



Guiding Mobile Robot by Applying Fuzzy Approach on Sonar Sensors

Nabeel K. Abid Al-Sahib* Ahmed Rahman Jasim**

Department of Mechatronics Engineering/ Al-Khwarizmi College of Engineering/ University of Baghdad

* Email: dr_nabeelalsahab@hotmail.com

** Email: engmktron@gmail.com

(Received 4 June 2009; Accepted 10 August 2010)

Abstract

This study describes how fuzzy logic control FLC can be applied to sonars of mobile robot. The fuzzy logic approach has effects on the navigation of mobile robots in a partially known environment that are used in different industrial and society applications. The fuzzy logic provides a mechanism for combining sensor data from all sonar sensors which present different information. The FLC approach is achieved by means of Fuzzy Decision Making method type of fuzzy logic controller. The proposed controller is responsible for the obstacle avoidance of the mobile robot while traveling through a map from a home point to a goal point. The FLC is built as a subprogram based on the intelligent architecture (IA). The software program uses the Advanced Robotics Interface for Applications (ARIA), it is programmed with C++ package (Visual C++.Net), and Networking software is used for setup Wireless TCP/IP Ethernet-to-Serial connection between robot and PC. The results show that the developed mobile robot travels successfully from one location to another and reaches its goal after avoiding all obstacles that are located in its way. The platform mobile robot is a Pioneer 3 DX that is equipped with Sonar sensors.

Keywords: FLC, mobile robot, sonar sensors, navigation, fuzzy decision making method, obstacle avoidance, intelligent architecture.

1. Introduction

One of the main objectives of mobile robotics is the construction of autonomous systems that are capable of moving in real environments without the help of a human operator. There are numerous difficulties in the attainment of this objective, due principally to the fact that real environments are, generally, uncertain, unknown and dynamic. Thus, the knowledge which is available a priori on these environments may be non-existent, incomplete, uncertain or imprecise, or the environments may undergo modifications over time, which renders them more complex and unpredictable [1]. Pioneer 3 DX is a mobile robot which has sonar sensors, [2]. The data from sonar sensors is the distance between the sonar and object. The sonar sensor arrangement for mobile robot is shown in Figure 1.

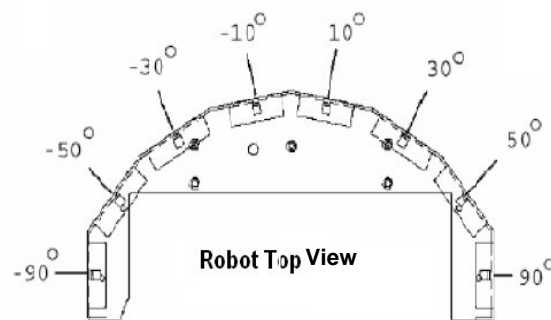


Fig. 1. The Sonar Arrangement for p3dx.

The sonar sensors locations are important for designing the fuzzy controller. The application of fuzzy control to obstacle avoidance has been implemented for Pioneer robot. All sonar sensors send data to the inputs of fuzzy controllers. Each fuzzy controller for obstacle avoidance has two

inputs and two outputs. Inputs have membership functions. Each membership function is considered as a triangle function. In this paper, the fuzzy controller for the orientation of a robot is described. All fuzzy controllers are implemented under the fuzzy decision making approach. Fuzzy decision making approaches have been established as a main alternative to conventional robot control in recent years [3]. The system architecture, implemented in the IA in this application, has two objectives. The first one is the obstacle avoidance and the second one is the orientation behavior.

2. Navigation

The navigation of a mobile vehicle can be considered as a task of determining a collision free path that enables the vehicle to travel through an obstacle course from an initial configuration to a goal configuration [4].

3. Obstacle Avoidance

The obstacle avoidance algorithm is based on the relative angle and distance between a robot and other objects; each time the robot calculates the relative angle and distance by sensor of robots. Autonomous mobile robot has sensors and the robot calculates the relative angle and distance. These calculative values are important factors of algorithm [5].

4. Fuzzy Control

Fuzzy logic control is selected as it is easiest to implement for a highly nonlinear robot model. Although a relatively new concept, fuzzy logic is being used in many engineering applications because it is considered by designers to be the simplest solution available for the specific problem[6].

