Hardware Implementation of Fuzzy Logic using VHDL

Vikas Kumar Sharma
Supervisor : Prof. Laurent Cabaret and Prof. Celine Hudelot

July 23, 2007
Abstract

In this project, we propose a Fuzzy Logic approach to design the controller for a robot which has to come out of a maze by finding its own path, if it is left anywhere inside the maze. The project involves designing a Fuzzy Logic controller to control its speed and direction of motion and its hardware implementation on FPGA using VHDL. First, the appropriate Fuzzy Logic functions were chosen and they were simulated on VHDL. Then it is implemented on actual robot through FPGA.
## Contents

1. Introduction 4
2. Fuzzy Logic 4
3. Controller Design 4
   3.1 Fuzzification 4
   3.2 Rule Base 5
   3.3 Defuzzification 8
4. Results of Simulation in VHDL: 9
5. Limitations and Suggestions for improvement 10
6. Conclusions and Future Work 10
7. Acknowledgement 11
1 Introduction

There has been a significant increase in the application of Artificial Intelligence (AI) to many practical problems in recent years. Fuzzy Logic has been one of the major tools in the application of AI. This project also seeks to have some help from artificial intelligence and the tool used for this is Fuzzy Logic. The project involved designing a controller for a robot. The robot has to come out of a maze by finding its own path, if it is left anywhere inside the maze. So, it needs to remember the previous path and should follow the next path accordingly.

One of the major steps in designing this controller is to control its speed and direction. So, this project is mainly oriented in controlling speed and direction.

The most commonly used controller is proportional-plus-derivative-plus-integral (PID) controller, which requires a mathematical model of the system. Fuzzy Logic controller provides an alternative to PID controller since it is a good tool for control of systems that are difficult in modeling. The control action in Fuzzy Logic controllers can be expressed with simple “if-then” rules.

2 Fuzzy Logic

It is a mathematical technique for dealing with imprecise data and problems that have many solutions rather than one. Here we have a wide range of speeds for robot and it is difficult to define precisely High, Low and Good speed. Hence we use Fuzzy Logic to deal with this problem. To implement Fuzzy Logic, one has to follow three steps.

1. Fuzzification
2. Rule Base
3. Defuzzification

The implementation of all the three steps in this project have been presented in the next section.

3 Controller Design

3.1 Fuzzification

Our aim was to come to the desired speed from any initial speed. So, we define a parameter

\[ \text{Speed} \ _{\text{diff}} = \text{Desired} \ _{\text{speed}} - \text{Real} \ _{\text{speed}} \]  

(1)

For input of the controller:

\[ \text{Speed} \ _{\text{diff} \ _{\text{in}}} = \text{Desired} \ _{\text{speed}} - \text{Actual} \ _{\text{speed}} \]  

(2)

For output of the controller:

\[ \text{Speed} \ _{\text{diff} \ _{\text{out}}} = \text{Desired} \ _{\text{speed}} - \text{Controlled} \ _{\text{speed}} \]  

(3)

Then we made 3 membership functions for input
• speed_diff_in_neg (if Actual_speed > Desired_speed)
• speed_diff_in_zero (if Actual_speed = Desired_speed)
• speed_diff_in_pos (if Actual_speed < Desired_speed)

And similarly, we made 3 membership functions for output
• speed_diff_out_neg (if Controlled_speed > Desired_speed)
• speed_diff_out_zero (if Controlled_speed = Desired_speed)
• speed_diff_out_pos (if Controlled_speed < Desired_speed)

Figure 1: Membership functions

3.2 Rule Base

We define three rules for this Fuzzy Logic.

1. If speed_diff_in is negative, then speed_diff_out is (less) negative
2. If speed_diff_in is zero, then speed_diff_out is (less) zero
3. If speed_diff_in is positive, then speed_diff_out is (less) positive

To find the max function:

We are given the Desired_speed and Actual_speed, and hence speed_diff_in. First we find the corresponding mf value of negative, zero and positive speed_diff_in function, then we make three functions corresponding to each of negative, zero and positive speed_diff_in having:

- x-axis = x-axis of speed_diff_out.
- y-axis = minimum of (mf value of negative, zero and positive speed_diff_in) and (corresponding speed_diff-out functions according to Rule Base).

