Lookup Table Driven Uncertainty Avoider Based Interval Type-2 Fuzzy System Design

Sirwan M. KekShar, Sadegh A. Aminifar

Computer Science Department, Science Faculty, Soran University, Soran, Erbil, Kurdistan Region, Iraq

Abstract: This research presents the design and simulation of interval type II fuzzy systems (IT2FSs) using the uncertainty avoidance concept and the lookup table method. In addition to the conventional lookup table method used in most literatures for type_1 fuzzy systems (T1FSs), which is based on selecting rules with higher weights and strength, this research presents a novel method for T1FS that uses an antecedent with opposing consequents and leads to multiple rules for each input membership function. Furthermore, this research provides interval type_2 membership functions (IT2MFs) by injecting uncertainty properly. It utilizes the lookup table method to design IT2FS by selecting a higher strength rule for each input membership function. The fuzzy system will then use the uncertainty avoidance formula for designing a multiple-rule system for each input membership function. The new structure was used for modeling a steam chamber process in a scientific application. The first set of measurement data used two inputs and one output from the steam chamber system to train the process model according to the type_1 and interval type II structures proposed in this paper. In step two, the designed model was tested with the measured data, and the model approximated the test data outputs with a reasonable accuracy compared to type_1 fuzzy system. A MATLAB code for data driven interval type II fuzzy structure were presented.

1- Introduction

In general, the main goal of scientific and engineering research is to present solutions and optimize real-world problems. One such problem is modeling real systems. Real systems are also suffer from problems with various uncertainties (unreliability), and finding a compatible system that is able to model and mitigate uncertainties is a fascinating subject for engineers.

Interest in games and similarities between gaming elements and real-world conditions, such as football players and the ball, motivated me to study this subject. Modeling the behavior of a football player, the ball's movement in response to different hits, or whether the ball is stationary or moving beforehand are complex mathematical problems. At the same time, there are numerous interactions that occur in real phenomena, not all the data needed for system design can be obtained, and the existing data cannot be described in such systems with mathematical models.It is commonly known that most systems suffer from a lack of information due to natural or fringe reasons. Hence, even if model approximation is possible through mathematical functions, obtaining the complete information required for modeling such systems can be too complex to be economical and cost-effective. Hence, fuzzy systems are well known for their use in such situations. Mathematical system modeling is avoided for the aforementioned reasons, and one of the goals of this research is to model according to system input and output data measurements, which will necessitate using the lookup table method. The higher performance of interval type II systems, especially interval type II fuzzy systems with uncertainty avoidance, was the main motivational factor for using such systems to model and applying them to the steam chamber as an example to test the proposed methods.

Even if the model is sufficiently accurate, there are numerous cases of uncertainty, e.g. sensor inaccuracies or non-linear device performance. In such cases, fuzzy systems, specifically interval type II systems for uncertainty avoidance, and the lookup table due to its ability to select optimal rules, can be regarded as optimal solutions for system design.

Defuzzification and Uncertainty: The defuzzification block includes a set of rules that define a system's behavior while assuming a description of a broad set of control levels. Moreover, none of these control levels can be implemented in the proposed system, and only one control level that the fuzzy system is able to find is considered. Rather, the system should only be controlled by a specific control level that often has a weak position in word computing. At the same time, fuzzy methods are used for restricting the space described by the rules database. Different defuzzification methods lead to different results. Hence, one defuzzification method is more compatible with word computing than others, which is used by designers for a more probable estimation of outputs according to a descriptive understanding of system behavior. Furthermore, the system is considered as a CW, which will enable designers to use conceptual tools and make useful changes to systems according to descriptive information.

As shown in an interval type II system's structural block diagram in figure 28, the defuzzification block consists of two parts: Uncertainty avoidance and defuzzification using a common method. In this regard, uncertainty in type II fuzzy systems is explored briefly.

2- Methodology

IT2FS Structure: In this paper uncertainty is the uncertainty of membership degree of members of a fuzzy set. Figure 1 shows our interpretation of IT2MF as an extension of type-1 fuzzifier. The fuzzy system proposed in this research uses a readout of IT2FSs that defines IT2FS as an extension of T1FSs. Inference engine normally works like standard IT2FS. In last stage of IT2FS as shown in Figure 1, uncertainty remover (Decisiveness block) is added before type 1 defuzzification, which is has uncertainty avoidance property.

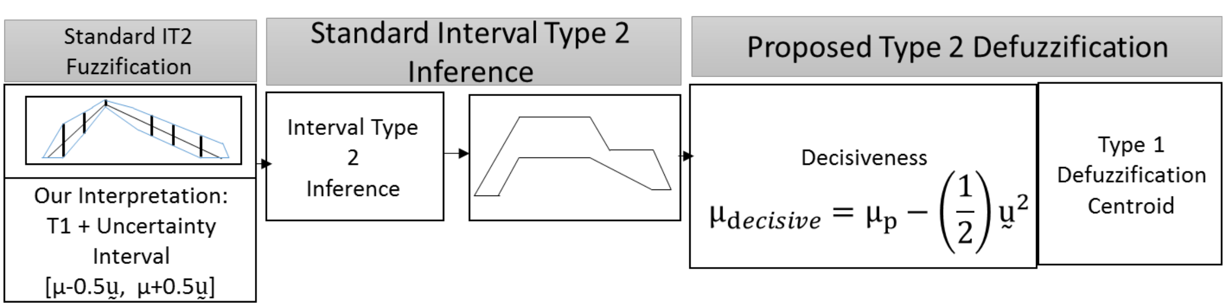


Figure 1: IT2FS uncertainty avoidance structure

Data driven modelling: After examining system modeling, brief explanations regarding data modeling and the data structure used in this thesis will be discussed and evaluated next.

