Lebanese American University

School of Engineering

Mechatronics System Design

Dr. Jawad Khoury

Project Report

Fuzzy Logic Controlled Ball Position

Guy Abi Hanna
Agapius Bou Ghosn
Ayman El Baarini
Fadi El Kalash

Table of Contents

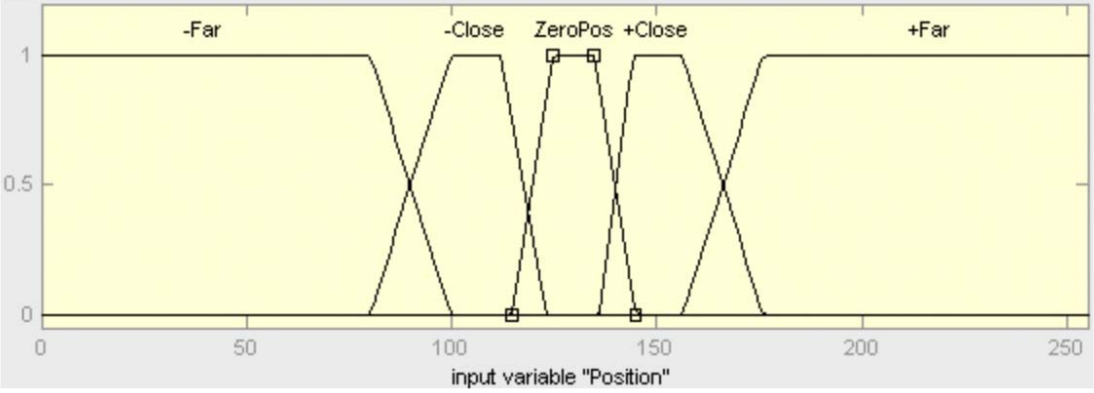
Introduction

This project integrates together multiple new ideas that we have learnt in semester into one. The aim of this project was to control the position of the ball in a tube – about 1 meter long – using a fan. Starting from zero, we had to come up with a fuzzy controller in order to control the fan. The fuzzy controller takes three inputs: the error relative to the desired position, the ball position, and the ball's velocity. Alongside that, a neural network was implemented that would mimic the behavior of our plant.

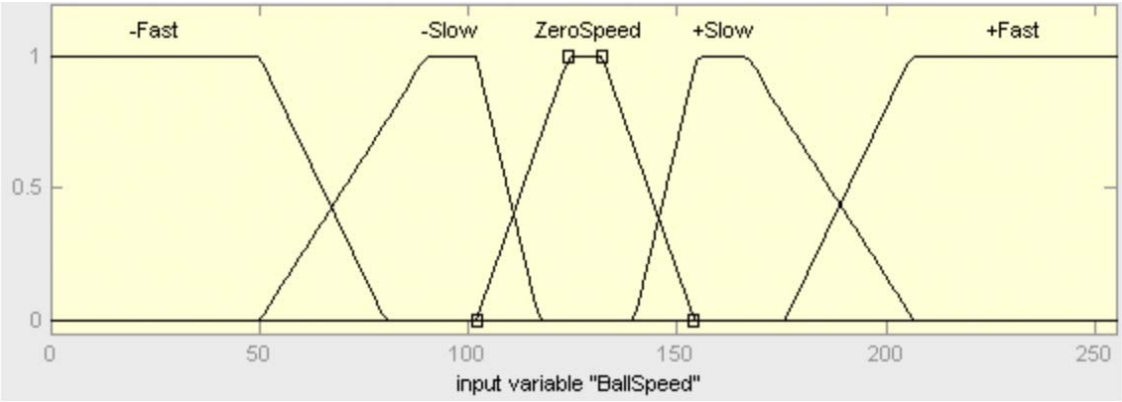
In this report, we will be discussing in depth about the details that had to go into the fuzzy logic controller as well as the neural network. We will also be including some background information about previous similar works as well as some of the problems we faced.

Background Research

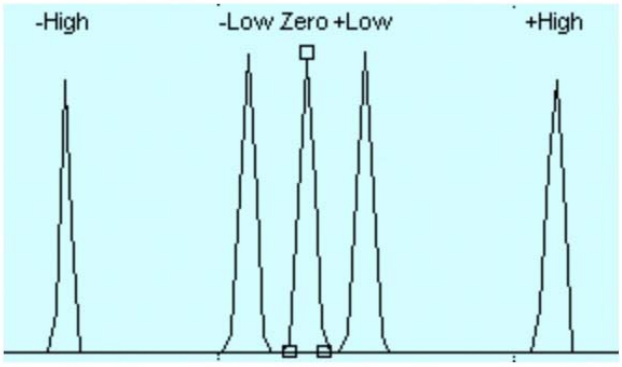
Before beginning the project, a bit of background research was done in order to obtain some hints and tricks that could make this project plausible. A similar project was found where the creator used a fuzzy logic control system to control the position of the ball. He used the ball position relative to the center as an input as well as the speed of the ball. The tube used was 250 cm so his universe of discourse of ball position was [0,250]. His membership functions can be found below.



Similarly, he came up with membership functions for the ball speed.



Finally, his output membership functions were singletons that looked like this:



After looking at these membership functions, he then included them all into one fuzzy associative matrix as follows.

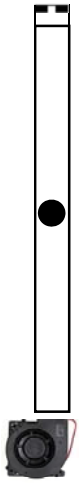
		ball_position				
		neg_far 0	neg_close 1	zero_p 2	pos_close 3	pos_far 4
ball_speed	pos_fast 9	zero_motor 12	neg_low 11	neg_high 10	neg_high 10	neg_high 10
	pos_slow 8	pos_low 13	zero_motor 12	neg_low 11	neg_high 10	neg_high 10
	zero_speed 7	pos_high 14	pos_low 13	zero_motor 12	neg_low 11	neg_high 10
	neg_slow 6	pos_high 14	pos_high 14	pos_low 13	zero_motor 12	neg_low 11
	neg_fast 5	pos_high 14	pos_high 14	pos_high 14	pos_low 13	zero_motor 12

We used this project as an inspiration towards our own project however not much could be replicated into our controller because of the many different factors that caused our project to be different especially the random behavior and the high dependency on the ball's position (which made us later on to add an extra input to the fuzzy logic controller).