Fuzzy Decision Making FDM is a useful method of fuzzy control theory to implement different behavior of work. Within FDM, it may form the basis for the implementation of behaviors as modular building blocks for complex control by using the techniques of fuzzy control. The concepts of Fuzzy Decision Making theory found wide application in various areas such as robotics path planning, navigation and mission planning, it has the potential to benefit from the

application of fuzzy logic to provide for soft decisions when there is the need for decision making under uncertainty[3].

The advantage of Fuzzy logic control is that it enables control engineers to easily implement control strategies which can be used by a human operator. [7].

There are three steps for designing a fuzzy controller: fuzzification, inference engine, and defuzzification.

4.1 Fuzzification

Fuzzification is defined as the mapping from a real-valued point to a fuzzy set. In most fuzzy decision systems, non fuzzy input data is mapped to fuzzy sets by treating them as singleton membership functions, Gaussian membership functions, triangular membership functions, etc.

4.2 Inference Engine

Fuzzy Inference Engine is used to combine the fuzzy IF-THEN in the fuzzy rule-base, and to convert input information into output membership functions. An inference mechanism emulates the expert's decision-making in interpreting and applying knowledge about how to perform good control. The fuzzy inference rules are based on reasoning similar to that of a human driver [8].

This can be implemented as a fuzzy rule-base. The rules may use the experts experience and control engineering knowledge. There are three types of fuzzy rule-based models for function approximation: Mamdani model, Takagi-Sugeno-Kang model, and Kosko's additive model.

4.3 Defuzzification

There are many methods which can be used for converting the conclusions of the inference mechanism into the actual input for the process or for the plant.

Center of Gravity (COG) defuzzification method is defined as equation (1).

$$Y = \frac{\sum_{i=1}^n \mu_A(y_i) y_i}{\sum_{i=1}^n \mu_A(y_i)} \quad \dots(1)$$

Where : Y= controller output, n= the number of rules, $\mu_A(y_i)$: member ship function of fuzzy set A calculated for the input variables, y_i : the center of area of of fuzzy set A_i of i-th of the rule, Other defuzzification techniques, are Center-Average defuzzification, Maximum defuzzification, etc. [3].

5. Obstacle Avoidance Design

The fuzzy controllers for obstacle avoidance are designed as follows:

First read sonar data and construct three membership functions for input. All data from sonar sensors are received and displayed. Three membership functions for each input in its universe are constructed. Each membership function is considered as a triangle membership function. The rules for avoiding frontal obstacles are developed in many stages. The sonar notations are the same as shown in figure (2) for convenience. The rule base of each table depends on the sensor arrangement.

A fuzzy controller can be designed as follows:

The input membership functions represent the right and left side readings of distance from the object ,if observed, in path by signal from Sonar sensors, Avoid Object on Right side (AOR) and Avoid Object on Left side (AOL).

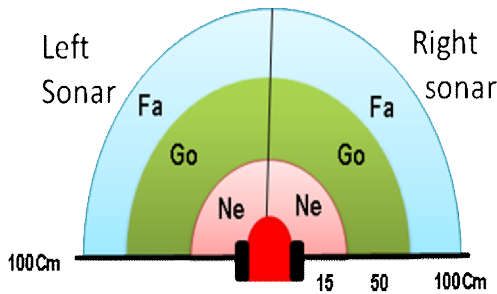


Fig.2. Sonar Range Notation for Robot.

The membership function is a three triangular membership function representing robot conditions that are good, near and far from the object lines.

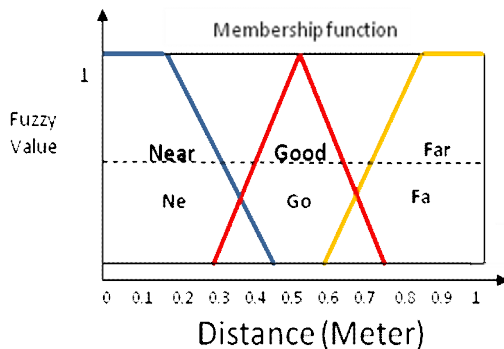


Fig.3. Avoidance Input Membership Function.

Table (1) shows the relation between linguistic input and ranges. The near and far membership functions overlap the good membership function, so that the distance can be both good, and near the object line, or both good and far from the line, this overlap in the membership functions allows application of multiple rules in this region. This membership functions are used for right sensors and left sensors inputs.

