# *Mobile Robot Controller using VHDL by Fuzzy Logic Algorithm*

**Sangharsh S. Jaiswal**
***B.E VI Semester***
***Electronics & Telecommunication Engineering***
***S.B. Jain Institute of Technology Management & Research, Nagpur***
**sangharsh.jaiswal876@gmail.com**

***Abstract*— This research paper presents Fuzzy Logic Algorithm (FLA) for designing of model for an autonomous mobile robot controller (MRC) hardware with navigation concept. The MRC enables with navigation for a formless environment to avoid any encountered obstacles without human interference. The designed hardware architecture of autonomous mobile robot can be easily used in formless environments appropriately to avoid collision with obstacles by turning to the proper angle. Additionally, the FLA concept has proven a creditable intelligent solution in dealing for the certain control problems, when the situation is undefined. In this the designed hardware architecture of MRC will perform with the concept of autonomous mobile robot controller using the theories of Fuzzy algorithm. The architecture of MRC consists of Fuzzifier, Fuzzy Rule Base, Inference mechanism and Defuzzifier. After getting the confidence on MATLAB results, the developing model of MRC will translate into VHDL model for hardware implementation, followed by the synthesis tool, Quartus II from Altera to get synthesized logic gate levels. The synthesized codes will be downloaded into Field Programmable Gate Array (FPGA) board to verify the correctness of the MRC algorithm in behavioural level for VLSI execution.**

***Keywords*—** Mobile Robot Controller, Fuzzy rules, VHDL ,Navigation algorithm, FPGA.

## I. INTRODUCTION

Uncertain environments with incomplete and inaccurate information pretence fundamental difficulties into conventional control systems. In this case, the feasibility of applying Fuzzy logic to enable common sense reasoning in decision-makings to counter such problems. One of the main difficulties is faced by conventional control systems are the failure to operate in a condition within complete and imprecise information [1]. As the difficulty of a situation increases, a traditional mathematical model will be difficult for implementing the process. Fuzzy Logic is a tool for modelling uncertain systems by enabling common sense reasoning in decision-making in the absence of complete and precise information [2-3]. It enables the arrival of a definite conclusion based on input information, which is inexplicit, indefinite, noisy and inaccurate. In this research paper, the controller of an autonomous mobile robot has been designed based on the theories of Fuzzy Logic [4-5]. The wheeled robot is able to navigate by itself in a completely unstructured environment. The FLC collects limited information of the environment through sensors, and decides an appropriate angle to turn while it moves with constant velocity to avoid any objects within its surrounding area. The main focus is to develop the modelling of Fuzzy rule based algorithm for MRC and realize its hardware functionality using FPGA board.

On top of this research paper introduces the FLC algorithm and after this part this papers describes concisely the modelling of mobile robot controller with Navigation Algorithm (NA). VHDL modelling and Synthesis are describes after the MATLAB simulation respectively. Besides conclusion, this research paper ending with FPGA realization and the application by the proposed MRC hardware architecture.

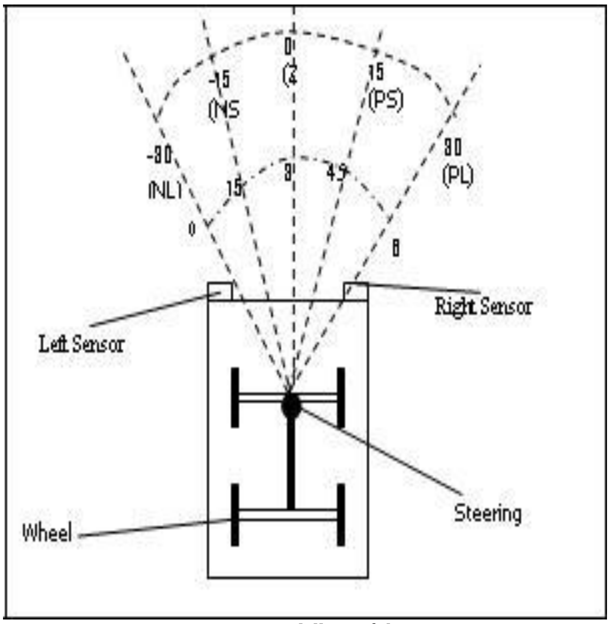


Fig.1 Modelling of the MRC

## II.MODELING OF MRC WITH NA

The MRC monitored by the FLA for navigating through an unstructured environment without human intervention [6-9]. In this process of moving from one place to another, it is necessary for the robot to have information on its current position, where the intake of this vital information is through the two sensors on its body. Fig.1 shows the physical modelling of the MRC.

