# A Design of Fuzzy Controller for Autonomous Navigation of Unmanned Vehicle

**Vinod Kapse** Kapse.vinod@rediffmail.com
*Research Scholar, Jabalpur Engineering College,*
*Jabalpur , India*

**Bhavana Jharia** bhavanajharia@yahoo.co.in
*Dept. of Electronics and Communication Engg.*
*Jabalpur Engineering College,*
*Jabalpur , India*

**S. S. Thakur** samajh_singh@ rediffmail.com
*Department of Applied Mathematics.*
*Jabalpur Engineering College,*
*Jabalpur , India*

**Abstract**

This paper presents a design approach for fuzzy logic controller for autonomous navigation of a vehicle in an obstacle filled environment with obstacle avoidance layer, orientation control layer and passage detection module. It provides a model for multiple sensor input fusion and is composed of eight individual controllers. Each controller will calculate collision possibility in different directions of movement, according to which main controller would take action to avoid the collision. The designs have been carried out in the digital domain with VHDL using Altera Quartus-II Software.

**Keywords:** Fuzzy controller, Autonomous Navigation, Sensor, Collision Avoidance, Digital Domain.

## 1. INTRODUCTION

The field of Robotics is advancing very rapidly and navigation is one of the main issues in this field. The Recent developments in this field and automation have made human life a lot comfortable and safer. The robots can do jobs that are difficult, dangerous, or dull. Robots have also been developed for clearing landmines, autonomous wheel chair for disabled, lawn mowing, vacuum cleaning etc. The *Mars Sojourner* developed by NASA is a semi-autonomous robot that explores the Mars, while taking its instruction from Earth. These examples are of mobile robots that travel from one location to another either randomly or by following the defined instructions. These mobile robots can be categorized as Automatic Guided Vehicles (AGV) and Autonomous Mobile Robots (AMR). The AGV's navigate with a description of the environment in their memory, and thus have a limited flexibility and application. Autonomous Mobile Robots, on the other hand, are more flexible and adapt quickly to different environments. They do not have any preset description of the environment in their memory and rely on the sensor information and the control algorithm to achieve their target. This makes it all the more important for an autonomous mobile robot to have a good control algorithm so as to adapt to the changing environment. Various

Control algorithms like hardwired control and traditional control model have been implemented for navigational purposes [1-3].

Ajit P. Khatra et. al. presented controller with modular, multi-layered autonomous mobile robot capable of traversing through an unknown environment, avoiding all the obstacles, and reaches a pre-defined destination [2]. The specified control scheme for the robot had been implemented and tested successfully on a microcontroller but they are not precise to achieve more accurate control of the robot. The other fuzzy based controller discussed by I. Baturone et. al.[4] is capable of performing different tasks that an expert driver would performed, like deciding the driving direction, the speed magnitude, and the turning of the steering wheel when driving forward or backward. But in previously proposed controller nothing is discussed about the alignment of robot, collision possibility, status of battery backup and terrain selection etc. To design a more capable controller these areas need to be considered. Another fuzzy logic controller, proposed by L. Doitsidis et. al.[5] is capable of detecting the collision possibility only for the four direction i. e. front, back, left and right but there are other direction where the collision possibility need to be detected. Nowadays, several circuits known as fuzzy coprocessors are available in the microelectronics market place. They are general-purpose devices that work together with standard processors to speed up some of the typical operations of fuzzy-logic-based inference systems. But this kind of circuits is not efficient enough in terms of silicon area, power consumption and inference speed when considered for the applications in industrial sectors related to telecommunication, automotive or consumer products. A more advantageous solution is to use dedicated hardware adapted to the particular problem. Viability of this approach is greatly increased by using design methodologies, circuit techniques and CAD tools that ease system realization, thus reducing time-to-market [2].

Considering all above issues it is thought to design an FPGA based fuzzy Controlled Autonomous Mobile Robot System.

## 2. DESIGN APPROACH

From a conceptual point of view, autonomous navigation of robotic vehicles is achieved via continuous interaction between perception, intelligence and action. Navigation of autonomous robotic vehicles in obstacle filled dynamic environments requires derivation and implementation of efficient real-time sensor based controllers. Effective control algorithms for autonomous navigation, should imitate the way humans are operating manned or similar vehicles.

