Contents

Table of Figures

1-Introduction

1.1 Overview

Fuzzy logic is widely used in machine control. When logic is referred to as "fuzzy," it means that it can deal with concepts that cannot be labelled as "true" or "false." They can only be described as "partially true." Even though genetic algorithms and neural networks often outperform fuzzy logic, the solution can be expressed in terms that human operators can understand. It allows their expertise to be incorporated into the controller's design. It makes it simpler to automate tasks that people are already good at.

1.2 Fuzzy control

In fuzzy logic, analogue input values are evaluated using logical variables with values ranging from 0 to 1. It contrasts with the classical or digital sensor, which employs discrete values. As a result, fuzzy control systems can now be created (true or false, respectively).

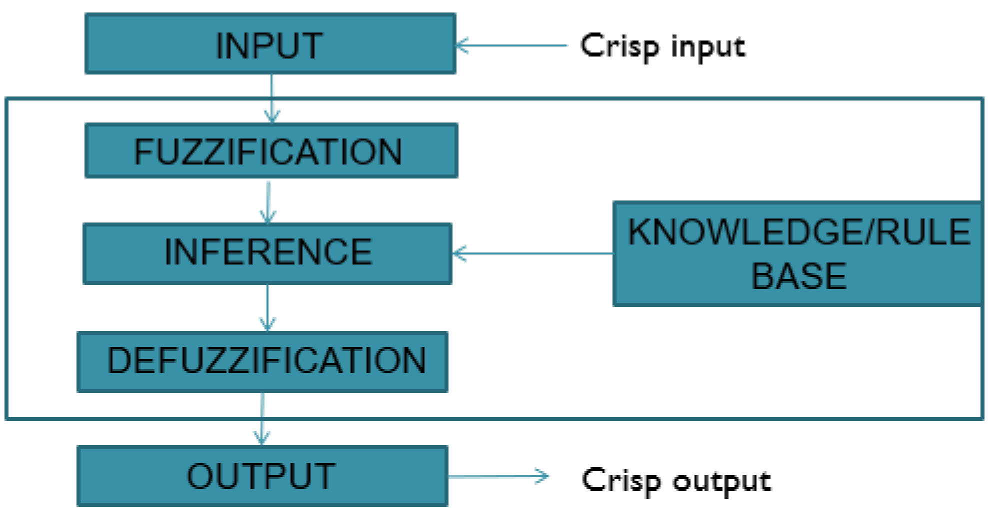

Figure 1 Fuzzy control architecture

"If the temperature is low, open the heating valve slightly," for example, is an example of an if-then rule that a fuzzy logic controller can use to describe a control protocol. The overlapping edges of fuzzy sets indicate ambiguity (uncertainty) in the definition of a language term. A domain element can be a member of more than one set simultaneously under the fuzzy set paradigm (with varying degrees of membership). As a result, t=20°C belongs to both the High temperatures (0.4

members) and the medium temperatures (0.6 members) sets (0.2 members). Because the transition from member to a non-member is gradual, fuzzy if-then rules can be used to reach a satisfactory conclusion.

1.3 Fuzzification

Dividing a system's input or output into multiple fuzzy sets is called "fuzzification." Because triangles and trapezoids are more straightforward to display in embedded controllers than other curves and tables, they are used in most membership functions.

But there are also crucial points to be remembered:

- The functions of membership should overlap so that the system map works appropriately.
- For the system map to work as planned, functions related to membership should be linked together.

Trapezoidal and triangular membership functions describe input in the fuzzy set system depicted in Figure 2. For each fuzzy set, a region of input (or output) values is plotted against membership. When given input, this fuzzy set is used to determine how much of it belongs to the set. The functions of membership must overlap for a system mapping to work correctly. Fuzzification simplifies applying rules to complex systems by allowing the system's inputs and outputs to be expressed in natural language.

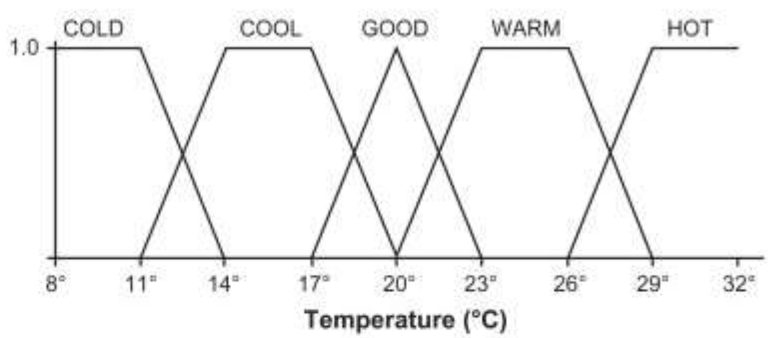

Figure 2 Set of scale by fuzzy set

1.4 Membership function

Simply put, fuzzy logic is a type of logic used to describe uncertainty, not a difficult-to-define logic. The membership function best describes how hazy this is [1]. In other words, the membership function in fuzzy logic is a way to show how trustworthy something is.

