ANKARA UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

COM-2536 PROJECT REPORT

Fuzzy Logic System

Ali Furkan KARAMAN 19290029

Ömer Faruk UYSAL 19290337

Ümit BOLELLİ 18290679

İnstructor Dr. Murat Osmanoğlu

Date (05, 2022)

DESIGNING A FUZZY LOGIC SYSTEM FOR DETERMINING CARBON DIOXIDE PERCENTAGE IN CARBONATED BEVERAGES

1. Abstract

Along with this project, the main idea is how to determine the carbon dioxide percentage in the Fizzy drink (seven-up). To accomplish this, a Fuzzy Logic System is built. With the help of this system, the carbon dioxide percentage comparison can be done between different situations according to the temperature of the product and the pressure on it. While building this system Mamdani inference and the center of area defuzzification method is used. Also, the system is implemented in the Matlab environment.

Keywords

Fuzzy logic, fuzzy logic system, Mamdani inference, center of area defuzzification, carbon dioxide percentage.

2. Introduction

The complexity of real-world problems increased with the development of technology and humanity built up more complex systems like Fuzzy systems. Actually, over time, Fuzzy systems have become necessary for modeling real-world problems since the classical sets theory and the binary logic system became inadequate to deal with uncertainty. But before getting into more detail about the Fuzzy system, first, we need to understand the concept of fuzzy logic.

.

2.1 What is Fuzzy Logic

In the boolean system truth value, 1.0 represents the absolute truth value and 0.0 represents the absolute false value. Therefore this system is not inadequate when dealing with multiple possible truth values. But in fuzzy logic, there is an intermediate value to present which is partially true and partially false, f.e. following figure.

Figure 1

So, Fuzzy logic is defined as a computing approach based on "degrees of truth" rather than the standard "true or false" (1 or 0). Boolean logic used by modern computers. The idea of fuzzy logic was first developed by Azerbaijani scientist Lotfi Zadeh of the University of California at Berkeley in the 1960s.

2.1.2 Fuzzy Sets

A fuzzy set is a set having degrees of membership between 1 and 0. Fuzzy sets are represented with a tilde character(~). For example, the Number of cars following traffic signals at a particular time out of all cars present will have a membership value between [0,1].

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$$

Figure 2

Fuzzy sets also satisfy every property of classical sets. Common operations on fuzzy sets:

$$\mu_{\tilde{C}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

Figure 3. Union of two fuzzy sets

$$\mu_{\tilde{D}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

Figure 4. The intersection of two fuzzy sets

$$\mu_{\tilde{E}}(x) = 1 - \mu_{\tilde{A}}(x)$$

Figure 5. Complement of fuzzy set

2.1.3 Membership Function

The "**μ**" represents the membership function that defines how each point in the input space is mapped to membership value between 0 and 1. Input space is often referred to as the universe of discourse or universal set (u), which contains all the possible elements of concern in each particular application.

There are largely three types of fuzzifiers:

- Singleton fuzzifier
- Gaussian fuzzifier
- Trapezoidal or triangular fuzzifier

2.2 What is Fuzzy System

Fuzzy systems are structures based on fuzzy techniques oriented towards information processing, where the usage of classical sets theory and binary logic is impossible or difficult. The Fuzzy system is based on four essential components: a fuzzification interface, knowledge base (rule and databases), inference, and defuzzification interface.