Methodology

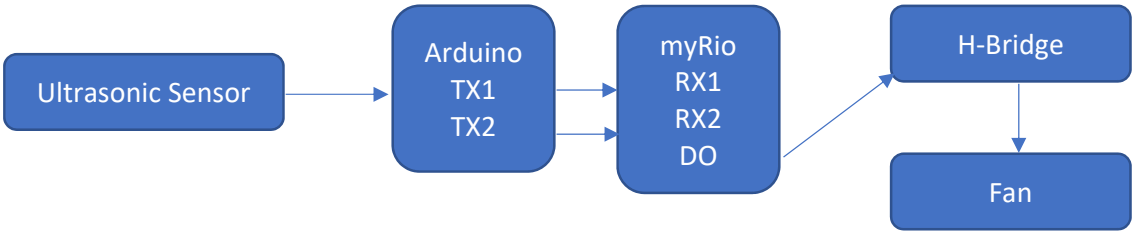
Setup

The setup of the project is basic. A fan is fixed onto a wooden stand at the bottom. A 1-meter long plexi-glass tube sits on the fan. On top of a tube an ultrasonic distance sensor is placed within a plexi-glass holder that is glued 5 centimeters above the top of the tube to be able to measure the exact distance of the ball.



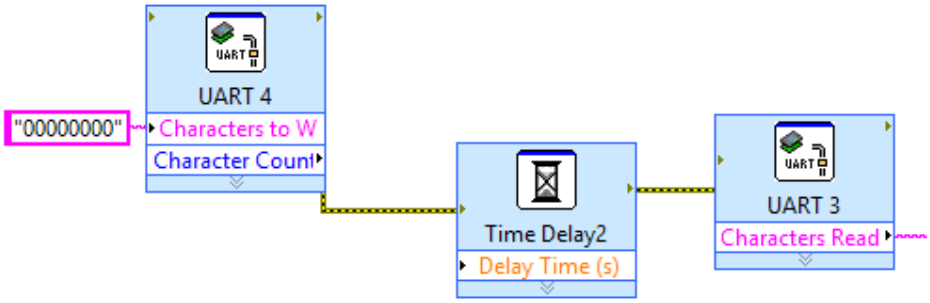
The fan is connected to the myRio which controls it with different PWM signals through the enable port of the H-Bridge. The ultrasonic sensor however is connected to an Arduino Uno since the implementation of its code on the myRio led to an unbalanced behavior of the controller. In order to transfer data from the Arduino to the myRio, a serial communication stream is created by joining the Tx and Rx ports of both controllers. Two serial communication ports were used (for both the distance and the velocity) knowing that the velocity is calculated on Arduino and sent to myRio.

Wiring Diagram:

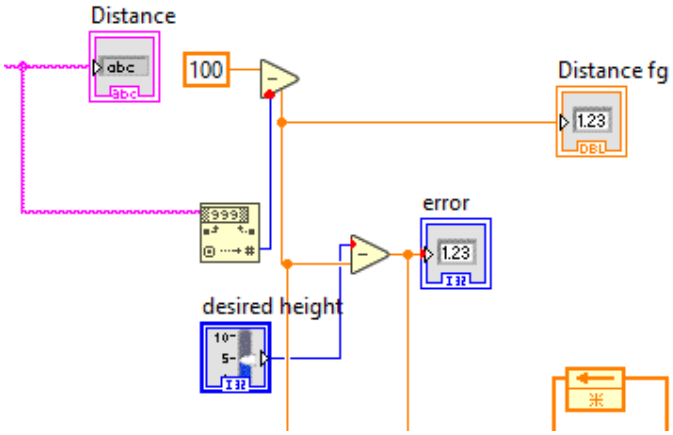


The different sections of the VI (Labview Code) are the following:

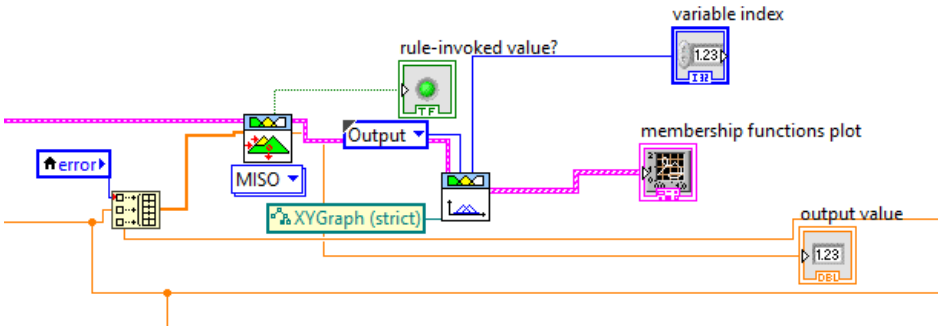
1- The Serial communication with myRio (reading data):



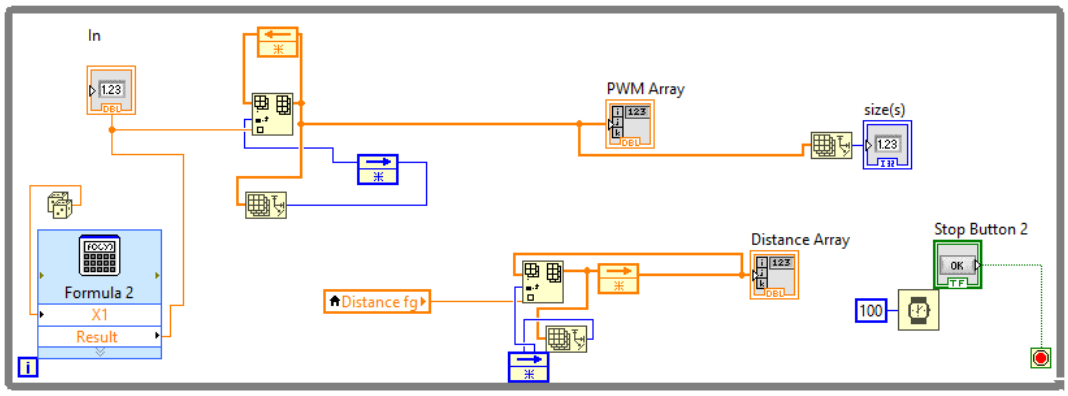
2- Error Calculation:



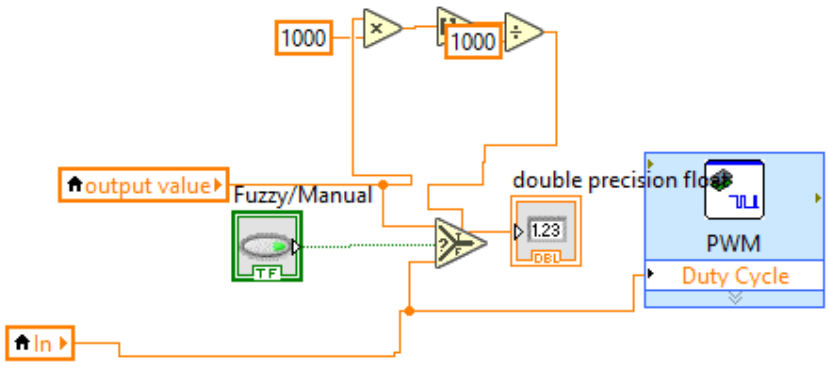
3- The Fuzzy Controller (3 inputs, 1 output)



4- The data points generator:



5- The manual input / fuzzy controller switcher:



The Arduino code is the following (the velocity calculation is done in the Arduino and sent as we will explain later on):

```
const int trigger_pin = 2;    // trigger pin from ultrasonic sensor
const int echo_pin = 3;       // echo pin from ultrasonic sensor

// Define duration, distance, time and speed variables:
long duration1;
long duration2;
int i = 0;
#include <SoftwareSerial.h>
int distance1;
int distance2;
int traveled_distance;
unsigned long time1;
unsigned long time2;
float elapsed_time;
float measured_speed;

SoftwareSerial ser(0,1);
SoftwareSerial ser2(10,11);
```

```
void setup() {
  pinMode(trigger_pin, OUTPUT);
  pinMode(echo_pin, INPUT);


  ser.begin(9600);
  ser.setTimeout(1);
  ser2.begin(9600);
  ser2.setTimeout(1);
  Serial.begin(9600);
  pinMode(5, OUTPUT);

}

void loop() {
  digitalWrite(trigger_pin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger_pin, LOW);
  duration1 = pulseIn(echo_pin, HIGH);
       time1 = millis();
  if(duration1*0.034/2>distance1+1 || duration1*0.034/2<distance1 - 1)
   {

    distance1 = duration1*0.034/2;
   }
   Serial.println(distance1);
  delay(100);
 i++;
 if(i%5==0) {
  time2 = millis();
  elapsed_time = time2-time1;
  traveled_distance = distance2-distance1;
  distance2 = distance1;
  measured_speed = traveled_distance/elapsed_time*36;
  ser2.println(measured_speed);
 }

}
```

Fuzzy Logic Controller

The first part of the project was focused mainly on implementing a fuzzy controller which can sustain the height of the ball at a certain position with a small fluctuation before the plant returning to its random behavior. In order to do so we first had to find the range of the PWM signal that we could work with. Upon undergoing a few tests, we found that the take-off voltage needed was at a duty cycle of 66.9%. The maximum PWM that we need in order for the ball to reach the top of the tube was 69%. Therefore, the output membership functions will have a universe of discourse of [0.649,0.701]. For the inputs, the universe of discourse of the error is [-100,100] representing the difference in distance between current position and desired position in centimeters, the universe of discourse for the current ball position was [0 100] divided into three regions (low, medium and high). Finally, after testing, we came up of a universe of discourse of [-10,10] for the velocity (the velocity was converted to km/h).

The next step was to come up with the membership functions for each variable. For the output, the universe of discourse was split up into 9 membership functions of triangular shapes. These functions are called in the position's increasing order: LL, LM, LH, ML, MM, MH, HL, HM, HH. As you see the functions are split evenly according to the position of the ball in the tube. For example, LL-LM-LH will be triggered when the ball is in the bottom third of the tube; similarly, for the other two parts of the tube.
The three values (low, medium and high) for all the parts of the tube were taken as follows: the low value will let the ball fall down smoothly, the medium value will try to maintain the ball at the same position, the high value let the ball move up.
The fuzzy controller is summarized in the next few pages:

Input variables

Name	Range	Number of membership functions
Error	-100 -> 100	5
Velocity	-10 -> 10	5
Position	0 -> 100	3

Output variables

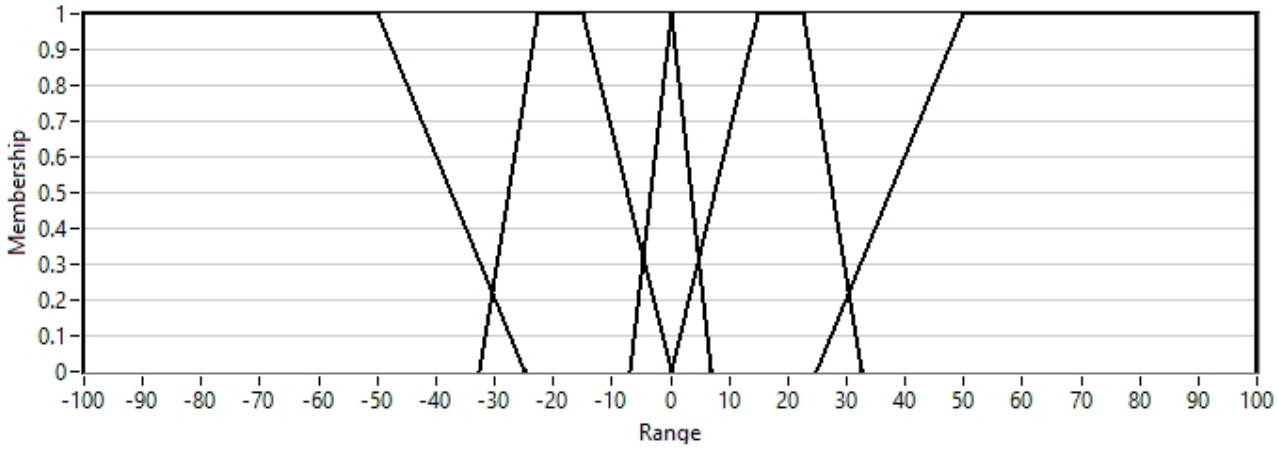
Name	Range	Number of membership functions
PWM	0.648 -> 0.701	9