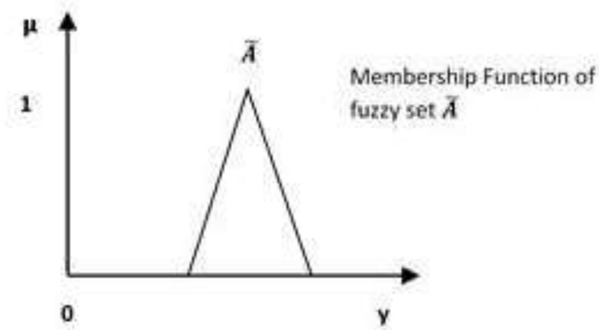

Figure 3 Membership function

There are also important points of the membership functions

- The definition of fuzziness is all the information in a fuzzy set, whether the elements are discrete or continuous.
- A membership function is a method of dealing with real-world problems based on personal experience rather than theoretical knowledge.
- Membership functions are illustrated graphically.

1.4.1 Core

The part of spacetime fully included in any fuzzy set A is essential to a membership function [2]. All of these are the fundamental building blocks of the information universe.

$$\mu \tilde{A}(y) > 0$$

1.4.2 Boundary condition

A membership function’s limit is the part of the universe defined by a nonzero but incomplete membership in any fuzzy set A. As a result, everything in the information universe is a part of the center.