Table 1, Linguistic Input Fuzzy Set and Their Ranges.

Sonar distance (AO)ranges		
Linguistic Variable	Notation	Range in meter
Near	Ne	[0.00,0.45]
Good	Go	[0.25,0.75]
Far	Fa	[0.55, 1.0]

6. Orientation Control Design

The direction angle membership function is a 5- triangular membership functions, representing angle conditions that are very negative, negative, zero, positive, and very positive.

The objective of the orientation is to keep the robot pointing toward a given target so that progress is made. If the robot has to travel from a known present location (Xrobot , Yrobot) to a goal (Xgoal,Ygoal); then the following procedure is performed.

$$\Delta X = X_{robot} - X_{goal} \quad \dots(2)$$

$$\Delta Y = Y_{robot} - Y_{goal} \quad \dots(3)$$

Figure (4) represents the robot orientation with goal with orientation respect to global map.

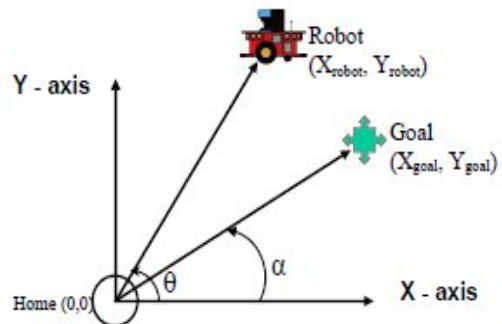


Fig.4. Robot Orientation with Respect to the Goal.

θ : The angle formed between the robot center and the X-axis.
 α : the angle formed between the goal center and the X-axis.
 Figure (5) represents the direction membership function.

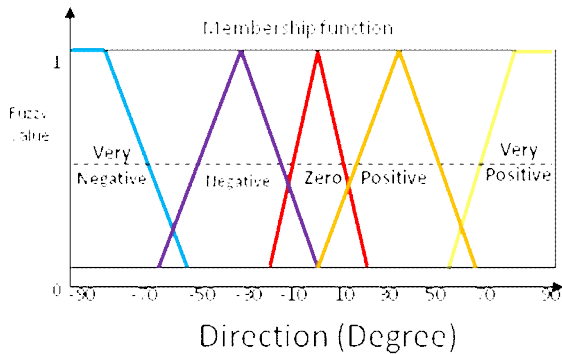


Fig.5. Direction Membership Function.

The output direction angle membership function is a 5-triangular membership functions, representing output angle conditions that are: Big Left, Left, Zero, Big Right and Right. Figure (6) represents direction output membership function.

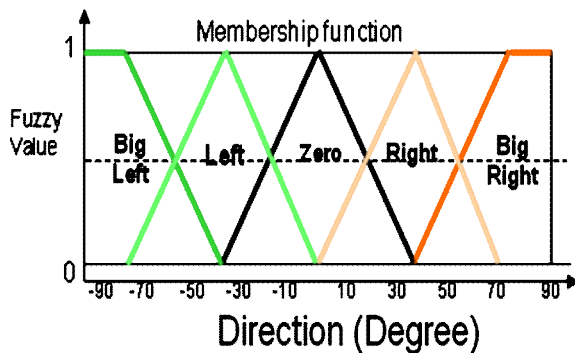


Fig.6. Direction Output Membership Function.

The fundamental principle of all of the controller logic is to bring the robot in the path, which means defining a start point and the goal. The path is drawn to the robot by programming a fuzzy logic system into the controller code. This information is used to select the robot angular velocity (VR) and output direction by rule base statements, as follows:

Rule Base Statements

- If (Dir = P) and (AOR = Ne) and (AOL = Far) then (VR= SF) and (DirR=L)
- If (Dir = VP) and (AOR = Ne) and (AOL = Far) then (VR= S) and (DirR=L)
- If (Dir = VN) and (AOR = Go) and (AOL = Far) then (VR= SF) and (DirR=BL)
- If (Dir = N) and (AOR = Go) and (AOL = Far) then (VR= SF) and (DirR=L)
- If (Dir = Z) and (AOR = Go) and (AOL = Far) then (VR= F) and (DirR=Ze)
- If (Dir = P) and (AOR = Go) and (AOL = Far) then (VR= SF) and (DirR=R)

For output variables:

- F: Forward
- SF: Slow Forward
- S: Stop
- SB: Slow Backward
- B: Backward
- BL: Big Left
- L: Left
- R: Right
- BR: Big Right
- Ze: Zero

In the step of defuzzification , the membership function for the control action of angular velocity is defuzzified by using the centroid method.. Finally, the output for angular velocity is given by equation (1).