As an example, the Mobile Robot resembles of a car, in which navigation is made possible by four wheels, controlled by the steering. A significant difference of steering is controlled by the FLC instead of human intelligence. The sensors are positioned in such a way that the robot is able to detect its distance from obstacles, one from its left and another from its right. The steering has a maximum turning angle of 30 degrees. It should be noted that the robot moves in a constant velocity and the only variable controlled by the FLC is the orientation of the robot (the steering angle, in degrees). The FLC receives two inputs pattern, one from the Front Left Sensor, and another from the Front Right Sensor. The inputs are measured in distance. The output is an orientation of the steering, measured in degrees.

### *A. Inputs Pattern*

The inputs from both the sensors measures distance by a numerical value ranging from 0 to 40. An input of “0” denotes the minimum possible distance, which can be detected, where the robot almost or actually touches the obstacle. An input of “40” denotes the maximum distance as

the obstacle is either far away from the robot’s vicinity or could not be sensed at all.

### *B.Output Pattern*

Referring to Fig.1, it shows that the robot is capable of turning an angle of -30 to 30 degrees. However, the output of the FLC, like the inputs is also a numerical value, ranging from 0 to 60. The 0 value represents -30 degrees while “60” represents 30 degrees. In short, the input parameter for both the left and right sensors is the linguistic variable distance, while the output parameter is the linguistic variable angle. They are modelled by the following sets:

Distance1(left sensor) )A = { A1,........,A3} = { *far, near, vnear}*

Distance2(right sensor)≅ B ={B1,........,B3} = { *far, near, vnear* }

Angle *C = {C1…......,C6} = { NL, NS, Z, PS, PL }*

Both input and output membership functions are identical which are shown in Fig.2 & Fig.3 respectively.

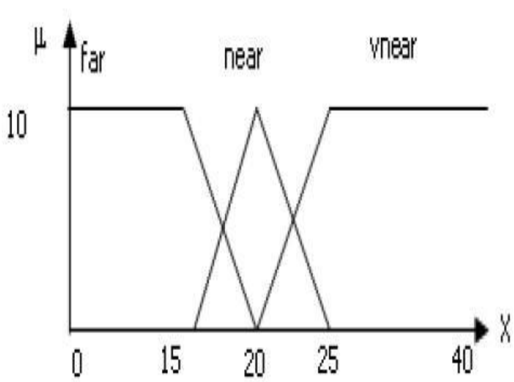


Fig.2 Left & right sensor input Membership function

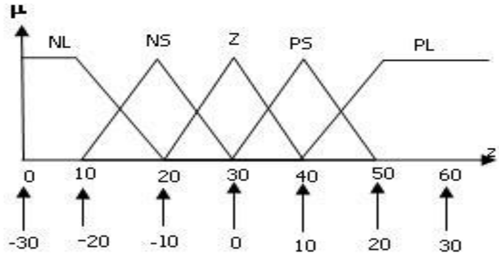


Fig.3 Output membership function

The navigation of the mobile robot is fairly simple. “Structure of the Mobile Robot” and “Fuzzy Logic Controller Architecture” has generally given an idea of the movements of the robot. The sensors are the “eyes” of the robot. They give information on how far the robot is from an object. The FLC is its “brain”. It decides on which angle to turn in order to avoid those obstacles. The FLC takes in a crisp input from each of the sensors. The Fuzzifier categories focus the inputs into Fuzzy sets according to the input membership function. The Rule Base determines the behavior of the robot and decisions are made depending on the rules. The Inference Engine determines rule at a specific situation. The Deffuzifier converts the decision made according to the rule-base, which is in Fuzzy terms, into a crisp output. The output is an orientation of the robot (decides which angle to turn). The algorithmic flow chart of MRC is shown in Fig.4.