Defuzzification method: Center of Area

Input membership functions

Error

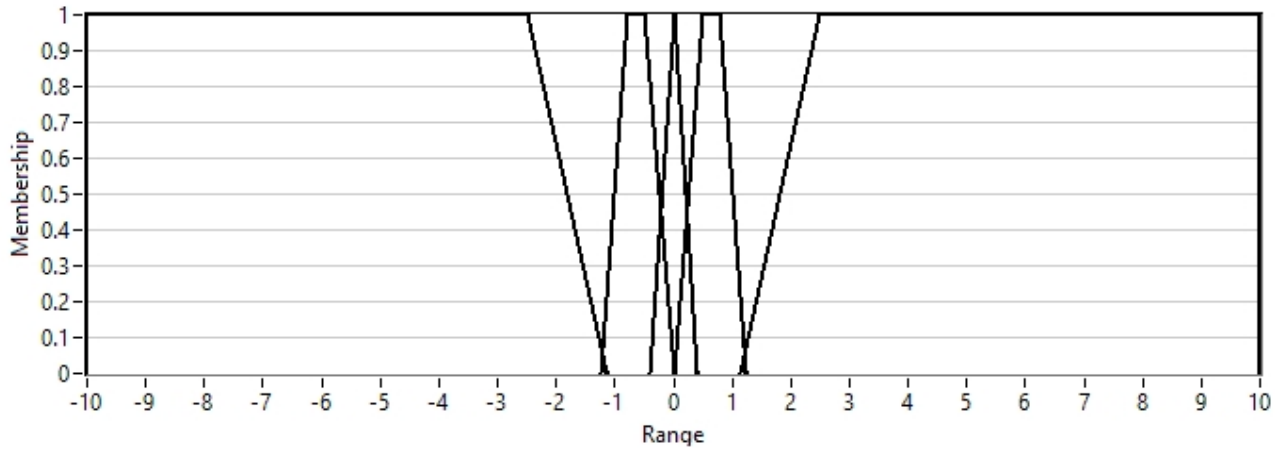
Membership function	Shape	Points
Negative	Trapezoid	-100 ; -100 ; -50 ; -25
SNegative	Trapezoid	-32.5 ; -22.5 ; -15 ; 0
Zero	Triangle	-7 ; 0 ; 7
SPositive	Trapezoid	0 ; 15 ; 22.5 ; 32.5
Positive	Trapezoid	25 ; 50 ; 100 ; 100



Velocity

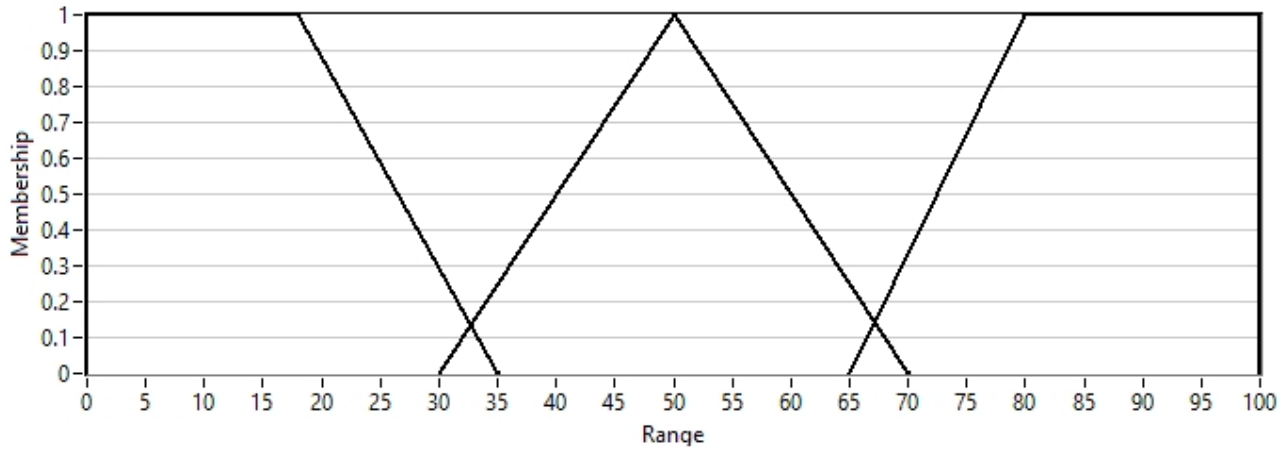
Membership function	Shape	Points
NFast	Trapezoid	-10 ; -10 ; -2.5 ; -1.15

NSlow	Trapezoid	-1.25 ; -0.8 ; -0.5 ; -0
Zero	Triangle	-0.4 ; 0 ; 0.4
PSlow	Trapezoid	0 ; 0.5 ; 0.8 ; 1.25
PFast	Trapezoid	1.15 ; 2.5 ; 10 ; 10



Position

Membership function	**Shape**	**Points**
Low	Trapezoid	0 ; 0 ; 18 ; 35
Medium	Triangle	30 ; 50 ; 70
High	Trapezoid	65 ; 80 ; 100 ; 100