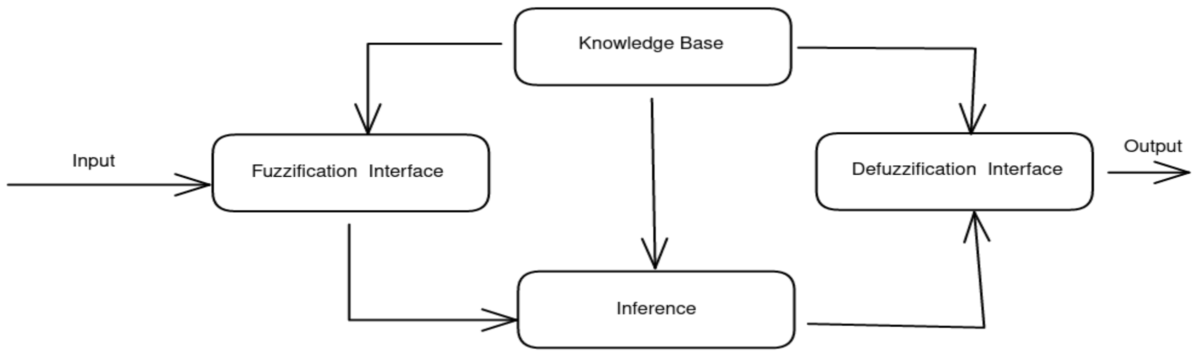


Figure 6. Fuzzy System

2.2.1 Fuzzy System Development Steps

A) Initialization: Defining the linguistic variables and terms, constructing the membership function, and setting the rule base to the system.
B) Fuzzification: Translating crisp inputs to fuzzy values with the membership function.
C) Inference: Setting the rules to the rule base and combining the outputs to each rule.
D) Defuzzification: Converting the output data to crisp values (readable values).

2.2.2 Linguistic Variables

The linguistic variable can be defined as a linguistic expression (one or more words) labeling information. It takes words as a value and converts them into the form of numerical values. For example, a membership function is labeled by expressions like “hot temperature” or “rich customer”.

2.2.3 Fuzzy System Components

2.2.3.1 Fuzzification Interface

The process of decomposing a system's crisp input and/or output values into a fuzzy value is known as fuzzification. The fuzzification technique allows system inputs and outputs to be represented in language terms, allowing simple rules to be used to express a complex system.

2.2.3.2 Knowledge Base

The fuzzy knowledge base represents the facts of the rules and linguistic variables based on the fuzzy set theory so that the knowledge base systems will allow approximate reasoning. Rules are generally created in the if-then form.

2.2.3.3 Fuzzy Inference

The process of generating a mapping from a given input to an output using fuzzy logic is known as fuzzy inference. The mapping then serves as a foundation for making decisions and identifying patterns. All of the previously mentioned components, such as membership functions, fuzzy logic operators, and if-then rules, are used in the fuzzy inference process. There are two ways commonly used to implement fuzzy inference. The first one is Mamdani-type inference.

Mamdani fuzzy inference was first introduced as a method to create a control system by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani-type inference expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable, which needs defuzzification.

Mamdani systems are well-suited to expert system applications where the rules are created from human expert knowledge, such as medical diagnostics because their rule bases are more intuitive and easier to understand.

The second one to implement fuzzy inference is the Larsen-type fuzzy inference. In Larsen fuzzy inference output is generated as the product operation.

2.2.3.4 Defuzzification Interface

It can be defined as the process of reducing a fuzzy set into a crisp set or converting a fuzzy member into a crisp member.

Since the fuzzification process involves conversion from crisp quantities to fuzzy quantities in the end, it is necessary to defuzzify the result or rather "fuzzy result" so that it must be converted to a crisp result. Mathematically, the process of defuzzification is also called "rounding it off".

Some of the different methods of defuzzification are described below;

2.2.3.4.1 Max-Membership Method

This method is limited to peak output functions and is also known as the height method. Mathematically it can be represented as follows.

$$\mu_{\tilde{A}}(x^*) > \mu_{\tilde{A}}(x) \ \ for \, all \, x \in X$$

Figure 7

Here, x^* is the defuzzified output.

2.2.3.4.2 Centroid Method

This method is also known as the center of area or the center of gravity method. Mathematically, the defuzzified output x^* will be represented as follows.

$$x^* = \frac{\int \mu_{\bar{C}}(x) \, . \, x \, dx}{\int \mu_{\bar{C}}(x) \, dx}$$

Figure 8

2.2.3.4.3 Mean-Max Membership

This method is also known as the middle of the maxima. Mathematically, the defuzzified output x^* will be represented as follows.

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

Figure 9

3. Construction

To construct our Fuzzy Logic System program, we need inputs and outputs. We have 2 inputs: temperature of Fizzy drink, pressure on the Fizzy drink, and one output: percentage of the carbon dioxide (CO2) in the Fizzy.