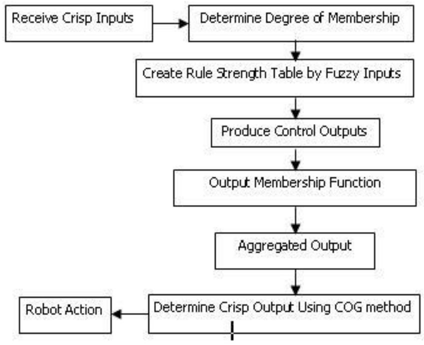


Fig.4 Flowchart

Finally, the FLC gathers information through its sensors, and decides on an action (the output) which is the turning angle of the robot to avoid the obstacles. In fact, after the intake of information (numerical values) through the sensors, fuzzification takes. The values into linguistic variables (far, near, very near), using the input membership functions. Decisions will be made complying to a Rule Base, the rules decides on each situation, to take an action of either giving the robot a “negative large (NL)”, “negative small (NS)”, “zero”, positive small (PS)” or “positive large (PL)” orientation. This is yet not the end of the process because the system requires an exact value as an output, a crisp output. This is achieved by Defuzzification, applied on the output membership function. An autonomous mobile robot is responsible of navigating by itself without human intervention (remote controller etc.) This is achieved using a FLC Algorithm. The computed MRC block diagram using FLC algorithm is shown in Fig.5.

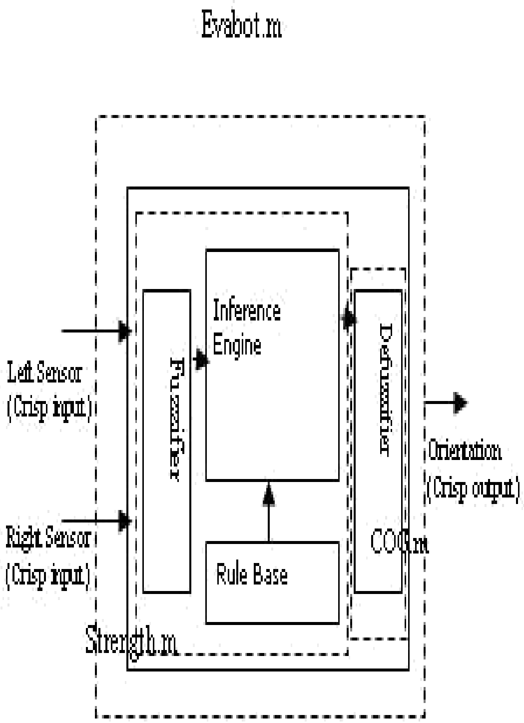


Fig.5 MRC block using FLC algorithm.

## III.MATLAB SIMULATION

The simulation result in MATLAB is partially graphical part, which consist of the graphs that show the shape of the input membership functions (one for the left sensor and another for the right sensor), the graph that shows the shape of the output membership function, and lastly the graph shows the aggregated output (final output). The more important part of the result is the values computed for:

i) Degree of membership
ii) Rule Strengths
iii) Aggregated Output (this is the final output, which is the orientation of the robot in degrees).

To obtain a crisp output using COG method are given below.

AGG(i) = max([min([cut1 Z(i)]) min([cut2 NS(i)]) min([cut3 NL(i)]) min([cut4 PS(i)]) min([cut5 Z(i)]) min([cut6 NS(i)]) min([cut7 PL(i)]) min([cut8 PS(i)]) min([cut9 Z(i)])]);
den = den + AGG(i);
num = num + i*AGG(i);
output = num/den

In the above equation, the control outputs are obtained. Here, “den” denotes the denominator of the COG equation while “num” denotes the numerator. Dividing “num” by “den” produces the final output. For different case of navigating the robot, the controller takes input, such as left sensor (LS) and Right sensor (RS) for different foreseen situation. Finally, the MATLAB simulation result is given in Table 1.