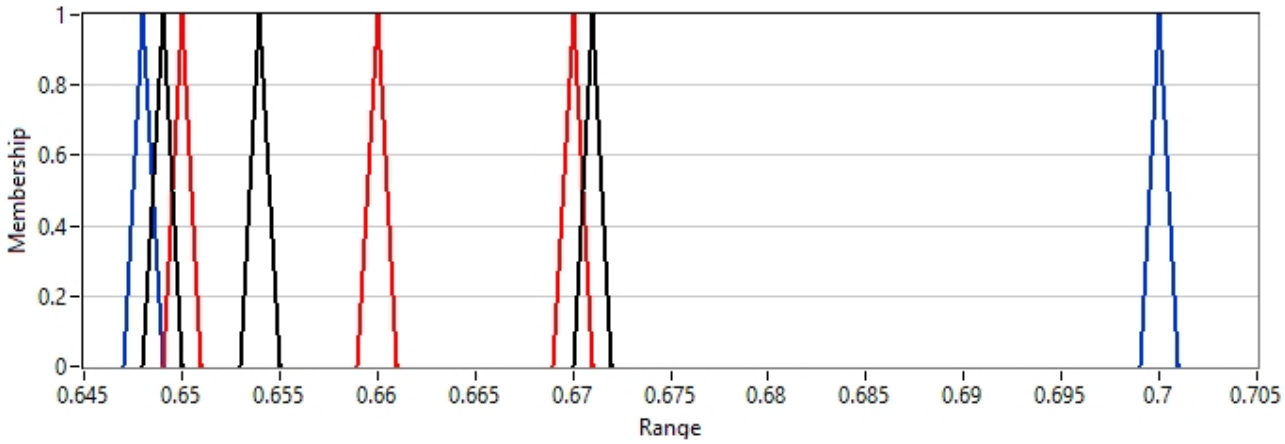


Output membership functions

PWM

Membership function	**Shape**	**Points**
LL	Triangle	0.648 ; 0.649 ; 0.65

LM	Triangle	0.653 ; 0.654 ; 0.655
LH	Triangle	0.67 ; 0.671 ; 0.672
ML	Triangle	0.649 ; 0.65 ; 0.651
MM	Triangle	0.659 ; 0.66 ; 0.661
MH	Triangle	0.669 ; 0.67 ; 0.671
HL	Triangle	0.647 ; 0.648 ; 0.649
HM	Triangle	0.669 ; 0.67 ; 0.671
HH	Triangle	0.699 ; 0.7 ; 0.701



Rules

1. IF 'Error' IS 'Negative' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

2. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

3. IF 'Error' IS 'Zero' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

4. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

5. IF 'Error' IS 'Positive' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

6. IF 'Error' IS 'Negative' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

7. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

8. IF 'Error' IS 'Zero' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'

connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

9. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

10. IF 'Error' IS 'Positive' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

11. IF 'Error' IS 'Negative' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Low' THEN 'PWM' IS 'LL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

12. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

13. IF 'Error' IS 'Zero' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

14. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

15. IF 'Error' IS 'Positive' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Low' THEN 'PWM' IS 'LH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

16. IF 'Error' IS 'Negative' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

17. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

18. IF 'Error' IS 'Zero' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

19. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

20. IF 'Error' IS 'Positive' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

21. IF 'Error' IS 'Negative' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

22. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

23. IF 'Error' IS 'Zero' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

24. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

25. IF 'Error' IS 'Positive' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Low' THEN 'PWM' IS 'LM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

26. IF 'Error' IS 'Negative' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

27. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

28. IF 'Error' IS 'Zero' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

29. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

30. IF 'Error' IS 'Positive' AND 'Velocity' IS 'NFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

31. IF 'Error' IS 'Negative' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

32. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

33. IF 'Error' IS 'Zero' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

34. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

35. IF 'Error' IS 'Positive' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

36. IF 'Error' IS 'Negative' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

37. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

38. IF 'Error' IS 'Zero' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

39. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

40. IF 'Error' IS 'Positive' AND 'Velocity' IS 'Zero' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

41. IF 'Error' IS 'Negative' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

42. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

43. IF 'Error' IS 'Zero' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

44. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

45. IF 'Error' IS 'Positive' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

46. IF 'Error' IS 'Negative' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

47. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

48. IF 'Error' IS 'Zero' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

49. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'ML'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

50. IF 'Error' IS 'Positive' AND 'Velocity' IS 'PFast' AND 'Position' IS 'Medium' THEN 'PWM' IS 'MM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

51. IF 'Error' IS 'Negative' AND 'Velocity' IS 'NFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

52. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'NFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

53. IF 'Error' IS 'Zero' AND 'Velocity' IS 'NFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

54. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'NFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

55. IF 'Error' IS 'Positive' AND 'Velocity' IS 'NFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

56. IF 'Error' IS 'Negative' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

57. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

58. IF 'Error' IS 'Zero' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

59. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

60. IF 'Error' IS 'Positive' AND 'Velocity' IS 'NSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

61. IF 'Error' IS 'Negative' AND 'Velocity' IS 'Zero' AND 'Position' IS 'High' THEN 'PWM' IS 'HL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

62. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'Zero' AND 'Position' IS 'High' THEN 'PWM' IS 'HL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

63. IF 'Error' IS 'Zero' AND 'Velocity' IS 'Zero' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

64. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'Zero' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

65. IF 'Error' IS 'Positive' AND 'Velocity' IS 'Zero' AND 'Position' IS 'High' THEN 'PWM' IS 'HH'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