Now we have 3 minimum functions
• min1 (corresponding to Rule 1)
• min2 (corresponding to Rule 2)
• min3 (corresponding to Rule 3)

Now, we find max function as:
\[
\text{max function} = \text{maximum of} \{\text{min1, min2, min3}\}
\]

For example, if we take \(\text{speed}_{\text{diff in}} = +32\), the max function will be calculated as following:

Figure 2: min1 function

Figure 3: min2 function
Figure 4: min3 function

Figure 5: max function
### 3.3 Defuzzification

After we have found out the max function, we find the point on x-axis where maximum information has been stored. For this, we use **Gravity Centre method**. According to this method, we find the point which has equal area under max function curve on its both sides. We call this value as speed\_diff\_out. For the above max function, the output of defuzzification is speed\_diff\_out = +4.

![Figure 6: Defuzzification](image)

Now we find the controlled speed as:

\[
\text{controlled speed (out\_reg)} = \text{desired speed (consigne)} \times \text{speed\_diff\_out}.
\]

This controlled speed is fed to the motor after modulated by PWM and again a new speed\_diff\_in is taken as input to the controller and a new speed\_diff\_out is computed again. This process continues till actual speed becomes equal to desired speed (or speed\_diff\_in becomes 0).

![Figure 7: Fuzzy Logic Controller](image)

**Associating the controller to the actual robot:**

- **Speed detector**: It takes in the inputs from Odometer and gives out the actual speed of motor scaling from 0 to 127 cm/s, i.e. 00h to 7Fh.
• Rotation detector: It takes in the inputs from Odometer and gives out the direction of rotation of motor wheel. The output is one bit ‘0’ for the forward direction and ‘1’ for the reverse.

• PWM: It checks the sudden increase or decrease in the output of the controller and hence sudden change in the speed of motor.

• Sens rotation: It takes in the outputs of PWM and desired direction for motor and sends signals to motor to run accordingly.

4 Results of Simulation in VHDL:

Our main purpose is to minimise the Speed_diff (best is to make it equal to 0) in the as minimum time as possible. If we give the Desired_speed = +100 and Actual_speed = +32, then our controller will increase the Controlled_speed of the robot to +82 in one computation, which is feedback to Actual_speed. It shows the proper working of Defuzzification and hence the fuzzy logic.
Eventually in 3 computations, Actual speed reaches the Desired speed. It shows the correctness of fuzzy logic controller. Now, If we change the Desired speed, the computation will start again and Controlled speed (and hence Actual speed) will reach the Desired speed again in a maximum of 5 computations, which will be needed if the separation between Desired speed and Actual speed is maximum i.e. 127.

Figure 10: Final result of computations

Figure 11: Performance of controller if Desired speed is changed

5 Limitations and Suggestions for improvement

The major problem in this project was dealing with signed binary numbers because many softwares (in this project programming software “Xilinx” and testing software “Logic Port”) have different definitions of signed binary numbers. So, different fuzzification functions can be defined which don’t involve signed binary numbers. A different defuzzification method can be used which gives the desired output in lesser time.

6 Conclusions and Future Work:

In this project, we have presented a Fuzzy Logic approach to design the controller to control the speed and direction of robot. The hardware implementation of fuzzy logic controller on FPGA gives the desired result in simulation. The Actual speed of robot comes to the desired speed in atmax 5 computations and each computation takes 515 clock cycles. The number of clock cycles is directly proportional to the variation in the speed levels. With a little modification of signed binary values, the controller is ready to use in the actual robot and then the further work can be done easily.
The further work to be done on this robot is to make appropriate connections between the controller and vision sensors. Then, these results can be applied for the autonomous navigation of the robot so that it can decide its desired speed and directions by itself and store the previous path in its memory. Another thing can be done is to derive the acceleration of the robot from the odometers inputs and applying Fuzzy Logic for the accelerometer.

7 Acknowledgement

I would like to thank Prof. Laurent Cabaret and Prof. Celine Hudelot for their valuable support and fruitful discussions during the project.

References