Table 1: Simulation result of robot controller using MATLAB

| Case 1 LS=0 RS=0 | Case 2 LS=16 RS=21 | Case 3 LS=21 RS=16 | Case 4 LS=10 RS=35 | Case 5 LS=35 RS=10 |
|---|---|---|---|---|
| Y=10,0,0 | Y=8,2,0 | Y=0,8,2 | Y=10,0,0 | Y=0,0,10 |
| Z=10,0,0 | Z=0,8,2 | Z=8,2,0 | Z=0,0,10 | Z=10,0,0 |
| S1=0 | S1=0 | S1=0 | S1=0 | S1=0 |
| S2=0 | S2=8 | S2=0 | S2=0 | S2=0 |
| S3=0 | S3=2 | S3=0 | S3=10 | S3=0 |
| S4=0 | S4=0 | S4=9 | S4=0 | S4=0 |
| S5=0 | S5=2 | S5=2 | S5=0 | S5=0 |
| S6=0 | S6=2 | S6=0 | S6=0 | S6=0 |
| S7=0 | S7=0 | S7=2 | S7=0 | S7=0 |
| S8=0 | S8=0 | S8=2 | S8=0 | S8=0 |
| S9=0 | S9=0 | S9=0 | S9=0 | S9=0 |
| num=3100 | num=2860 | Num=5758 | Num=1320 | num=8290 |
| den=100 | den=139 | Den=139 | Den=155 | den=155 |
| ans=31 | ans=20.5755 | Ans=41.4245 | Ans=8.5161 | ans=53.4839 |

## IV. VHDL MODELING

The structures of MRC in MATLAB platform consist of three modules (evabot.m, strength.m and COG.m) are implemented in VHDL[10] as hardware blocks of evabot.vhd, STR3.vhd, and COG.vhd respectively. The modules are combined to form the structure of the MRC block as shown in Fig.6. The two STR3 modules separately receives one external input each and produces the degree of membership as y1,y2,y3 and z1,z2,z3. The “rule strengths” are produces by the evabot itself, as s1, s2, s3, s4, s5, s6, s7, s8 and s9 as internal signals. These are sent to COG for defuzzification and the final output is produced as the orientation of the robot. For each of the cases, it is observed that the “degree of membership” and the “rule strengths” obtained from VHDL simulation is exactly almost same values which are obtained in MATLAB simulation as shown in Table 2. The VHDL simulator is also generate Register Transfer Level (RTL) block of MRC which consists of hardware blocks to verify the MRC behavioural level of algorithm.

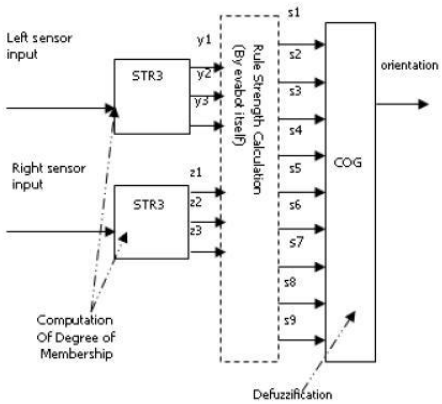


Fig.6 Structure of MRC block in VHDL

Table 2: Comparison Simulation Results in Matlab and VHDL

| Foreseen situation | VHDL | MATLAB |
|---|---|---|
| Case 1 (0,0) | 31 | 31 |
| Case 2 (16,21) | 20 | 20.5755 |
| Case 3 (21,16) | 41 | 41.4245 |
| Case 4 (10,35) | 8 | 8.5161 |
| Case 5 (35,10) | 53 | 53.4839 |

## V.SYNTHESIS

Synthesis is the process of transforming one representation in the design abstraction hierarchy to another representation. Synthesis process has performed using synthesis tool [11] for synthesizing the compiled VHDL design codes into gate level schematics.