66. IF 'Error' IS 'Negative' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

67. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

68. IF 'Error' IS 'Zero' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

69. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

70. IF 'Error' IS 'Positive' AND 'Velocity' IS 'PSlow' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

71. IF 'Error' IS 'Negative' AND 'Velocity' IS 'PFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

72. IF 'Error' IS 'SNegative' AND 'Velocity' IS 'PFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

73. IF 'Error' IS 'Zero' AND 'Velocity' IS 'PFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HM'

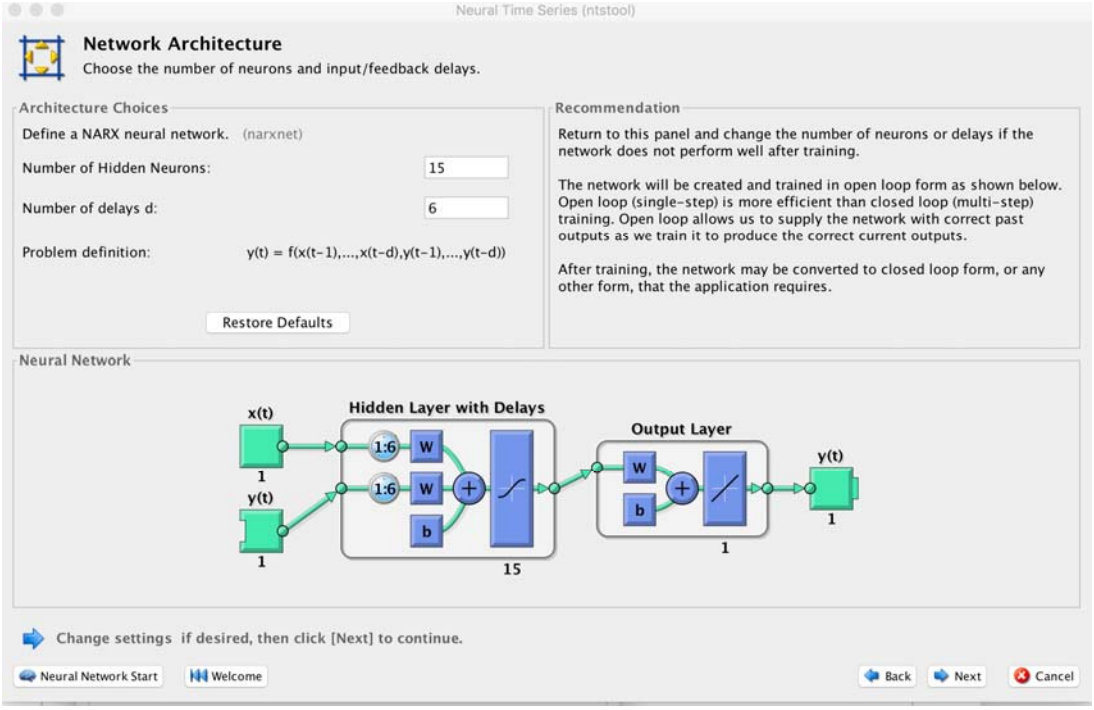
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

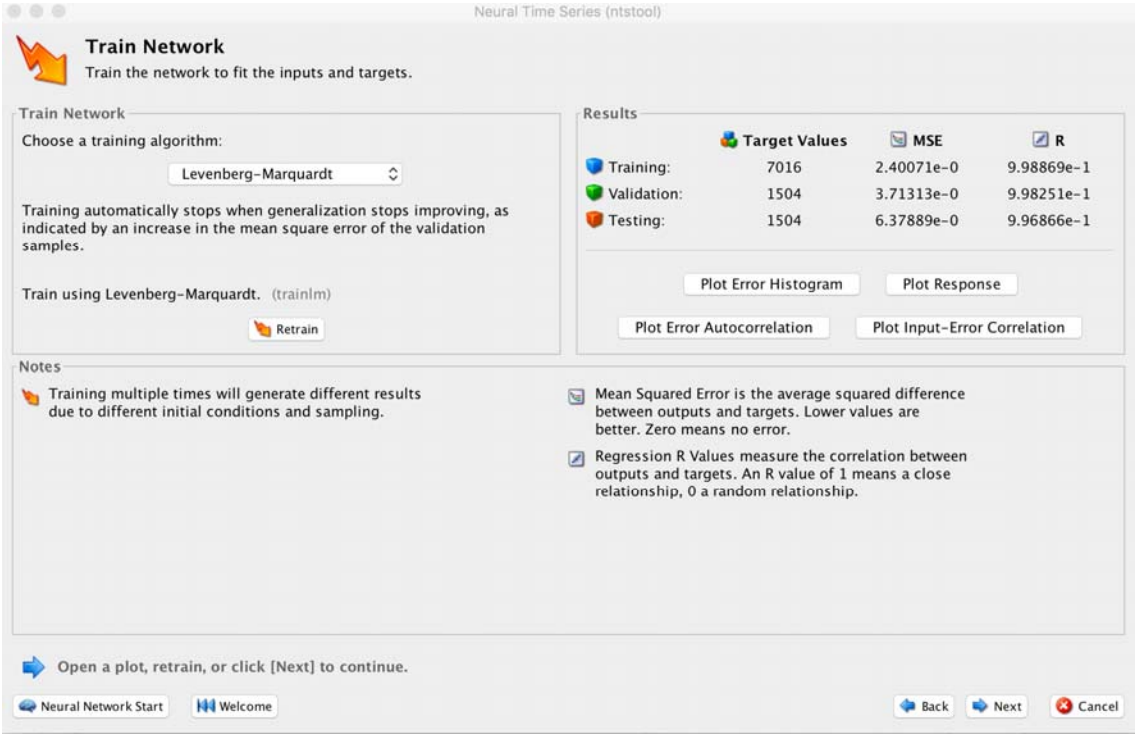
74. IF 'Error' IS 'SPositive' AND 'Velocity' IS 'PFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

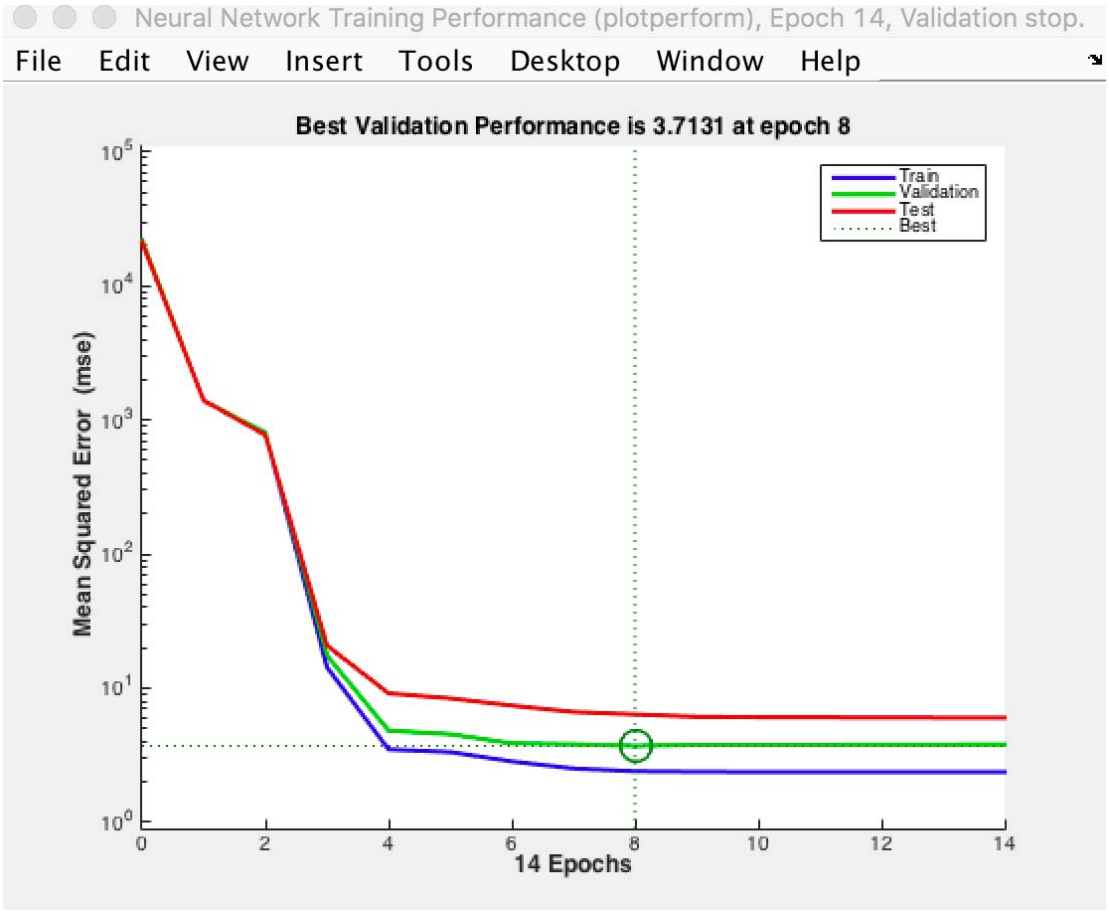
75. IF 'Error' IS 'Positive' AND 'Velocity' IS 'PFast' AND 'Position' IS 'High' THEN 'PWM' IS 'HL'
connective: AND (Minimum) ; implication: Minimum ; degree of support: 1.00