The main objective of designing autonomous robot controller is to perform tasks without human intervention. The mobile robot will be build based on the behavior-based artificial intelligence, where several levels of competences and behaviors will be implemented. A level of competence is a specification of a set of desired behaviors that the mobile robot will encounter in the real world. A higher level of competence indicates a more specific and complex desired class of behaviors. Individual layers can work on individual goals simultaneously. It will be directly connected to its problem domain through sensors and effectors. The system can change and affect its environment instantaneously by reacting through the effectors. Theoretical analysis of the fuzzy control algorithms of mobile robot control will be performed. The requirements for a suitable rule base selection in the proposed fuzzy controller will be provided, which can guarantee the asymptotical stability of the system. These rules may include:

- ⇨ The vehicle must maintain its alignment within established boundaries to define the environment
- ⇨ The direction of travel of the vehicle is usually fixed
- ⇨ The speed of the vehicle is restricted to an upper limit
- ⇨ The vehicle must not collide with other vehicle or the environment boundaries

The last point suggests that the vehicle must have a high level of autonomy in order to successfully detect and avoid collisions with vehicle and other objects in the environment. The sensor systems required for this autonomy typically consume considerable computing resources in order to successfully implementation the control strategy.

One of the current challenges in the development of robot control systems is to make them to give suitable response to adjust with changing environment. A fuzzy control system is proposed for car-like autonomous robots, which are used to solve a typical problem in motion planning of systems. The controller has a hierarchical structure made up of main modules in charge of the different tasks that an expert driver would perform: Orientation of vehicle, obstacle detection, Passage Detection, Deciding the driving direction, the speed magnitude, and the turning of the steering wheel when driving forward or backward.

Various control algorithms- hardwired control and traditional control model have been implemented for navigational purposes [1-6]. These algorithms are based on a mathematical model of the robot, which takes the sensor's information as inputs and uses fuzzy logic to generate the output. Navigation algorithm is divided into different layers to keep the algorithm simple and to have a better control of the robot. Each layer will model the behavior of the robot in different environments. A layered fuzzy algorithm on a modular platform to control the mobile robot; implemented in presented work is shown in fig. 1. Different layers are: Orientation Layer, Passage Detection (PD) Layer, and Obstacle Avoidance Layer. The Orientation layer assists the robot to keep itself pointed in the general direction of the goal frame to achieve it's final destination. The PD layer helps the robot to navigate through passageways. The Autonomous Mobile Robot (AMR) is prevented from crashing with an obstacle by the help of Obstacle Avoidance Layer.

Modular approach helps to distribute the processing power amongst its *sensor, motor driver and supervisor modules* and also, easy future enhancements in robot design.