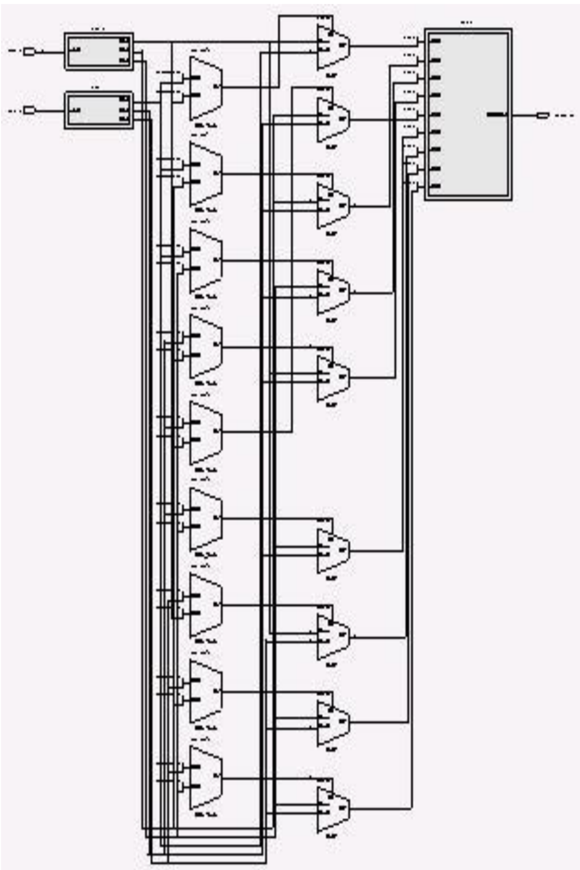

Fig.7 RTL View of MRC architecture.

In order to get the RTL view of MRC architecture, the VHDL codes are synthesized using Quartus-II[12]. The Technology mapping has chosen in this project from Altera (FLEX10K).

The technology view of the various modules for MRC architecture has been carried out. As an example, the top level of RTL views of the MRC is shown in Fig.7. The flattened technology view of Evabot block consisting of different functional logic components of primitive cells are shown in Fig.8. As for example, few particular primitive cells from Evabot block are shown as enlarge image form to

describe its functionality at the behavioural level of the MRC algorithm.

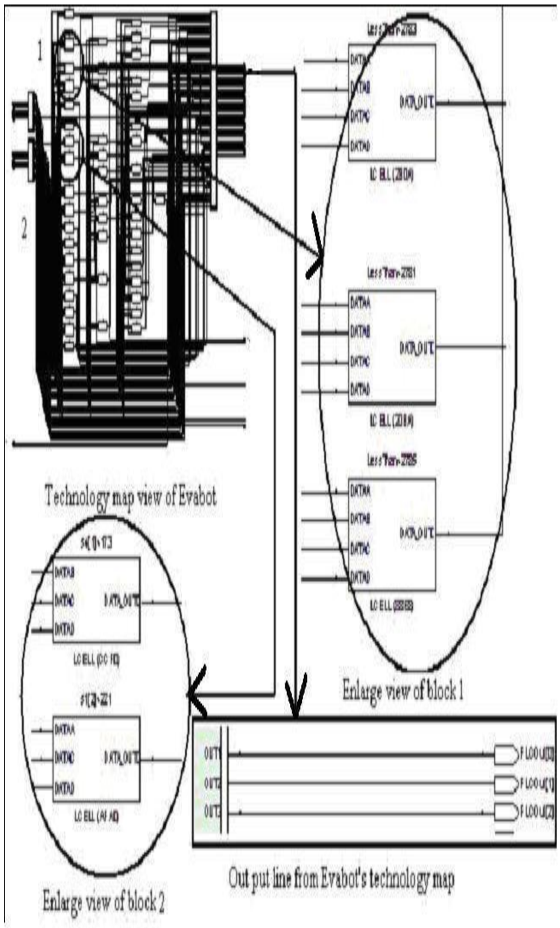


Fig.8 One of the Flattened technology views of MRC architecture (Evabot module)

## VI. FPGA REALIZATION

Next the Timing Analyser shows the delay of data path and give reports the performance of the design in terms of maximum clock speed (fMAX). From the report, it is observed that the longest delay

is 46.13ns from a source register to a destination register in STR3 module. Hence, the maximum clock speed, fMAX is 21.68MHz. Setting the hardware working frequency with 20MHz, the circuit will be confirmed reliability in working condition. This section focuses on the implementation of the MRC algorithm using FLEX10K device using in UP1 Educational Board. During the implementation step, the design codes of the Controller have downloaded into Altera (APEX 20K200EF484) FPGA board [13] as shown in Fig.9.