7. Robot Description

The platform used in work is Pioneer3 DX which is a family of two wheel mobile robots. The robot has reversible DC motors, motor-control driver, high-resolution motion encoders, and battery power; all are managed by an onboard microcontroller and mobile-robot server software as shown in figure (7). The software includes foundation Advanced Robotics Interface for Applications (ARIA) and Advanced robot for ARNetworking .



Fig.7. Pioneer Platform.

The mobile robot platforms require a PC connection: software running on a computer connected with the robot’s microcontroller via the HOST serial link and which provides a high-level mobile robot software, intelligent robot controls, Advanced Robotics Interface for Applications (ARIA) software and ARNetworking. ARIA is a C++ based open-source development environment that provides a robust client-side interface to a variety of intelligent robotics systems, including the robot’s microcontroller and accessory systems.

ARNetworking Software provides the critical layer for communication with mobile robot platform over the network via Transmission Control Protocol/Internet Protocol (TCP/IP), The TCP/IP is the most popular network protocol, and the basis for the Internet. Its routing capabilities provide maximum flexibility in an enterprise-wide network. On a TCP/IP network, IP must provide addresses to clients. Clients may also require a naming service .

8. Wireless Ethernet-to-Serial Connection

Wireless Ethernet to Serial is a little more complicated, but is the preferred method because it can use many different computers on the network to become the robot’s client so allowing greater degrees of flexibility and interoperability between mobile devices [9].

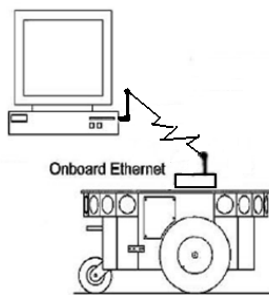


Fig.8. Client-Server Connection.

A wireless Ethernet-to-serial connects directly to robot’s microcontroller, it works by automatically translating network-based Ethernet packet communications TCP/IP into streaming serial for the robot microcontroller and back again. A major disadvantage of the wireless Ethernet-to-serial device is that it requires a consistent wireless connection with the robot; disruption of the radio signal a common occurrence in even the most modern installation

leads to poor robot performance and to very short ranges of operation. Secure connection is important to save robot from undesirable connection. Figure(8) shows client-server connection.

9. Software program

The programmer usually includes idea along the path at space intersections or other choice points which the robot should achieve as quickly as possible, taking into consideration safety and power. And as the robot work to achieve those larger goals, it should avoid obstacles and respond to accident, such as closed doors or blocked way, Go-to-Goal is defined for path. It is a value of directed difference to the nearest goal with respect to current robot position.

Figure (9) shows the flowchart of the program, a crisp value for AOL, AOR, DirC and DirR input to program; then they are fuzzified to linguistic variables such as “ Object near left sonar “, in program inference system list the rule base statments like “ if *Dir* = *positive* and *AOR* = *Negative* then *robot angle is positive*”. By defuzzifying these values, it puts crisp output parameters like “ robot angle = +30” send to robot platform.

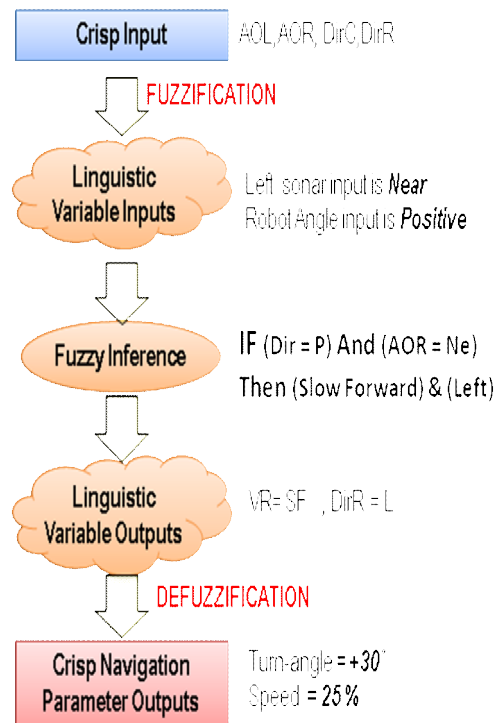


Fig. 9. The Flowchart of the Program.

