from the target. And it is detected by the distance between robot and the target. If the distance is great, i.e. robot is receding from the target. On the other hand, if the distance between the robot and the target is becoming low, meaning that the robot is approaching the target. Creating a virtual barrier is stopped when the robot passes from the reference point. However, the reference point is set when the side obstacle avoidance behavior or path reminder behavior is activated. Algorithm to construct a virtual obstacle is shown below:

1. If the side obstacle avoidance behavior or path reminder behavior is activated go to Step 3.

2. If the sensor detects an obstacle go to Step 6, otherwise go to Step 7.

3. Active the Construction of virtual obstacle behavior.

4. If the current position is a stored position (in the x or y direction) go to Step 5, otherwise Go to Step 7.

5. Create a virtual obstacle and go to step 7.

6. Save the current position.

7. End

Combining basic behaviors

For compounding the four basic behaviors, the behavior controller is designed. The process of the behavior controller can be explained as follows:

Step 1: Always, the goal seeking behavior begins the behavior controller process. If there are no obstacles in the path of the robot to the target, the robot goes directly to the target. Otherwise go to the second step.

Step 2: When the robot encounters an obstacle, the controller drops the goal seeking behavior and acts a different behavior. The prior behavior based on the distance between robot and obstacle is selected. The threshold of the distance to obstacle is defined 1 meter. The relationship between the obstacle and the robot will evaluate and then go to Step 3 to decide which behavior is prior to control the robot.

Step 3: If the relationship between the threshold and the distance to the obstacle is as follows:

(DL> 1) & (DR> 1) & (DF> 1) or ((DL <1)|(DR <1)) & (DF <1)

Obstacle avoidance behavior will be prior behavior. So, robot avoids from obstacles. If the relationship is as follows:

(DL <1) & (DR <1) & (DF> 1) or ((DL <1) | (DR <1)) & (DF> 1)

The obstacle converts to a side obstacle. If the relationship is as follows:

(DL <1) & (DR <1) & (DF <1)

The deadlock disarming behavior will be the prior behavior. In this situation path reminder and the virtual obstacle construction behavior is activated to help the robot to release from deadlock.
Step 4: The obstacles is detected and the distance between obstacles and the robot's current position is measured at the same time. If there are no obstacles in the path, the robot goes directly to the target. Otherwise go to the step 3 until the robot reaches to the target.

SIMULATION

The implementation of project consists of two main phases:

A) The design of the fuzzy controller system based upon the problem

B) Implementation of the problem code

The design of the fuzzy controller system based upon the problem

To implement and design of problem's fuzzy controller, the Fuzzy Interface System that's in the Fuzzy Systems toolbox on MATLAB software is used [10]. For designing the controller in this project the following steps must be performed:

1. Determining the input and output variables: in this section determines the input and output variables of the problem. The input variables are the distance to obstacles and the target. Also, the output variables are angle of rotation, linear speed and motion orientation. The robot speed is a number between 1 to 6 that is determined by experiences. Other values are illustrated in Figure 6.

2. Definition of membership functions: The next step is to define the fuzzy membership functions. These functions are shown in Figure 6.

3. Rules definition: In this part defines if - then rules of fuzzy systems.

Implementation of the Code

The GUI interface in MATLAB is used to implement the code [9]. Here, it is assumed that the initial position of the robot and desired animal and distance to obstacles get to system. Then the animal location is randomly changed to simulate the animal motion. The speed of animal is assumed the number between 0 to 4.This number is depended on the animal must be searched and it's changeable and experimental. Size, location and shape of the obstacles are determined in the designing environment time.

Here to simplify simulation in an unobstructed environment is considered. Of course, such an environment that actually exist in the same desert or having hills with a gentle slope that robot enables to climb them. The environment is considered the 100 m* 100 m square.

The initial position of the animal and the robot is shown in Figure 7(left). Here, an animal with a red circle in the initial position (15, 70) and a robot with a blue cross in the initial position (75, 20) are showed. These initial numbers is given by the operator from the user interface. If there is an obstacle, must be determine the position and distance of it to the robot.

In any time, the animal randomly changes its location by take a step; here step size between zero and 4 is defined. Also, the robot changes its location according to the animal displacement, to achieve a location that its distance to the animal is

less than 1m. This number can be changed in the design code. Figure 7 (right) shows the final position of the animal and the robot. Finally, as can observe, the robot is able to capture the animal.

CONCLUSION

Fuzzy logic uses a set of simple fuzzy logic if-then rules to solve the problem instead of trying to model a system in mathematics. In this paper, fuzzy logic is used for navigating a robot that can find and catch the animals. This robot can build a hound robot or find endangered animals. This navigation method is based on the method presented in [5] together with the changes and reforms. This method is used a hierarchical behavior-based technique. This means that the robot first tries to find the target and if does not encounter the obstacles in the path goes towards it. If robot encounters an obstacle, depend on the distance to obstacle from front and left and right side, one of the obstacle avoidance behavior or deadlock disarming behavior can execute. To implement and test the proposed method the MATLAB software is used. The simulation of the proposed method indicates that Robot can follow the animals step by step and finally can reach it.

The proposed method not only has correctness and simplicity, but also does not need to special knowledge of robot structure. It is hoped that this new application of navigation robot helps to rescue the endangered animals.