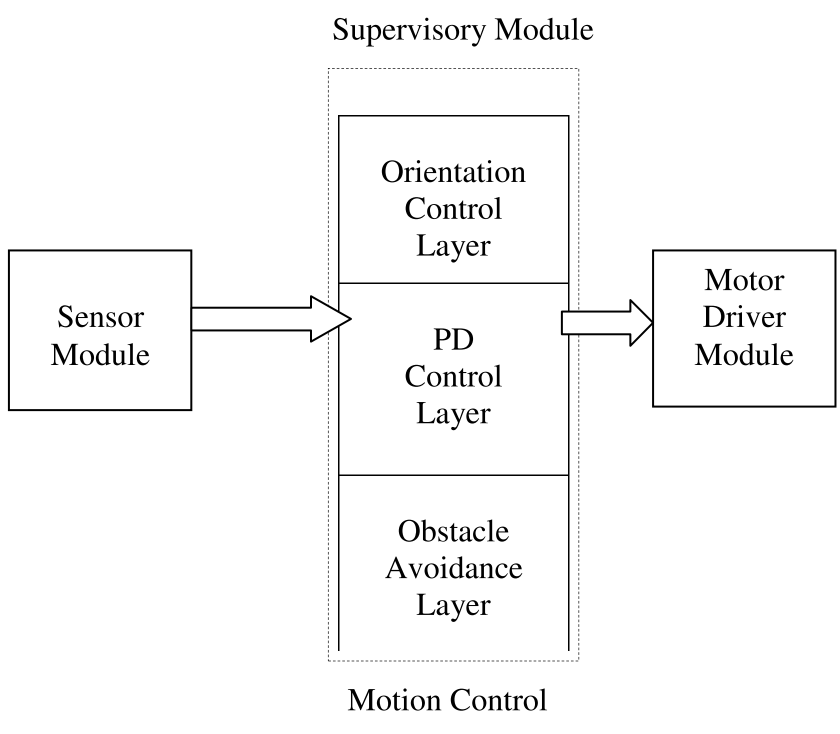


**FIGURE 1:** Block Diagram of System

The sensor module will sense all the obstacles in the path of the robot and if find any obstruction, it will compute the distance between them i.e. the obstacles and the robot; and notifies it to the motion controller to manage. The *motor driver module* maintains all the information needed to travel from source to destination distance traveled and also take care of movement of robot in a specific direction. The decision making part of the robot is handled by the supervisor module, which instructs the sensor module regarding the sensor sequence to be fired and commands the motor driver module to move in a particular direction along with the direction to turn if there is an obstacle. It also sets the speed with which the robot should move. These navigational decisions are made utilizing a control algorithm of the supervisor module.

### 2.1 Orientation Control Layer:

The Orientation Control Layer helps the robot to align with the goal frame. This layer will be active under various situations like: when the robot detects an obstacle, and/or when the robot is in the blind mode of the PD control Layer, etc. Thus, the resultant control action of the orientation layer depends on speed control of wheels and turn angle based on specified rules.

In order to ensure that the goals of unmanned autonomous mobile system should meet, the fuzzy controller must be supplied the following input data:

1) Distance from obstacle from Sensors
2) Alignment to maintain right path
3) The difference between the present orientation of the robot and the orientation of the goal frame

Sensed distance input will provide distance from the obstacle which is further useful for automatic avoidance of the obstacle, whereas Alignment input is used to track path of vehicle so that obstacle can be avoided. The difference between the present orientation of the robot and the orientation of the goal framed will be calculated by calculating the change in orientation angle.

These inputs are used as 8-bit control input by the fuzzy controller. If the distance and alignment inputs are 8-bit input then the range of input is from 00 to FF, which is equally distributed into 7 membership functions. There are various types of membership functions are available, among them the two most commonly used in practice are: the triangular and trapezoidal membership functions. Here, triangular membership functions are used. The membership functions for Distance Input consists of seven fuzzy logic ranges can be defined by using the linguistic terms as Tooclose, Close, Bitclose, Medium, Bitfar, Far, Toofar ; as shown in Fig.2.
Distance= {Tooclose, Close, Bitclose, Medium, Bitfar, Far, Toofar}

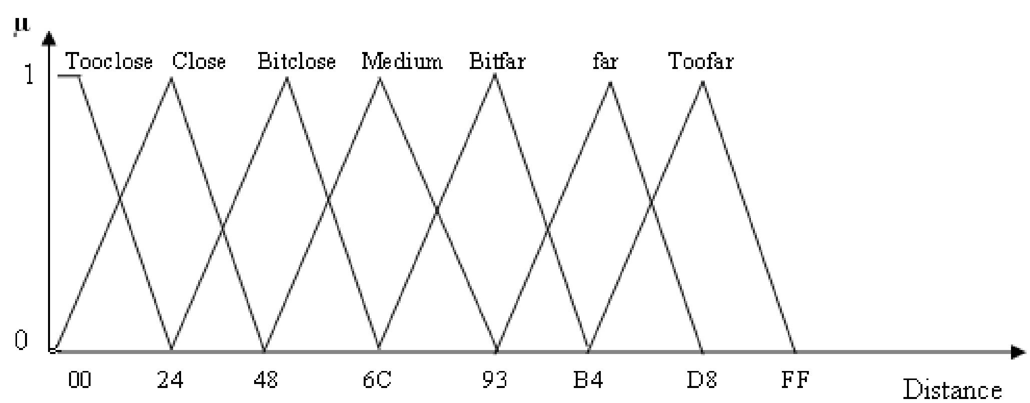


**FIGURE 2:** Triangular Membership Function for Distance

The membership functions for Alignment input consists of seven fuzzy logic ranges can be defined by using the linguistic terms as Farleft, Left, Bitleft, Center, Bitright, Right, Farright. The graphical representation of the member-ship function of Alignment input is depicted in Fig. 3.