To generate our program, as a first step, we declared the Mamdani fuzzy inference system in the following code snippet and stored it in the variable mem (Figure 10).

```
mem =  mamfis('Name',"P.O.C");
```

Figure 10. Creating Mamdani Inference Object

The fuzzified input values for the temperature to be imported into our Mamdani Inference Object are presented in Figure 11. The triangular membership function type is used for temperature input, as well. The graphic of the temperature membership function is shown in figure 11. Here, [7 16] range is chosen for 5 Linguistic variables: Very cold, Cold, Normal, Hot, Very hot.

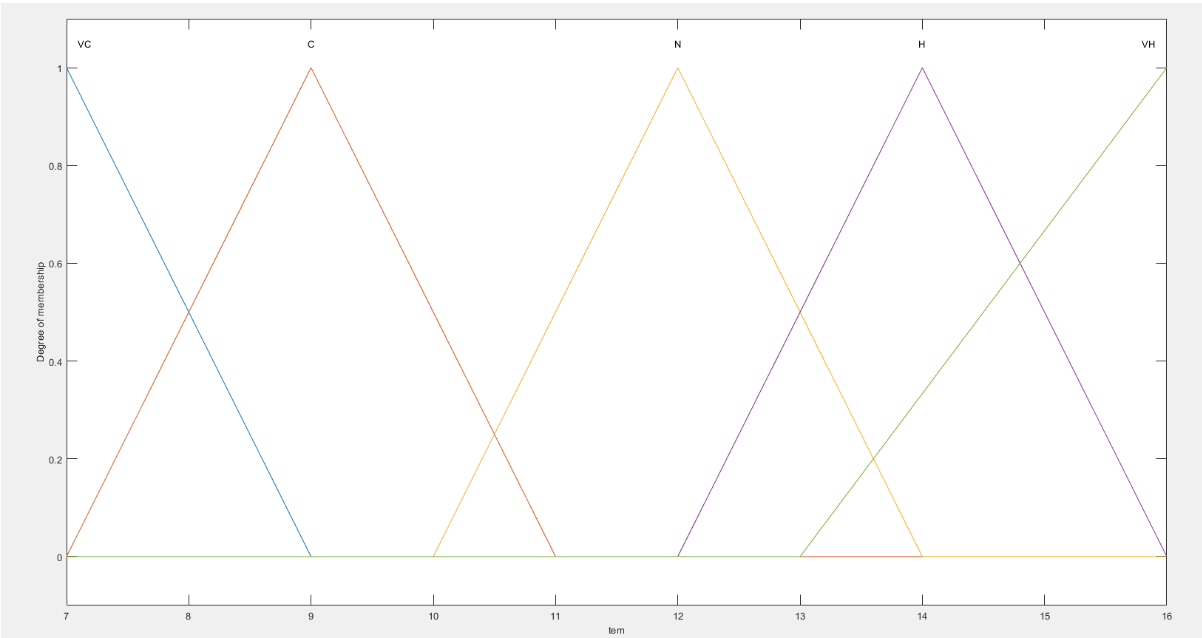


Figure 11. Temperature membership function values

So according to specified values, we added our first input with the addInput method and we named it as “tem”.

```
mem = addInput(mem,[7,16], 'Name', "tem");
```

Figure 12

Then, we added our membership function values in the given ranges with respect to specified linguistic terms. To do that we used the addMf method.

```
mem = addMF(mem,"tem","trimf",[7 7 9],"Name", "VC");
mem = addMF(mem,"tem","trimf",[7 9 11],"Name", "C");
mem = addMF(mem,"tem","trimf",[10 12 14],"Name", "N");
mem = addMF(mem,"tem","trimf",[12 14 16],"Name", "H");
mem = addMF(mem,"tem","trimf",[13 16 16],"Name", "VH");
```

Figure 13

Again the triangular membership function type is used for our second input pressure. The graphic of the pressure membership function is shown in figure 14. In here, [1.75, 4.00] range is chosen for 5 Linguistic variables: Very cold, Cold, Normal, Hot, Very hot.