$$1 > \mu \tilde{A}(y) > 0$$

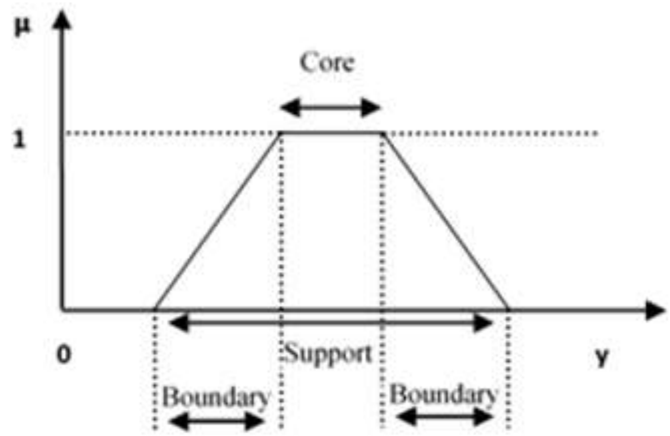

Figure 4 Features of the MF

1.4.3 Types of Membership functions

1.4.3.1 Singleton function

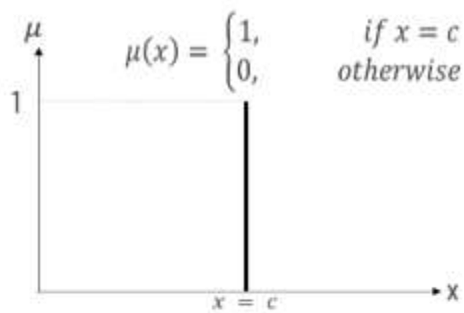

Figure 5 Singleton function

1.4.3.2 Triangular function

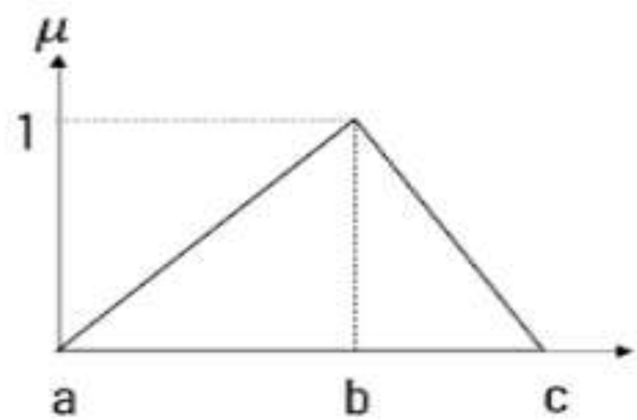

Figure 6 Triangular function

1.4.3.3 Trapezoidal function

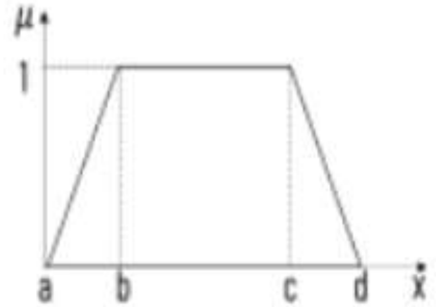

Figure 7 Trapezoidal function

1.4.3.4 Gaussian function

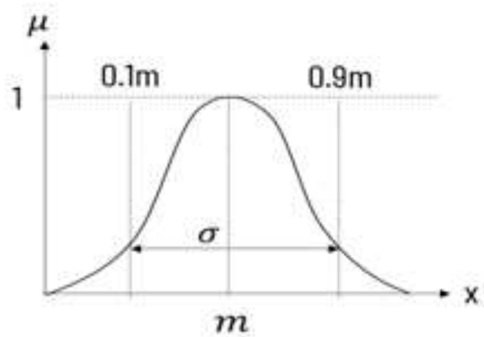

Figure 8 Gaussian function

1.5 Fuzzy Interference system

There are two fuzzy interfaces

- Mamdani Interference system,
- Sugeno Interference system

1.5.1 Mamdani Interference system

- The output membership function is present [2]
- The output of the surface is discontinuous
- Distribution of output
- Through defuzzification of rules consequent of crisp result is obtained

1.5.2 Sugeno Interference system

- No output membership function is present
- The output of the surface is continuous
- No distribution of output, only a Mathematical combination of the output and the strength of the rules
- No defuzzification here. Using a weighted average of the rules of consequent crisp result is obtained.

1.6 Defuzzification

Defuzzification, on the other hand, uses mapping to convert fuzzy results into clear ones. A decision-making algorithm selects the best clear value from a fuzzy set to defuzzify.

There are methods of defuzzification

- Center of gravity
- Mean of maximum
- Center average methods

1.6.1 Center of gravity

The Center of gravity (COG) method works by locating the point X at which a vertical line would divide the aggregate into two equal masses.

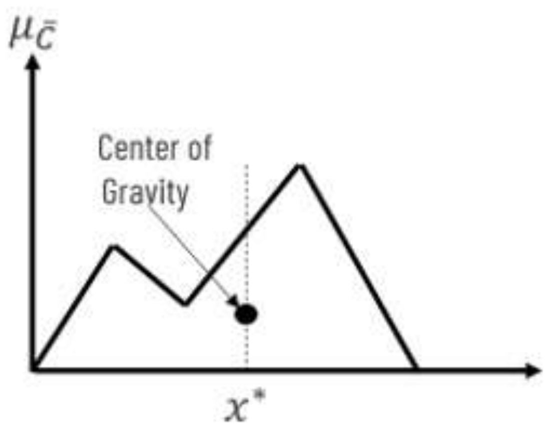

Figure 9 COG method

1.6.2 Mean of Maximum

This strategy is also known as “the middle of the maxima.” In math notation, the defuzzied result x is written as:

$$\mathbf{x}^{*} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} \overline{\mathbf{x}_{\mathbf{i}}}}{\mathbf{n}}$$

1.6.3 Center average method

This method can only be used with output membership functions that are the same on both sides. The weight of each membership function is set by the maximum membership value of that function. The output of the case is given by the equation $\boldsymbol{z} * = \ (\boldsymbol{z}')$.

$$\mathbf{z}^{*} = \frac{\sum \boldsymbol{\mu}(\mathbf{z}') \times \mathbf{z}'}{\sum \boldsymbol{\mu}(\mathbf{z}')}$$

2-Working

2.1 Fuzzy logic toolbox

Fuzzy Logic Toolbox includes MATLAB® functions, Simulink® apps, and a Simulink® block for analyzing, designing, and simulating fuzzy logic systems. This product allows you to select and configure the inputs, outputs, membership functions, and rule sets for type-1 and type-2 fuzzy inference systems [4].

A fuzzy inference system's membership functions and rules can be automatically tuned using the toolbox. The fuzzy logic systems that have been built can be tested and analyzed using MATLAB and Simulink. In addition, the fuzzy inference system can be used to explain AI models that contain black boxes (AI). Fuzzy logic systems can be tested and used by developing executables and C/C++ code.