Alignment= {Farleft, Left, Bitleft, Center, Bitright, Right, Farright}

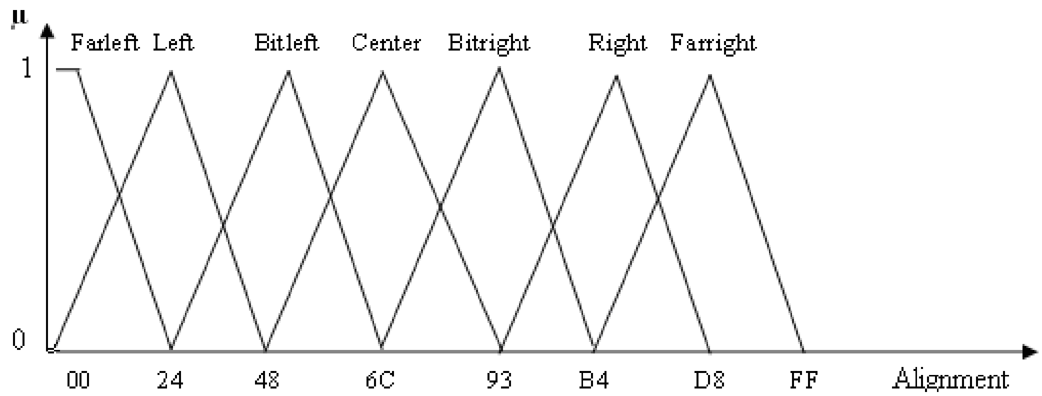


**FIGURE 3:** Triangular Membership Function for Alignment

Speed control for autonomous vehicle relates to the automatic maintenance of proper alignmentand the distance from obstacle using fuzzy control systems, without intervention from external supervisory systems or human. The rule base for given system are shown in table -1 and 2.

| | Input-Alignment | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Output LWS** | **Farleft** | **Left** | **Bitleft** | **Center** | **Bitright** | **Right** | **Farright** |
| **Input-Distance** | **Tooclose** | **Medium** | **Bitslow** | **Slow** | **Veryslow** | **Veryslow** | **Veryslow** | **Veryslow** |
| | **Close** | **Bitfast** | **Bitslow** | **Bitslow** | **Veryslow** | **Veryslow** | **Veryslow** | **Slow** |
| | **Bitclose** | **Fast** | **Fast** | **Bitfast** | **Veryslow** | **Veryslow** | **Veryslow** | **Bitslow** |
| | **Medium** | **Veryfast** | **Veryfast** | **Veryfast** | **Medium** | **Veryslow** | **Veryslow** | **Bitslow** |
| | **Bitfar** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** | **Bitslow** | **Slow** | **Medium** |
| | **Far** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** | **Bitfast** | **Bitslow** | **Medium** |
| | **Toofar** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** | **Fast** | **Bitslow** | **Medium** |

**TABLE 1:** Rule Base for LWS

| | Input-Alignment | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Output RWS** | **Farleft** | **Left** | **Bitleft** | **Center** | **Bitright** | **Right** | **Farright** |
| **Input-Distance** | **Tooclose** | **Bitslow** | **Veryslow** | **Veryslow** | **Veryslow** | **Slow** | **Bitslow** | **Medium** |
| | **Close** | **Slow** | **Veryslow** | **Veryslow** | **Veryslow** | **Bitslow** | **Bitslow** | **Bitfast** |
| | **Bitclose** | **Bitslow** | **Veryslow** | **Veryslow** | **Veryslow** | **Bitfast** | **Fast** | **Fast** |
| | **Medium** | **Bitslow** | **Veryslow** | **Veryslow** | **Medium** | **Veryfast** | **Veryfast** | **Veryfast** |
| | **Bitfar** | **Medium** | **Slow** | **Bitslow** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** |
| | **Far** | **Medium** | **Bitslow** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** |
| | **Toofar** | **Medium** | **Bitslow** | **Fast** | **Veryfast** | **Veryfast** | **Veryfast** | **Veryfast** |