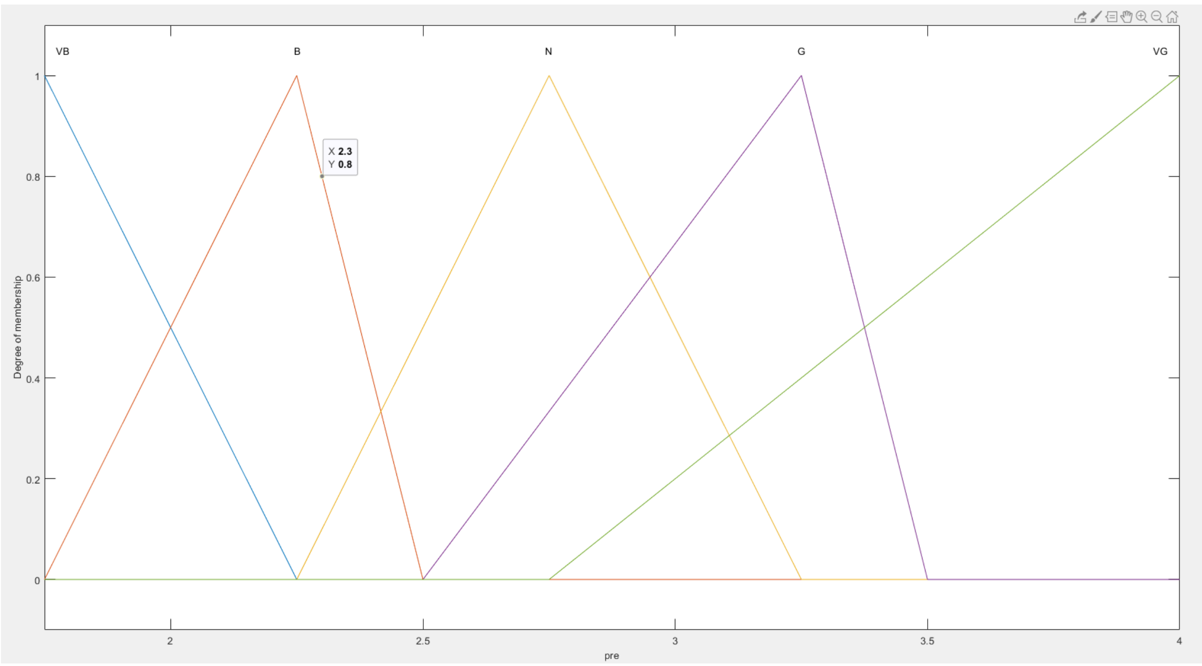


Figure 14. Input membership function values for pressure

The second input for pressure is added to our inference object and named “pre” in the same way.

```
mem = addInput(mem,[1.75 4.00], "Name", "pre");
```

Figure 15

And the membership function values are defined as following:

```
mem = addMF(mem,"pre","trimf",[1.75 1.75 2.25],"Name", "VB");
mem = addMF(mem,"pre","trimf",[1.75 2.25 2.50],"Name", "B");
mem = addMF(mem,"pre","trimf",[2.25 2.75 3.25],"Name", "N");
mem = addMF(mem,"pre","trimf",[2.50 3.25 3.50],"Name", "G");
mem = addMF(mem,"pre","trimf",[2.75 4.00 4.0],"Name", "VG");
```

Figure 16

At the end of the Fuzzification stage, we added our first and only output “percentage ” into our object as “per” (Figure 17).

```
mem = addOutput(mem, [2.0 6.0], "Name", "per");
```

Figure 17. Percentage Membership function values

Then defined the membership function values (Figure 18) .

```
mem = addMF(mem, "per", "trimf", [2.0 2.0 3.0], "Name", "VB");
mem = addMF(mem, "per", "trimf", [2.0 3.0 4.0], "Name", "B");
mem = addMF(mem, "per", "trimf", [3.0 4.0 5.0], "Name", "N");
mem = addMF(mem, "per", "trimf", [4.0 5.0 6.0], "Name", "G");
mem = addMF(mem, "per", "trimf", [5.0 6.0 6.0], "Name", "VG");
```