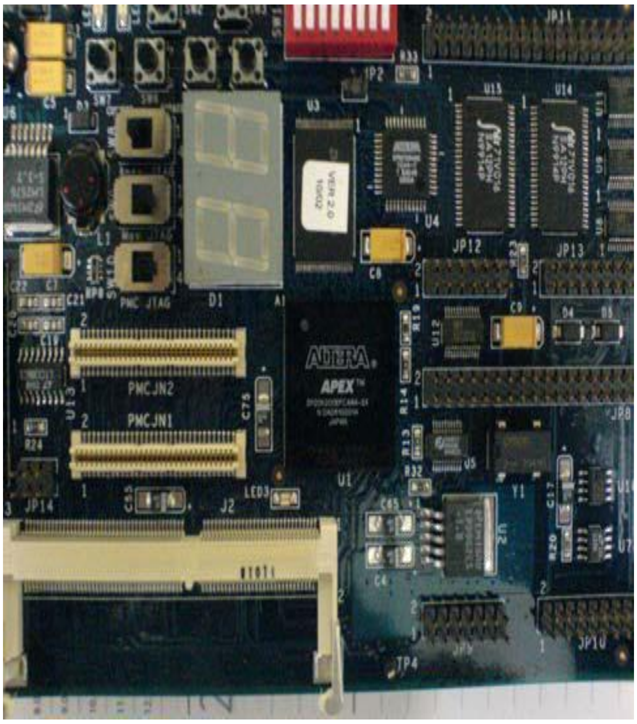

Fig.9 Downloaded into FPGA board using Quartus –II tool

## VII. APPLICATION

Generally mobile robots have the capability to move one place to another place in their environment and are not fixed to one physical location. The most common class of mobile robots is wheeled robots. Other classes of mobile robots are legged robots, aerial robots and autonomous underwater vehicle type robot. All these four types of mobile robots share several unfavourable features such as reaction with unplanned situation, energy autonomy and reaction after getting stuck. Conventional MRC are not capable to improve there behaviours by modifying internal control parameters based on experienced situations. As for example, Neural networks, and Genetic algorithmic approaches also do not offer yet enough guarantee of stability in unsupervised situations for being accepted as a viable method in critical applications where risks and costs rank high.

In this paper, our proposed MRC architecture can be used in any kind of environment by which we can get an improvement of relative performance with respect to the conventional scheme. It shows better performance during unplanned situation. As for example after getting stuck, the MRC architecture by using navigation algorithm can able to find out a way without human intervention. So this kind of MRC can be used widely and efficiently in searching antipersonnel landmines. This MRC architecture can be used on vacuum cleaning of sensitive area where human can not reach and it can be able to show a dramatic performance if used in the vehicle for handicap people.

## VIII.CONCLUSION

The design step of MRC architecture has successfully presented in this paper using navigation concept based on FLA. The constructed MRC has simulated with MATLAB platform. The MRC algorithm in MATLAB has been translated into VHDL model and then synthesized it into logic gate level of MRC architecture for VLSI implementation. Finally the designed synthesized codes have downloaded into FPGA board for verifying the MRC algorithm functionality. The results show that the robot reacts accordingly to the environment. It turns to the appropriate angle to avoid obstacle within its vicinity while navigating the completely unstructured environment.

## IX. REFERENCES


[1] Alessandro Saffioti, Fuzzy Logic in Autonomous Robot Navigation: a case study, *The Handbook of Fuzzy Computations: Oxford University Press*, 1991

[2] Ellen Thro, “Robotics-The Marriage of Computers and Machines”, Facts on Files Inc, 1993.

[3] Francois G. Pin and Yutaka Watanabe, “Autonomous Robotic Systems Group”, Automatic Generation of Fuzzy Rules For the Sensor-Based Navigation of A Mobile Robot, 2002.

[4] Steven D. Kaehler, Fuzzy Logic–An Introduction. March 2006.

[5] George Bojadziev, Maria Bojadziev, “Fuzzy Sets, Fuzzy Logic, Applications,”Vol 5,World Scientific, 1995.

[6] John Lovine, “Robots Androids and Animatrons,” McGraw-Hill, 1998.

[7] Marley Maria B.R. Vellasco, Marco Aurelio and Ivo Lima Brasil, “Mobile Robot Control Using Fuzzy Logic,” In conf. rec 2000 ica conf.

[8] C. Canudas de Wit, B. Siciliano, G. Bastin (Eds.); Theory of Robot Control, Springer-Verlag, London 1996.

[9] R.R. Yager, D.P. Filev: Essentials of Fuzzy Modeling and Control. J. Wiley & Sons, New York 1994.

[10] Max + Plus II, Programmable Logic Development System, Altera Corporation 101 Innovation Drive San Jose, CA 95134, (408) 544- 7000.

[11] User Guide Release 5.0 with HDL Analyst. VHDL and Verilog Synthesis for FPGAs & CPLDs.

[12] User manual of Quartus-II software, Altera Company.