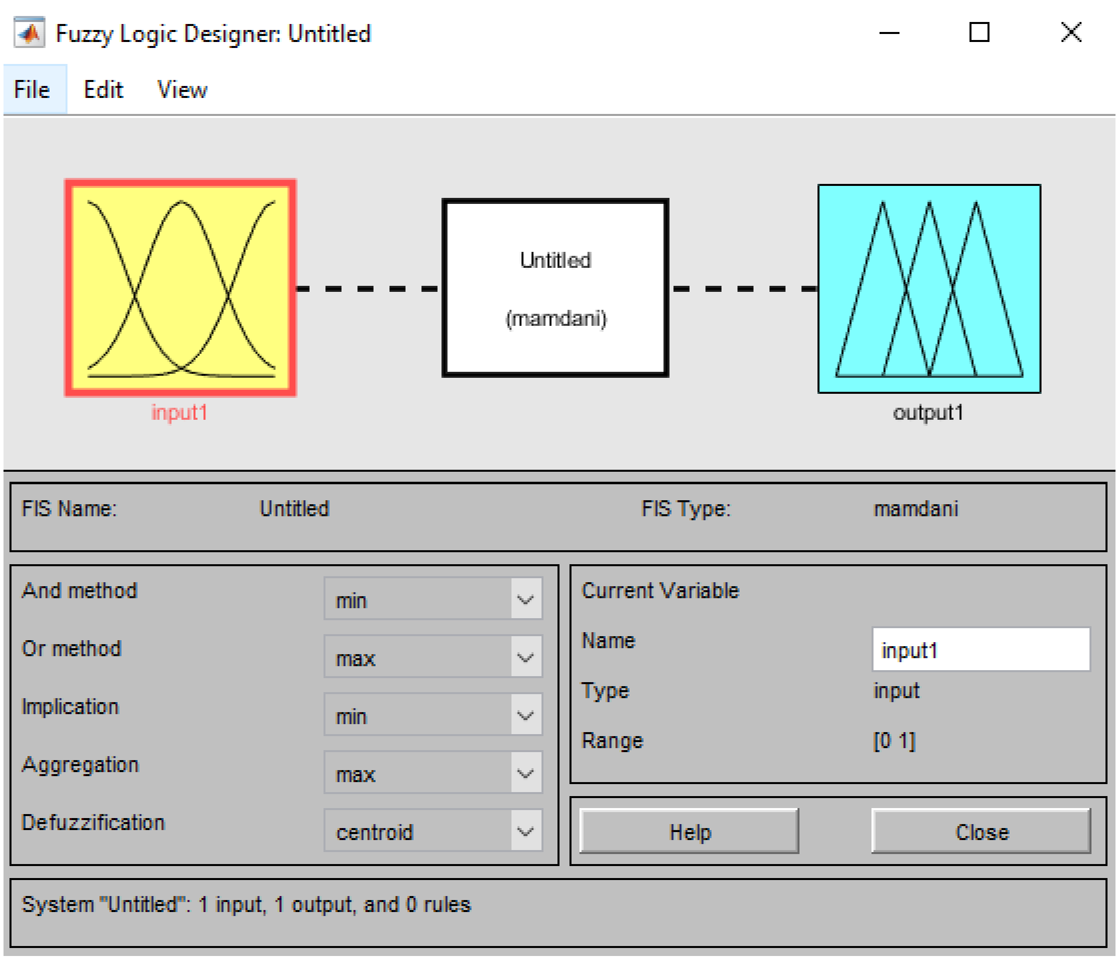

Figure 10 Fuzzy logic toolbox interface

2.2 Fuzzy logic controller for ball positioning

The overall sketch of designing the fuzzy controller for ball positioning

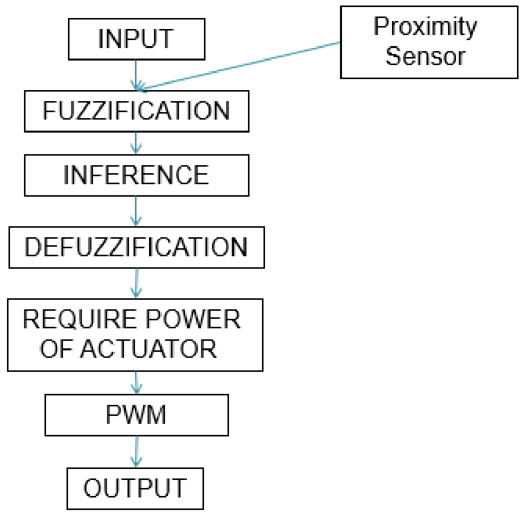

Figure 11 Fuzzy logic controller sketch

2.3 Overall work

This floating ball controller requires fuzzy logic and fuzzy membership functions. It also requires one output controller and two input controllers. We used an ultrasonic sensor, a computer cooling fan, and a microcontroller to calculate speed and distance. Figure 11 depicts how the project made a Ping-Pong ball float in the air. The device's hollow tube houses the ball. A proximity sensor emits infrared waves to determine how far the ball is from the top of the tube. A fan creates the airflow that moves the ping-pong ball up through the tube at the bottom of the tube. The most challenging part is getting the ping-pong ball into the tube and keeping it there once it's inside. A microcontroller controls the entire system. A few controls are required to keep the ping-pong ball in the air, such as measuring the speed and distance of the fans.