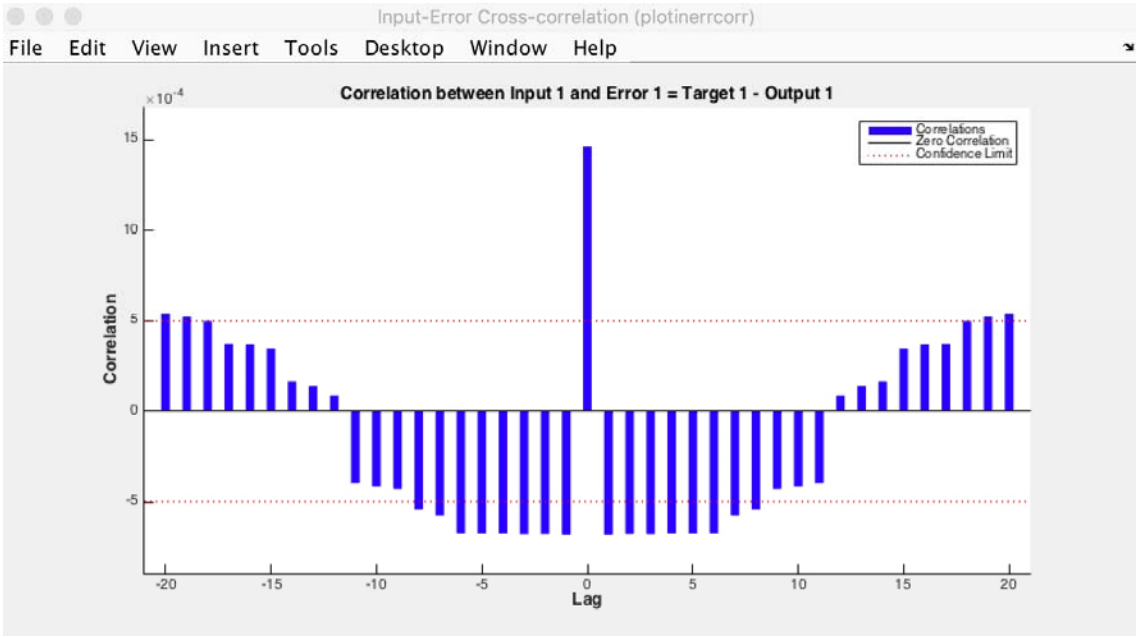
Neural Networks

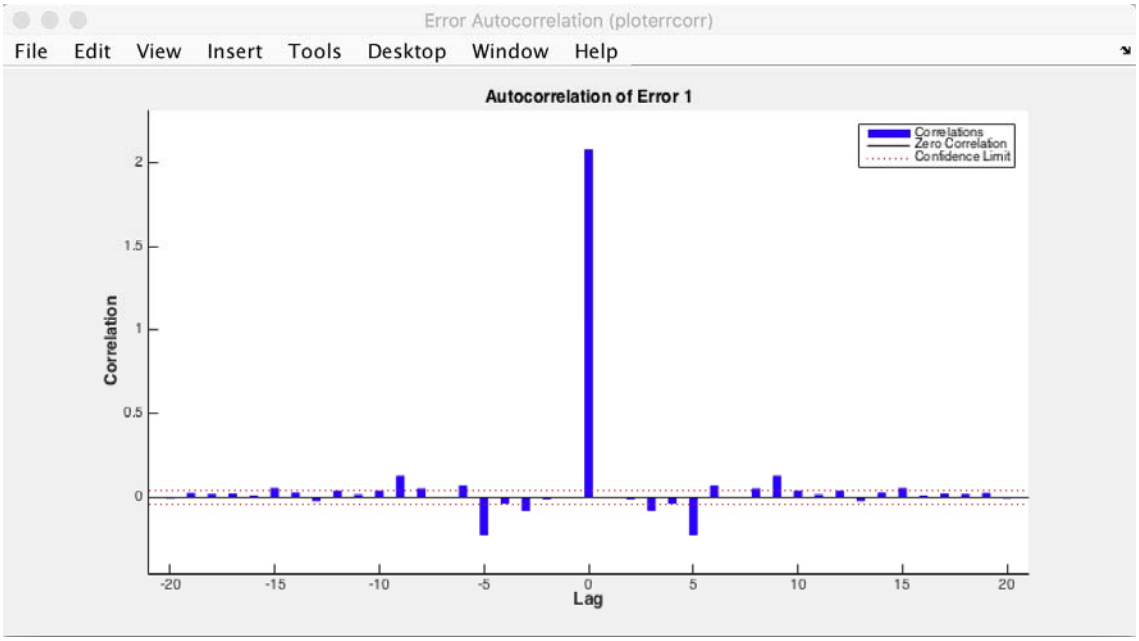
Our model did not have a mathematical equation that governs the behavior of its output (distance) in function of its input (fan duty cycle). So, we collected a set of empirical data (10000) points and use it to train our NARX architecture-based neural network with 15 neurons and 6 tap delays. The errors were almost close (2-3-6) and those errors are small relatively speaking to the output which is distance of maximum 100cm. The input error cross correlation and the error autocorrelation are within the margins of the confidence. So, our neural network model of the plant is reliable, and we tested it using a series of pwm (duty cycle) inputs, the distance increased to 90 then it stabilized at 75. This behavior was exactly noticed during the training.