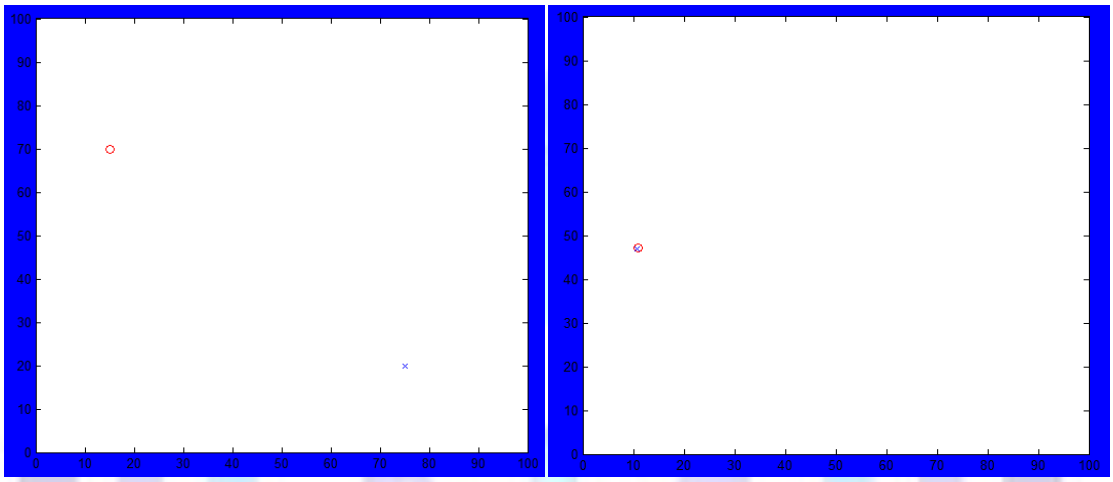


Figure 7. Left. the initial position of the robot and animal. Right. the final position of the animal and robot

References

[1] Torshabi, A.E., Riboldi, M., Pella, A., Negarestani, A., Rahnema M., and Baroni, G. 2012. A Clinical Application of Fuzzy Logic, a book chapter in Fuzzy Logic - Emerging Technologies and Applications, Prof. Elmer Dadios (Ed.), InTech publisher, 3-18.

[2] Rodjito, P. 2006.Position tracking and motion prediction using Fuzzy Logic, Honors Theses, Colby College, Paper 520.

[3] Hong, T.S., Nakhaeinia, D., and Karasfi, B. 2012. Application of Fuzzy Logic in Mobile Robot Navigation, a book chapter in Fuzzy Logic – Controls, Concepts, Theories and Applications, Prof. Elmer Dadios (Ed.), InTech publisher, 21-36.

[4] N. Daei. 2011. MSc Thesis: Navigation of autonomous mobile robot in indoor dynamic semi-structured environment , Iran, payam noor university,

[5] Dongshu, W., Yusheng Z., and Wenjie S. 2011. Behavior-based hierarchical fuzzy control for mobile robot navigation in dynamic environment, in Proceedings of Control and Decision Conference (CCDC), 2419 - 2424

[6] Raguraman, S.M., Tamilselvi, D., and Shivakumar, N. 2009. Mobile Robot Navigation Using Fuzzy logic Controller, in Proceedings of International Conference On “Control, Automation, Communication And Energy Conservation”, 1-5

[7] Abdel, M., Jaradat, K., Al-rousan, M.,Qaudan, L. 2011. Reinforcement based mobile robot navigation in dynamic environment, Robotics and Computer-Integrated Manufacturing, vol. 27, 135–149

[8] Pratihar, DK., Deb, K., Chosh, A. 1999. A genetic-fuzzy approach for mobile robot navigation among moving obstacles, International Journal of Approximate Reasoning, vol. 20, 145–72.

[9] Sincanandam, SN., Sumathi, S., and Deepa, SN. 2007. Introduction to Fuzzy Logic Using MATLAB. Berlin, Germany, Springer-Verlag Berlin Heidelberg,

[10] Fuzzy Logic Toolbox , Math Works, http://www.mathworks.com/academia/student_version /r2013a_products/fuzzy-logic.pdf

[11] Mucientes, M., Iglesias, R., Regueiro, CV., Bugarin, A., Carifiena, P., and Barro, S. 2001. Fuzzy temporal rules for mobile robot guidance in dynamic environments, IEEE Transactions on Systems, Man, and Cybernetics, vol. 31(3),391–398.

[12] Mucientes, M., Moreno, D. L., Bugarín, A., and Barro, S. 2006. Evolutionary learning of a fuzzy controller for wall-following behavior in mobile robotics. Soft Computing, vol. 10(10), 881-889.

[13] Ashokaraj, I., Tsourdos, A., Silson, P., White, B., and Economou, J. 2004. Feature Based Robot Navigation: Using Fuzzy Logic and Interval Analysis, IEEE International Conference on Fuzzy Systems, vol.3, 1461 - 1466

[14] Parasuraman, S., Ganapathy, V., and Shirinzadeh, B. 2008. Mobile Robot Navigation using Alpha Level Fuzzy Logic System: Experimental Investigations, IEEE International Conference on Systems, Man and Cybernetics (SMC), 1878-1884.

[15] Ayari, E., Hadouaj, S., and Ghedira, K. 2010. A Fuzzy Logic Method for Autonomous Robot Navigation in Dynamic and Uncertain Environment Composed with Complex Traps, Fifth International Multi-conference on Computing in the Global Information Technology, 18-23.