2.4 Procedure

- Open the fuzzy toolbox and select the two inputs on the top left side in the edit tab to get the two inputs of Ball position and Fan speed, and the output is PWM Output.

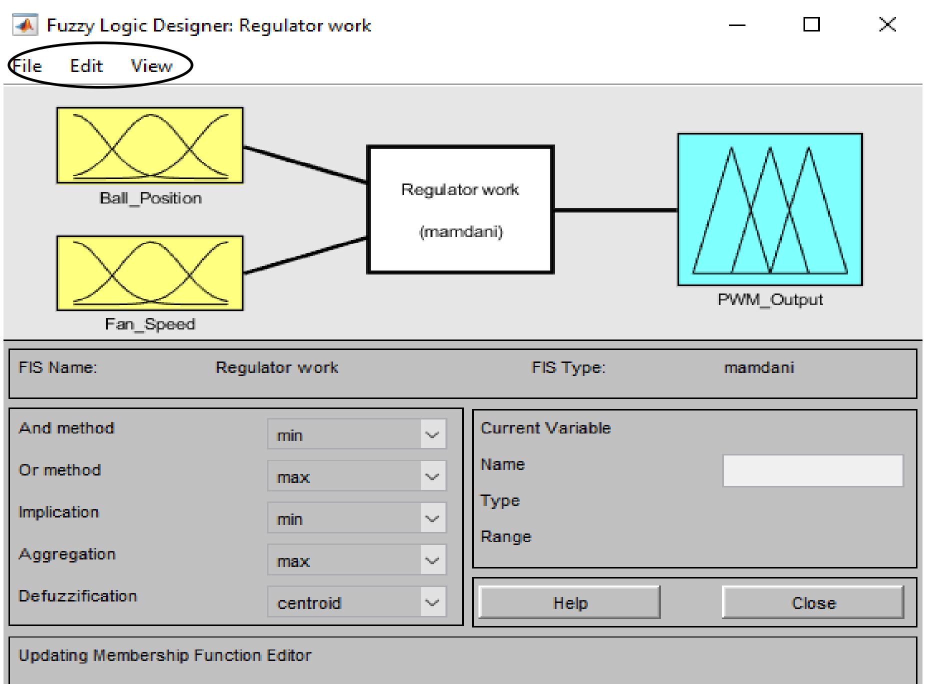

Figure 12 Input and Output of the regulator

- Set the membership functions and ranges of the input and output by double-clicking on them.