10. Obstacle Avoidance and Orientation Behavior

The obstacle avoidance behavior uses sonar sensor data to generate a fuzzy set that represents the distance relating to robot location. The behavior is operated by using the fuzzy controller. The fuzzy inputs are the fuzzy sets from front, left and right. The sensors on the front-left and the sensors on the front right are used for obstacle avoidance behavior. Therefore, the fuzzy controllers are required in this behavior. The output is the avoiding rotation. The results of obstacle avoidance behavior will be shown in the following cases.

Go-to-Goal program is a task effect on orientation behaviors.

11. Case Study

Cases are based on building’s map in which some fixed elements of the environment are located with Known positions: walls, doors, furniture, etc.

But unfixed elements, whose positions are unknown, can be added to the environment.

11.1. One Wall Map without Obstacle

Map drawn with dimension 3m Y axis and 5m X axis ,home point = (0.0, 0.0) and goal position = (-4.0, 0.0) while wall = 75cm long , 10cm width ,on 2.5m left from home is shown in figure(10).

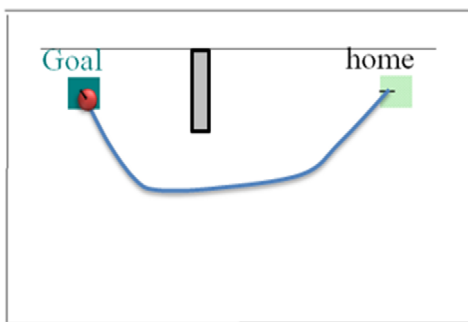


Fig.10. Robot Behavior on One Wall Environment.

An obstacle is added to the environment of the last map. The obstacle is located = (-1.5, -1)m from home.

When the obstacle is detected near the robot, FLC turns the orientation of the robot right until the robot escapes from the obstacle after the robot

crosses its the obstacle; then it tries to return back to the last path and move toward goal, as shown in figure(11).

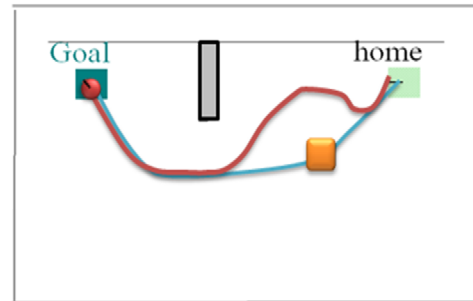


Fig.(11) obstacle added to environment

11.2. Three Obstacles Environment

A (5,3)m map, robot started with position (0,0); then the robot tried to take straight path but observing the obstacles induces the robot to turn left. After crossing the obstacle, the robot turns right straightly to the goal, as shown in figure (12).

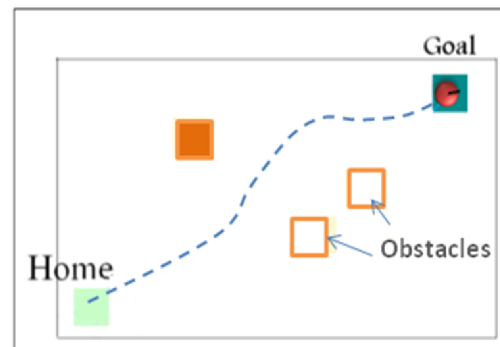


Fig.12. Robot Behavior in Three Obstacles.

12. Results

The result will be represented in a graph, where vertical left axis represents robot velocity (cm/sec) through the map, the right vertical axis represents the angle (degree) of robot , and horizontal axis represent the time (sec.) of case .

In case (1), one wall map without obstacle : figure (13) shows the relation between robot velocity , angle and time. The velocity of robot decreased at point (A) when robot try to avoid wall and turned right at point (B) , with straight open area motion in point (D) velocity increases at point (C) , after avoiding the wall ; the robot tried to reach goal and turn left in point (E) in the

end when the robot reached near the goal, velocity was decreased to zero and got goal orientation in point (F).