**TABLE 2:** Rule Base for RWS

### 2.2 Passage Detection Module:

The PD control layer is used to steer the robot through passageways in an efficient manner without crashing into the walls. This can be achieved by keeping the robot at a safe distance from the walls by constant monitoring of the left and right sensor values. These sensor values are monitored after a fixed interval of time and a change in orientation is calculated at each instant. This change in the orientation is known as the *error input*.

### 2.3 Obstacle Avoidance Layer:

Up till now only four directions of detection of obstacle has been considered by other researchers [5]. In order to control the vehicle movement more efficiently, a fuzzy controller has been designed and implemented. Presented fuzzy controller will be more accurate compared to previous controller [5] because it is responsible for obstacle detection and calculation of the collision possibilities in the eight directions i.e. front(F), back(B), left(L), right(R), front_left(FL), back_left(BL), front_right(FR) and back_right (BR). The controller receives inputs from the sensor's data and provides the output by calculating the collision possibility in different directions front, back, left, right, front_left, back_ left, front_right and back_right. Further, the collision possibilities calculated by the controller are the input to the other controller along with the angle error (the difference between the robot heading angle and the desired target angle), which results in the output with updated transnational and rotational speed.

The sensor module; a fuzzy logic controller; will takes input data provided by the various sensors and delivers information for eventual obstacles in respect to vehicle's position and orientation. Eight sensors are mounted on eight different direction of vehicle with separation of 45 degrees from each other as shown in figure 2. Three sensors are sufficient to detect the collision possibility in a particular direction. For example to detect front_left collision possibility sensors A2, A3 and A4 are used to find the distance of obstacle from vehicle. Similarly, for other direction collision possibility group of three sensors may be used. Every sensor has a range of 22.5° obstacle detection to each side as shown in Fig. 4 by shaded portion where A2 is representing range of front obstacle detection.

The information collected from the respective three sensors will give the information about the obstacle detection so the collision possibility; which is further sends to the motion control module. The collision possibilities together with position and/or orientation error will act as the inputs of the fuzzy controller, which is responsible for the output commands to the driving devices. Also, on the rule base information it will provide potential collisions possibilities in eight different directions, and by processing information guarantees collision avoidance with obstacles while following the desired trajectory.

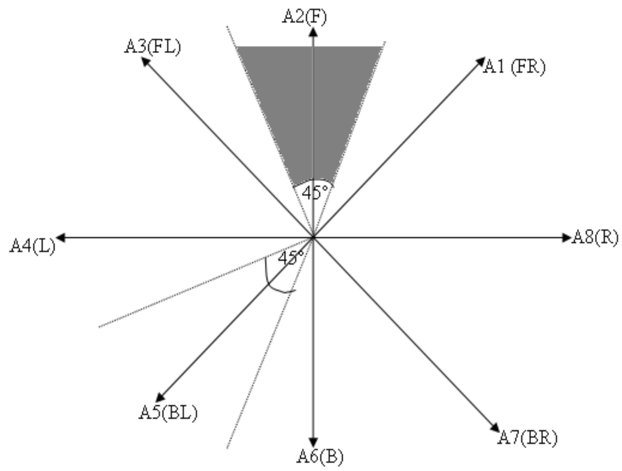


**FIGURE 4:** Angle of separation for mounted sensors

The fuzzy controller utilizes the membership functions to calculate the collision possibilities. The linguistic values of the variable distance_from_obstacle are defined to as near, medium, far with membership functions reflecting the accurate information about potential obstacles.