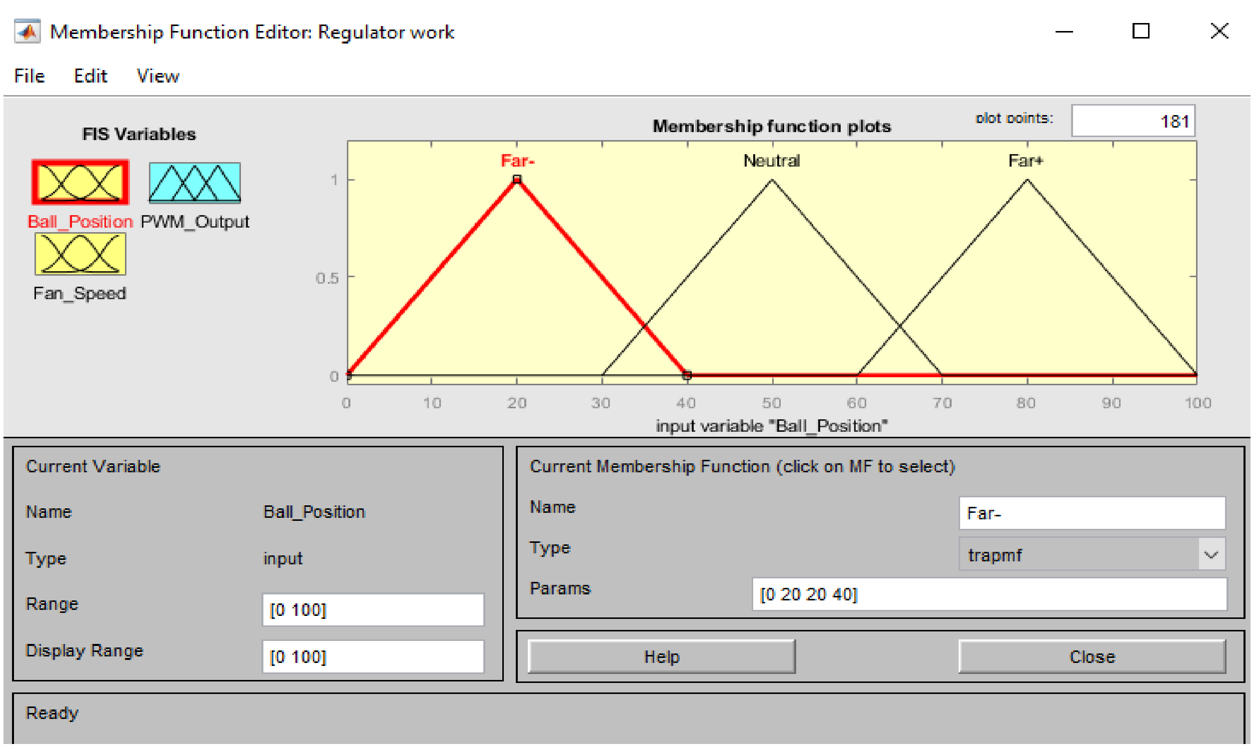

Figure 13 Ball Position membership function and ranges

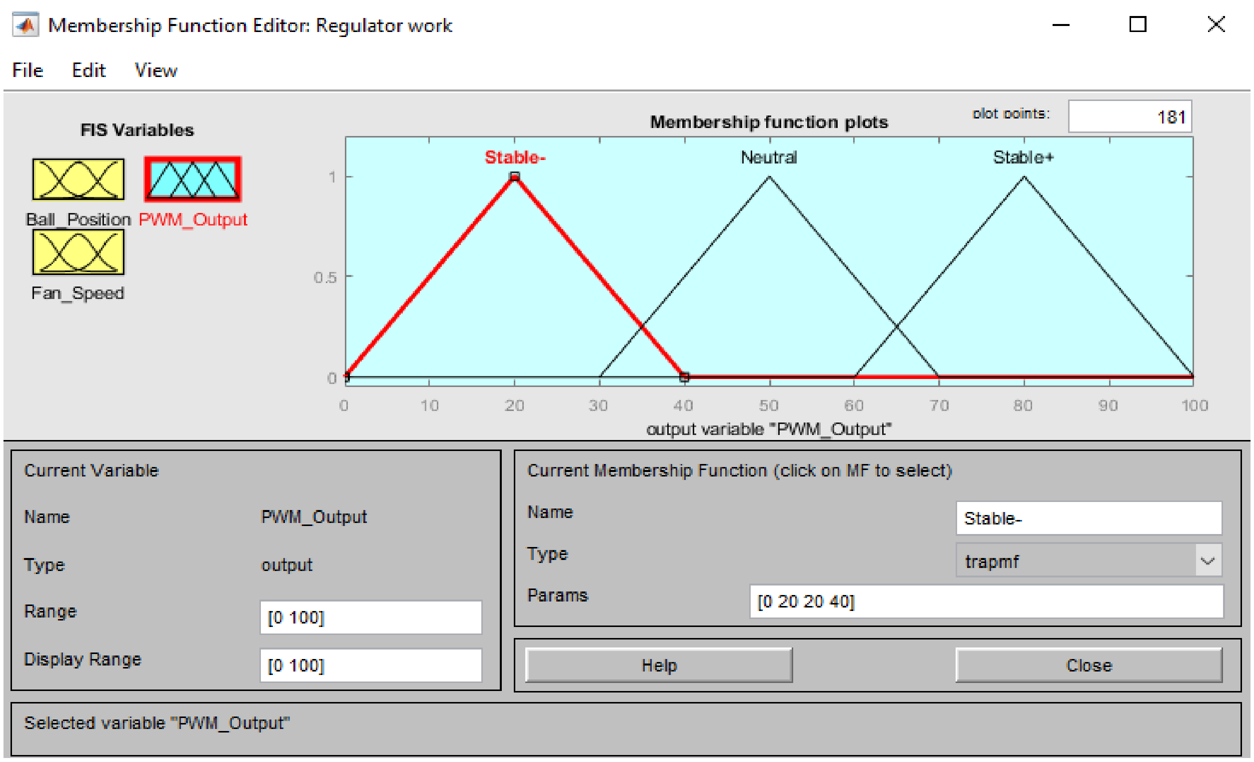

Figure 14 Fan Speed membership functions and ranges