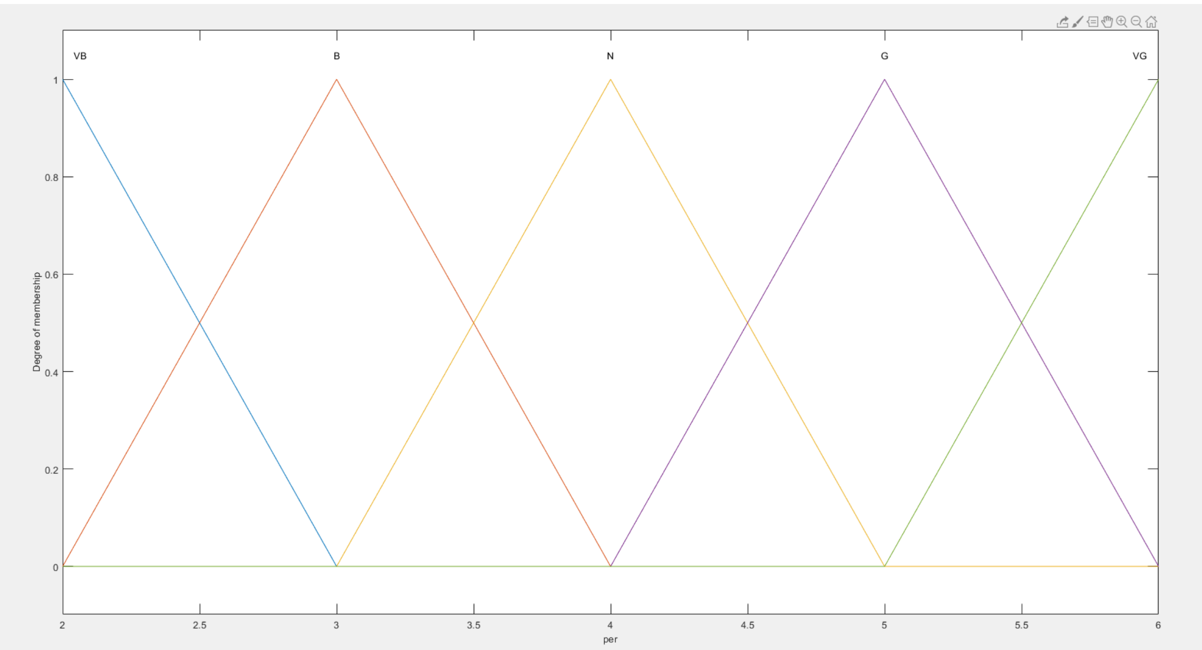


Figure 18

The second step is adding the rule base into our fuzzy inference system, knowledge base stage. We have 25 fuzzy rules come by 5 linguistic terms for Temperature and 5 terms for Pressure. Also, the rules that construct the output are given as follows in the form of rule-matrix (Figure 19).

Rule Matrix	Verycold	Cold	Normal	Hot	Veryhot
Verybad	Normal	Bad	Bad	Bad	Verybad
Bad	Normal	Good	Normal	Bad	Bad
Normal	Good	Good	Normal	Normal	Normal
Good	Verygood	Good	Good	Normal	Normal
Verygood	Verygood	Verygood	Verygood	Good	Good

Figure 19. Rule matrix

After that, we specify if-then rules in our program using linguistic expressions (Figure 20).

```
rule1 = "If tem is VC and pre is VB then per is N";
rule2 = "If tem is C and pre is VB then per is B";
rule3 = "If tem is N and pre is VB then per is B";
rule4 = "If tem is H and pre is VB then per is B";
rule5 = "If tem is VH and pre is VB then per is VB";
rule6 = "If tem is VC and pre is B then per is N";
rule7 = "If tem is C and pre is B then per is G";
rule8 = "If tem is N and pre is B then per is N";
rule9 = "If tem is H and pre is B then per is B";
rule10 = "If tem is VH and pre is B then per is B";
rule11 = "If tem is VC and pre is N then per is G";
rule12 = "If tem is C and pre is N then per is G";
rule13 = "If tem is N and pre is N then per is N";
rule14 = "If tem is H and pre is N then per is N";
rule15 = "If tem is VH and pre is N then per is N";
rule16 = "If tem is VC and pre is G then per is VG";
rule17 = "If tem is C and pre is G then per is G";
rule18 = "If tem is N and pre is G then per is G";
rule19 = "If tem is H and pre is G then per is N";
rule20 = "If tem is VH and pre is G then per is N";
rule21 = "If tem is VC and pre is VG then per is VG";
rule22 = "If tem is C and pre is VG then per is VG";
rule23 = "If tem is N and pre is VG then per is VG";
rule24 = "If tem is H and pre is VG then per is G";
rule25 = "If tem is VH and pre is VG then per is G";
```