The output of fuzzy controller is a collision possibility in each direction taking values from 0 to 1. The linguistic variables describing each direction output variable collision possibilities are Nearly_possible, Impossible, possible, highly_possible membership function. A part of the rules base for left collision is presented in Table 3. For example of the rules used to extract left collision possibilities is: IF A1 is near AND A3 is near AND A4 is near AND A5 is near AND A7 is near AND A8 is near THEN collision_possibility is high. Similarly for the other collision possibilities rule base are designed.

| Inputs from Sensors | | | | | | | | Output |
|---|---|---|---|---|---|---|---|---|
| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | (Collision Possibility) |
| N | - | N | N | N | - | N | N | HighPossibility |
| N | - | N | M | M | - | M | M | Possibility |
| M | - | M | N | M | - | M | N | Possibility |
| M | - | M | M | N | - | N | M | Possibility |
| M | - | M | M | M | - | M | M | Possibility |
| F | - | F | M | M | - | M | M | NearlyPossible |
| M | - | M | F | M | - | M | F | NearlyPossible |
| M | - | M | M | F | - | F | M | NearlyPossible |
| F | - | F | F | F | - | F | F | Impossible |

**TABLE 3:** Part of Rule Base for controller

In presented work it is tried to develop the system for some input and output parameters and comparative study has been done to find out the effect of various approaches on hardware requirement and operating speed of circuit with various synthesis constraints like Speed and Area. The comparative study is shown in table 4. It is clear that implementation using state machine is better as compared to other approaches as per as hardware requirement and operating speed are concerned.

| **Implementation** | **Logic Element Requirement From Synthesis Report With Constraints (% Improvement of Logic Element w. r. t. Normal Implementation)** | | **Clock Speed(Mhz) From Timing Analysis Report With Constraints(% Decrement in Clock Speed w.r. t. Normal Implementation)** | |
|---|---|---|---|---|
| | AREA | SPEED | AREA | SPEED |
| Normal Implementation (VHDL) | 227 | 247 | 185.36 | 191.64 |
| Normal Implementation (VERILOG) | 227 | 247 | 185.36 | 191.64 |
| Implementation Using State Machine(VHDL) | 136 (67%) | 144 (71%) | 96.15 (92%) | 103.15 (85%) |
| Implementation Using State Machine(Verilog) | 133 (71%) | 143 (73%) | 65.36 (185%) | 68.49 (180%) |
| Implementation Using Overlapping Membership Function(VHDL) | 136 (67%) | 157 (57%) | 103.09 (80%) | 104.17 (84%) |
| Implementation Using Less No.Overlapping Membership Function(VHDL) | 124 (83%) | 135 (83%) | 89.29 (108%) | 123.46 (55%) |
| Conventional Fuzzy Controller with Triangular Membership Function(VHDL) | 4575 | 4930 | 5.35 | 5.49 |
| Conventional Fuzzy Controller with Trapezoidal Membership Function(VHDL) | 4573 | 4971 | 6.51 | 7.17 |

**TABLE 4:** Comparative Summary of Synthesis and Timing Analysis Report

## 3. CONSLUSIONS

A fuzzy logic controller for autonomous navigation system is designed , simulated and implemented on FPGA using Quartus-II Altera tools. The design of the FLC is highly flexible as the membership functions and rule base can be easily changed.

Presented digital fuzzy system contains logic circuits to compute the fuzzy algorithm, memories to store fuzzy rules, and generators or look-up tables for membership functions of the input and output variables. The proposed designs are programmed using QUARTUS II (ALTERA) tool. Comparative results of operating frequency and logic elements requirement for different approaches are shown in table 2. Fuzzy logic methods have been proved to be effective tools to design highly responsive controllers for autonomous mobile system. These controllers are capable of implementing motion and perception behavior so as to attain multiple possibly conflicting goals.

Focus of the work was on the application of fuzzy logic techniques to the design and implementation of basic behaviors, and to the combination of basic behaviors to form complex behavior to execute full navigational plans. The continual developments on FPGA / CPLD technology and their associated cost, and reprogram ability make this approach a viable alternative to the development of custom fuzzy hardware for real-time applications.

The proposed controller is an attempt for Blind Goal-Oriented Navigation that the robot is required to autonomously reach a desired goal but it does not have a priori known environmental knowledge. The results have been analyzed, verified and found satisfactory.