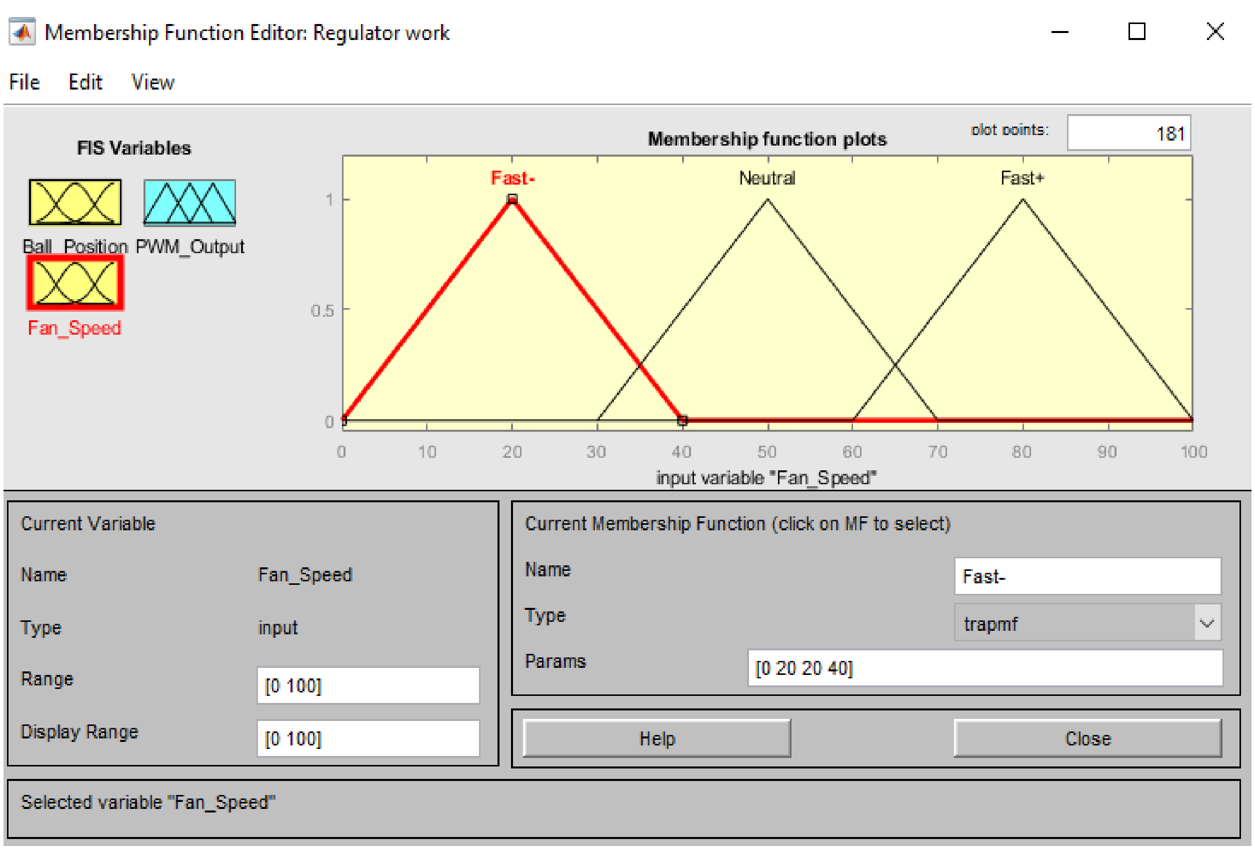

Figure 15 Output membership functions and ranges

- The main task is to set the rules; for this, click on the edit tab and the additional rules.