Figure 20. If - then rules

In order to add rules to our Inference we use addRule method (Figure 21).

```
mem = addRule(mem, rules);
```

Figure 21. addRule method

As a third and last step, we are ready to evaluate the percentage of the carbon-dioxide for specific input values with our Mamdani inference object mem. After the evaluation, defuzzification is needed since Mamdani inference generates a fuzzy output as mentioned in the previous section. This process is handled by the following code snippet,

```
output = evalfis(mem,[12, 3]);
```

Figure 22

The generated output is,

```
output =

    4.5738
```

Figure 23

4. Implementation

First example input: Suppose that temperature is 14 (Hot) and the pressure is 2.75 (Normal).

.

```
yellow = evalfis(mem,[14, 2.75]);
```

```
>> run("Mamdani")
>> yellow

yellow =

    4.0000
```

Figure 24

Second example input: Suppose that temperature is 9.5 (Cold) and the pressure is 3.75 (Very Good)

```
red = evalfis(mem,[9.5, 3.75]);
```

```
>> run("Mamdani")
>> red

red =

     5.6612
```

Figure 25

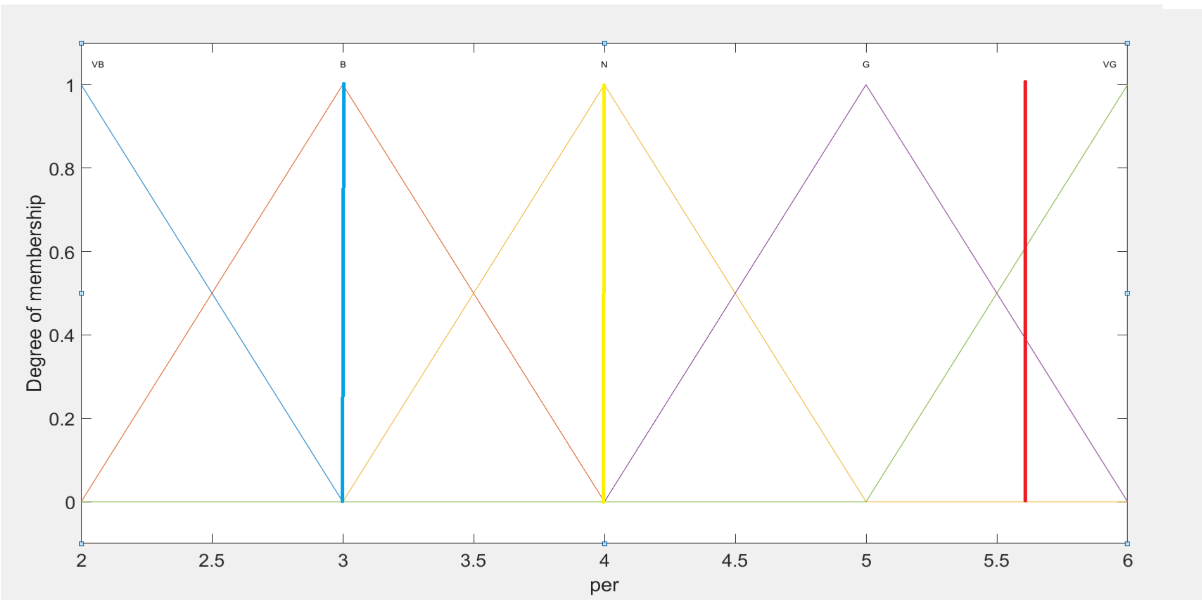
Third example input: Suppose that temperature is 12 (Normal) and the pressure is 1.75 (Very Bad)

```
blue = evalfis(mem,[12, 1.75]);
```

```
>> run("Mamdani")
>> blue

blue =

     3.0000
```



Rule Matrix	Verycold	Cold	Normal	Hot	Veryhot
Verybad	Normal	Bad	Bad	Bad	Verybad
Bad	Normal	Good	Normal	Bad	Bad
Normal	Good	Good	Normal	Normal	Normal
Good	Verygood	Good	Good	Normal	Normal
Verygood	Verygood	Verygood	Verygood	Good	Good

Figure 26

5. References

[1] Lotfi A., Z. 1965. "Fuzzy sets". *Information and Control*. San Diego. **8** (3): 338–353. doi:10.1016/S0019-9958(65)90241-X. ISSN 0019-9958. Wikidata Q25938993.