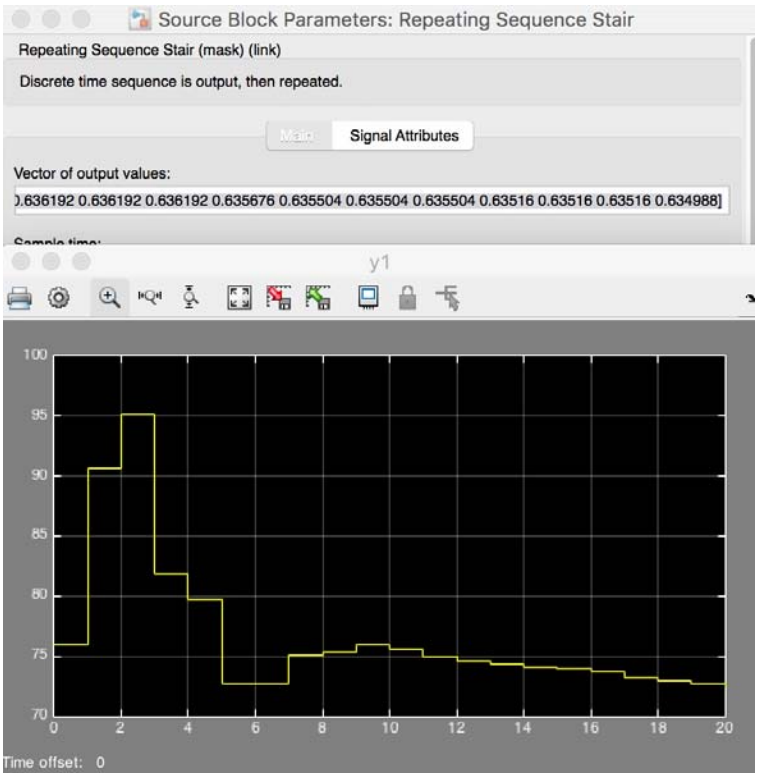


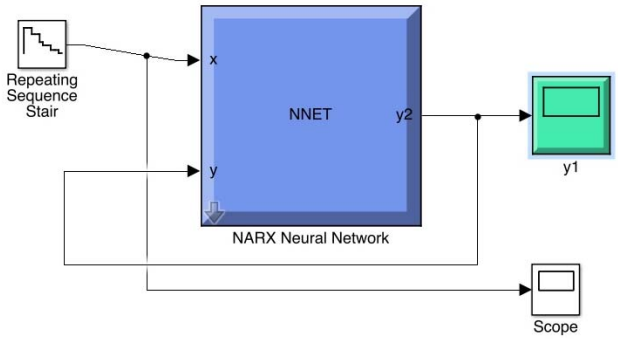

Neural Time Series (ntstool)
Train Network
Train the network to fit the inputs and targets.
Train Network
Choose a training algorithm:
Levenberg-Marquardt
Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.
Train using Levenberg-Marquardt. (trainlm)
Retrain
Results
Target Values
MSE
R
Training: 7016 2.40071e-0 9.98869e-1
Validation: 1504 3.71313e-0 9.98251e-1
Testing: 1504 6.37889e-0 9.96866e-1
Plot Error Histogram
Plot Response
Plot Error Autocorrelation
Plot Input-Error Correlation
Notes
Training multiple times will generate different results due to different initial conditions and sampling.
Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.
Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.
Open a plot, retrain, or click [Next] to continue.
Neural Network Start
Welcome
Back
Next
Cancel


Neural Network Training Performance (plotperform), Epoch 14, Validation stop.
File Edit View Insert Tools Desktop Window Help
Best Validation Performance is 3.7131 at epoch 8
Train
Validation
Test
Best
Mean Squared Error (mse)
14 Epochs


Input-Error Cross-correlation (plotinerrcorr)
File Edit View Insert Tools Desktop Window Help
Correlation between Input 1 and Error 1 = Target 1 - Output 1
×10^-4
Correlations
Zero Correlation
Confidence Limit
Correlation
Lag


Error Autocorrelation (ploterrcorr)
File Edit View Insert Tools Desktop Window Help
Autocorrelation of Error 1
Correlations
Zero Correlation
Confidence Limit
Correlation
Lag





Part List

Part	Quantity
Arduino Uno	1
NI myRio	1
Plexi Tube (6cm diameter)	1
Mini Breadboard	1
Centrifugal Fan	1
Ultrasonic Sensor	1
3D printed Ball	1
Wires	A lot

Problems Faced

Many problems were dealt with while accomplishing this project such as:

1. Problem: Failure to obtain a correct reading with an IR distance sensor after using multiple transfer functions.
   Solution: Using an Ultrasonic distance sensor and adding an Arduino for ease of use.
2. Problem: Obtaining false reading with the ultrasonic distance sensor.
   Solution: Using a wider tube and lifting the position of the sensor 5cm over the tube.
3. Problem: Instability of the project as a whole.
   Solution: Fixing the fan to a wooden base.
4. Problem: Incorrect calculation of the velocity
   Solution: Use the processors clock as a measure of time between two position readings.
5. Problem: Fluctuation of the position of the ball by a certain error and random behavior from time to time which causes retuning the PWM's duty cycle values
   Solution: Solved by removing the nozzle and giving the fan some space to leak air.

Conclusion

The myRio and Arduino work in tandem in order to implement a closed loop control of the ball position. The membership functions, fuzzy system, and fuzzy associative matrix were changed multiple times until we were able to reach an acceptable output. This project really showed how fragile this fuzzy control system really is where a slight change in any of the components or parameters of the system makes it useless. Some sources of errors are not yet discovered.

We will be providing you by email the labview files, and the fis files.

References

https://www.markbowers.org/levitating-ball/