## 4. REFERENCES

[1] S. X. Yang et. al. *"An Embedded Fuzzy Controller for a Behavior-Based Mobile Robot ith Guaranteed Performance".* IEEE TRANSACTIONS ON FUZZY SYSTEMS, 12(4):436-446, 2004

[2] A. P. Khatra et. al. *"A Multi-Layered Fuzzy Controller for a Mobile Robot".* 2006 IEEE International Conference on Fuzzy Systems Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 2006

[3] V. Muresan et. al. *"From VHDL to FPGA-A case study of a Fuzzy Logic Controller".* International conference of young lecturers, Hungary,1997

[4] I. Baturone et. al. *"Automatic Design of Fuzzy Controllers for Car Like autonomous Robots".* IEEE Transactions of Fuzzy Systems, 12(4):447-465, 2004

[5] L. Doitsidis et. al. *"Fuzzy Logic Based Autonomous Skid Steering Vehicle Navigation".* Proc. of the 2002 IEEE International Conference on Robotics and Automation Washington DC, 2002

[6] J. Xiao et. al. *"Fuzzy Controller for Wall-Climbing Microrobots"*. IEEE Transactions On Fuzzy Systems, 12(4): 2004

[7] S. Singh et. al. *"Implementation of a Fuzzy Logic Controller on an FPGA using VHDL"*. IEEE, 2003

[8] A. Barriga, R. Senhadji et. al. *"A Design Methodology For Application Specific Fuzzy Integrated Circuits"*. Proc. of the IEEE International Conference on Electronics, Circuits and Systems (ICECS'98), Lisboa, 1998.

[9] C. J. Jimenez et. al. *"Hardware Implementation of a General Purpose Fuzzy Controller "*. Sixth International Fuzzy Systems World Congress, Brazil, IFSA'95, 2:185-188, 1995

[10] D. Galan et. al. *"VHDL Package for Description of Fuzzy Logic Controllers".* European design Automation Conference, Brighton 1995

[11] V. Salapura et. Al. *"Implementing Fuzzy Control Systems Using VHDL and Statecharts"*. In the Proc. Of the European Design Automation Conference EURO-DAC'96 With EURO-VHDL'96, 1996

[12] W. S. Lin, C. L. Huang and M. K. Chuang. *“Hierarchical Fuzzy Control for Autonomous Navigation of Wheeled Robots”. IEE Proceedings - Control Theory and Applications*, 152(5):598-606, 2005

[13] E. Lago, M. A. Hinojosa, C. J. Jiménez, A. Barriga, S. Sánchez-Solano. *“FPGA Implementation Of Fuzzy Controllers”*. XII Conference on Design of Circuits and Integrated Systems (DCIS’97), Sevilla, 1997

[14] A.Y.Deshmukh, Dr.P.R.Bajaj, Dr.A.G.Keskar. *“Hardware Implementation of Fuzzy Logic controllers-Approach and Constraints”*. KES 2006, 10th International Conference on Knowledge Based Intelligent Systems, Bournemouth, UK, 2006

[15] M. McKenna, B. M Wilamowski. *“Implementing a fuzzy system on a Field Programmable Gate array”.* International Joint Conference on Neural Networks, Washington DC, 2001

[16] V. Muresan, D. Crisu, X. Wang. *“From VHDL to FPGA-A case study of a Fuzzy Logic Controller”*. International Conference of Young Lecturers, Hungary, 1997

[17] I. Baturone, S. Sánchez Solano, A. Barriga, J. L. Huertas. *“Optimization Of Adaptive Fuzzy Processor Design”.* XIII Conference on Design of Circuits and Integrated Systems (DCIS’98 Madrid, 1998

[18] A.Y.Deshmukh, Dr.P.R.Bajaj, Dr.A.G.Keskar. *“Autonomous Mobile Systems: Fuzzy Controller Design with VLSI Approach”*. Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference Seattle, WA, USA, 2007

[19] A.Y.Deshmukh, A. Bavaskar, Dr.P.R.Bajaj, Dr.A.G.Keskar. *“Implementation of Complex Fuzzy Logic Modules with VLSI Approach”*. International Journal of Computer Science and Network Security (IJCSNS), 8(9):172-178, 2008