[2] Mamdani, E.H., and S. Assilian. 'An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller'. *International Journal of Man-Machine Studies* 7, no. 1 (January 1975): 1–13. https://doi.org/10.1016/S0020-7373(75)80002-2.

[3] https://www.mathworks.com/help/fuzzy/fuzzy-inference-system-modeling.html?s_tid=CRUX_lftnav

6. The Source Code Used In The Project

```
mem =  mamfis('Name',"P.O.C");
mem = addInput(mem,[7,16], 'Name', "tem");
mem = addMF(mem,"tem","trimf",[7 7 9],"Name", "VC");
mem = addMF(mem,"tem","trimf",[7 9 11],"Name", "C");
mem = addMF(mem,"tem","trimf",[10 12 14],"Name", "N");
mem = addMF(mem,"tem","trimf",[12 14 16],"Name", "H");
mem = addMF(mem,"tem","trimf",[13 16 16],"Name", "VH");
mem = addInput(mem,[1.75 4.00], "Name", "pre");
mem = addMF(mem,"pre","trimf",[1.75 1.75 2.25],"Name", "VB");
mem = addMF(mem,"pre","trimf",[1.75 2.25 2.50],"Name", "B");
mem = addMF(mem,"pre","trimf",[2.25 2.75 3.25],"Name", "N");
mem = addMF(mem,"pre","trimf",[2.50 3.25 3.50],"Name", "G");
mem = addMF(mem,"pre","trimf",[2.75 4.00 4.0],"Name", "VG");
mem = addOutput(mem, [2.0 6.0], "Name", "per");
mem = addMF(mem, "per", "trimf", [2.0 2.0 3.0], "Name", "VB");
mem = addMF(mem, "per", "trimf", [2.0 3.0 4.0], "Name", "B");
mem = addMF(mem, "per", "trimf", [3.0 4.0 5.0], "Name", "N");
mem = addMF(mem, "per", "trimf", [4.0 5.0 6.0], "Name", "G");
mem = addMF(mem, "per", "trimf", [5.0 6.0 6.0], "Name", "VG");
rule1 = "If tem is VC and pre is VB then per is N";
rule2 = "If tem is C and pre is VB then per is B";
rule3 = "If tem is N and pre is VB then per is B";
rule4 = "If tem is H and pre is VB then per is B";
rule5 = "If tem is VH and pre is VB then per is VB";
rule6 = "If tem is VC and pre is B then per is N";
rule7 = "If tem is C and pre is B then per is G";
rule8 = "If tem is N and pre is B then per is N";
rule9 = "If tem is H and pre is B then per is B";
rule10 = "If tem is VH and pre is B then per is B";
rule11 = "If tem is VC and pre is N then per is G";
rule12 = "If tem is C and pre is N then per is G";
rule13 = "If tem is N and pre is N then per is N";
rule14 = "If tem is H and pre is N then per is N";
rule15 = "If tem is VH and pre is N then per is N";
rule16 = "If tem is VC and pre is G then per is VG";
rule17 = "If tem is C and pre is G then per is G";
rule18 = "If tem is N and pre is G then per is G";
rule19 = "If tem is H and pre is G then per is N";
rule20 = "If tem is VH and pre is G then per is N";
rule21 = "If tem is VC and pre is VG then per is VG";
rule22 = "If tem is C and pre is VG then per is VG";
rule23 = "If tem is N and pre is VG then per is VG";
rule24 = "If tem is H and pre is VG then per is G";
rule25 = "If tem is VH and pre is VG then per is G";
rules = [rule1 rule2 rule3 rule4 rule5 rule6 rule7 rule8 rule9 rule10 rule11 rule12 rule13 rule14 rule15 rule16 rule17 rule18
rule19 rule20 rule21 rule22 rule23 rule24 rule25];
mem = addRule(mem, rules);
yellow = evalfis(mem,[14,2.75]);
red = evalfis(mem,[9.5, 3.75]);
blue = evalfis(mem,[12, 1.75]);
```