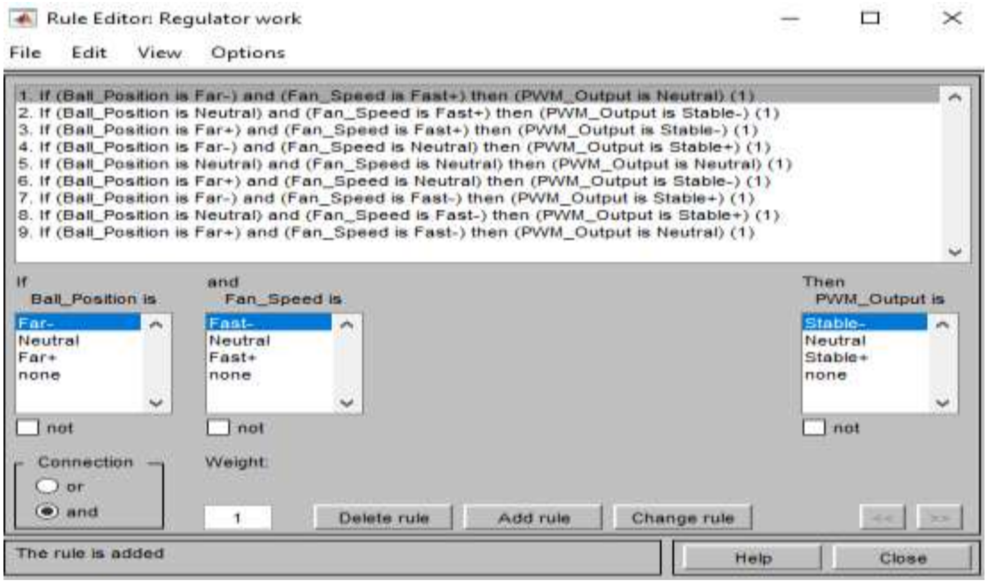

Figure 16 Rules defining for the controller

- After setting the rules, click on the rules viewer in the view to see the output of the rules in the form of graphs by sliding them left and right to see the changes in the regulator output.

Figure 17 Output of the rules in the rule viewer

2.5 Overall Functionality

To begin, first of all, built the prototype with the help of the fuzzy logic toolbox in MATLAB. You can use the controller's inputs for distance (position) and fan speed. It will allow you to continue your progress. Figures 14, 15, and 16 show how to define membership functions that determine an input's membership value. Following that, you will create fuzzy logic rules that determine what happens based on the inputs. To make the rules, Boolean logic was used. In this case, the output is the pulse width modulation (PWM) signal, which controls the rate at which the fan rotates.

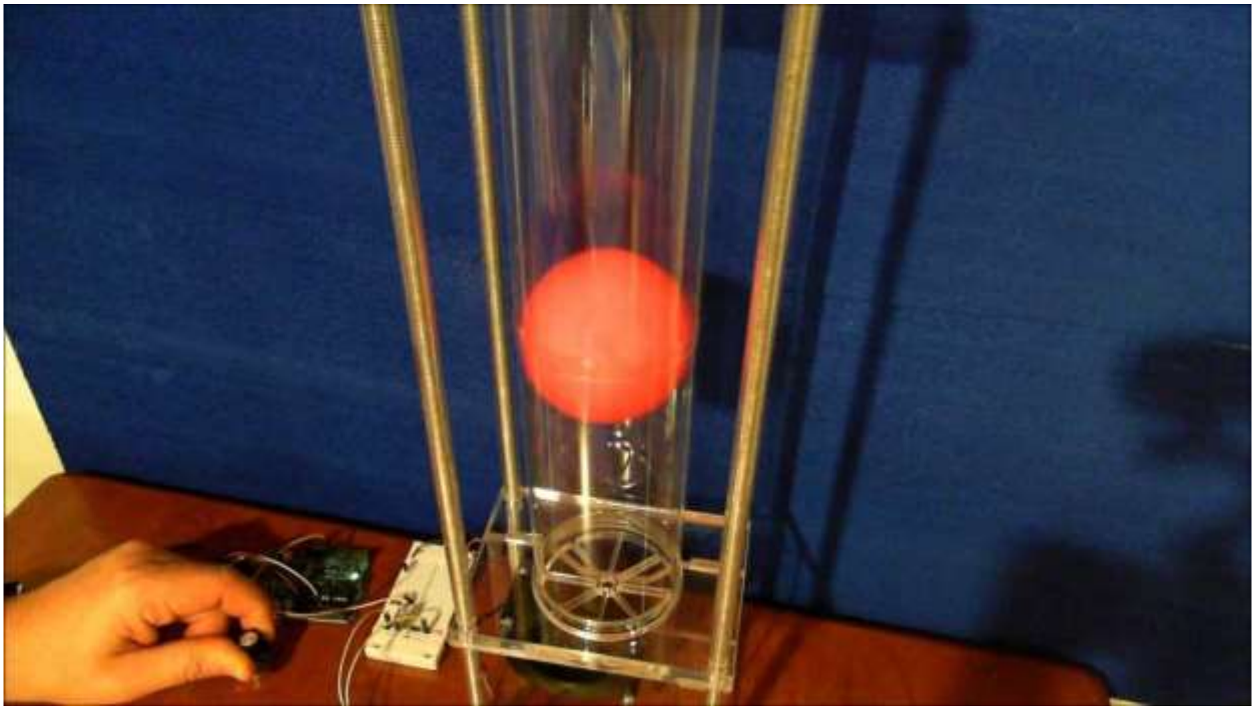

Figure 18 Air-held ping-pong ball in tube

References

[1]. https://codecrucks.com/what-is-fuzzy-membership-function-complete-guide/

[2]. https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html

[3]. https://researchhubs.com/post/engineering/fuzzy-system/fuzzy-membership-function.html

[4]. https://www.mathworks.com/help/fuzzy/index.html