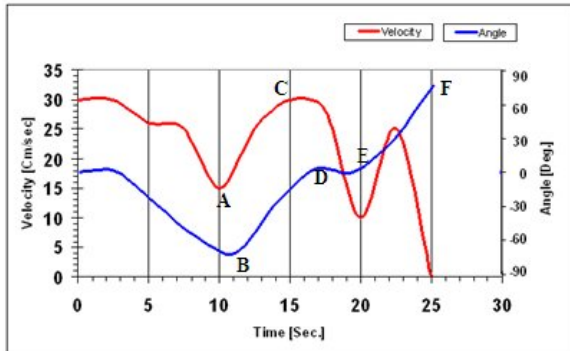


Fig.13. Navigation Without Obstacles.

Case (2): When an obstacle was added to the environment at the path of last map, the robot primary turned right at point (A). That caused reducing the velocity shown in figure (14) in point (B). The next points (C) and (D) represent a turn of robot to the left when it observed an obstacle and then returned to the save path of motion as shown in figure (13).

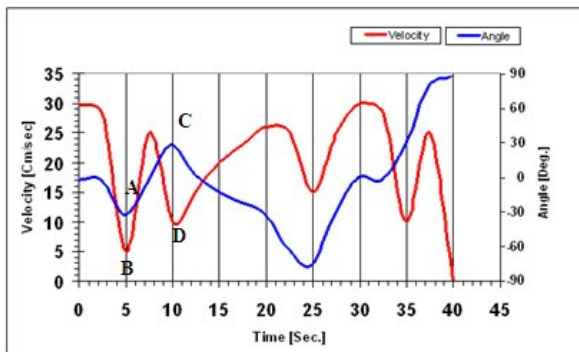


Fig.14. Navigation With Obstacles.

So a safe navigation of motion with a full known environment is approved in case1, and a safe navigation of motion with a partial known environment is approved in case 2.

13. Conclusions

Studying mobile robot control by FLC leads to many conclusions, as follows:

- 1- The differential mobile robot includes many kinematics constraint, which can't be reduced at one equivalent constraint of configuration

variables, so that using the methods which avoid the analytical modeling problem, such as fuzzy control methods, and neural network control.

- 2- The navigation system used by the robot is based on two receivers, robot current position and goal .
- 3- The Robot movement in case of FLC programming improves the robot navigation by avoiding damag itself.
- 4- A decision making program is constructed for the local-path planning. It is not necessary to define a reference point prior to home.
- 5- For obstacle avoidance, two types of execution panels are carried; one with landmarks and the other with objects. The latter is used to generate complex environments for the mobile robot to its ability to avoid obstacles. The obstacle avoidance success by using fuzzy rules.
- 6- The effect of varying sensor range showed that it does not only affect the distance between the mobile robot, obstacles and walls, but also affects the shape of the path to the goal.

14. References

- [1] M. Mucientes, R. Iglesias, C.V. Regueirob, A. Bugar-na, S. Barroa, "A fuzzy temporal rule-based velocity controller for mobile robotics", Fuzzy Sets and Systems, Vol. 134, pp. 83–99, 2003.
- [2] Pioneer reference book, www.mobilerobotics.com. 2008.
- [3] Kevin M. Passino, Stephen Yurkovich, "Fuzzy Control", Addition Wesley Longman, Inc., California, 1998.
- [4] Hani Hagra, Victor Callaghan, Martin Colley, "Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online Fuzzy–Genetic system", Fuzzy Sets and Systems, Vol. 141, pp.107–160, 2004.
- [5] Jang Hyun Kim¹, Jin Bae Park², and Hyunseok Yang, "Implementation of the Avoidance Algorithm for Autonomous Mobile Robots Using Fuzzy Rules", LNAI, Vol. 4223, pp. 836–845, 2006.
- [6] Vamsi Mohan, Dan Simonm, "FUZZY LOGIC CONTROL FOR AN AUTONOMOUS ROBOT", 2004.
- [7] Rerngwut Choomuang , Nitin Afzulpurkar , "Hybrid Kalman Filter/Fuzzy Logic based Position Control of Autonomous Mobile

- Robot", International Journal of Advanced Robotic Systems, Vol. 2, pp. 197 - 208, 2005.
- [8] Elie Maaloufa, Maarouf Saada, "A higher level path tracking controller for a four-wheel differentially steered mobile robot", Robotics and Autonomous Systems, Vol.54, pp. 23–33, October, 2005.
- [9] K.A.Hawick and H.A.James, "Middleware for Wireless Sensors and Robots", Technical DHPC-112, 2002.