In general, data modeling is very useful in various sciences, especially engineering, and is suitable for designing complex systems and converting them into easily fathomable structures and diagrams. This diagram can be used as a work plan for redesigning or reengineering a system. Data modeling is often created during project analysis and design to ensure that all aspects of a system are properly understood and considered. Data modeling can actually be considered as a flowchart that represents the relationship between data. Since showing all possible data relationships is a time-consuming and complex operation, this process cannot be undertaken in one step, but in stages and using a precise system that considers all aspects. Hence, the best approach is to use models based on conceptual and logical systems, since data modeling according to logical systems allows project designers to quickly identify errors and apply required changes before the process is initiated. To this end, the structure of fuzzy systems and the lookup table method will be used to approximate and model steam chamber function due to their advantages that have been mentioned.

A set of data is required to model any system and be able to evaluate the performance of the structure in question. Such data is divided into two general categories, namely training data1 and test data2. Training data are used for training oneself and preparing to use basic data in the system. This data type is used to direct the training process. Test data, however, is used independently to determine the system's final quality for the structure under consideration with regards to accuracy and determining the final result. Both data types are used in this thesis thus: Training data is used to evaluate model one input-one output type I and interval type II fuzzy systems using the lookup table method while assuming that the intended and final structure of this thesis for modeling the steam chamber is two inputs-one output. The training data used are formulas according to an accurate mathematical structure, as will be explained. The test data, however, are produced and extracted practically according to the current performance and the steam chamber system and are used for modeling two input-one output type I and interval type II fuzzy systems using the lookup table method so as to achieve accurate and 100% practical results, as will be presented in section 4.

1 Training Sets

2 Test Data

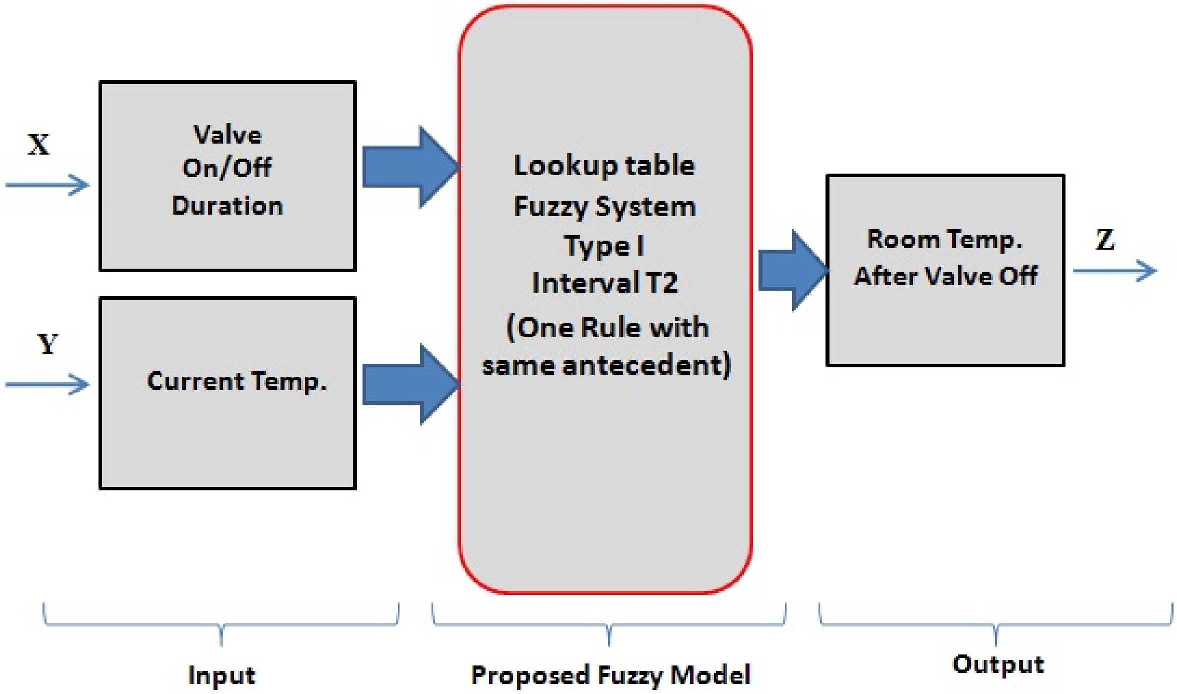


Figure 2 Fuzzy System Modeling According to the Normal Lookup Table Method (Selecting the Rule with the Highest Degree in Every Category)

Modeling the Two Input-One Output Interval Type II Fuzzy System Using the Optimal Lookup Table Method

Designing a two input-one output interval type II fuzzy system with the optimal lookup table method is exactly the same as the previous state regarding the type and range of system input and output parameter variance and creating Gaussian membership functions as well as the rules database, which correspond to the first, second and third steps of modeling. The only major difference in modeling using this method is in applying and extracting rules using the lookup table method. The rules are normalized in step four after computing the rule weights for each set, and 90% of the rules in each set with the highest strength are considered so that the operation enables input membership functions to cover more of the output membership functions according to existing rules and their different weights. As shown, in conditions involving the use of the normal lookup table, the entire interval type II fuzzy system in question is modeled and approximated only using a limited set of select rules. This section, however, uses more rules and compares the result with the previous state in terms of accuracy and error. To this end, the algorithm presented in figure 40 is considered for designing this system:

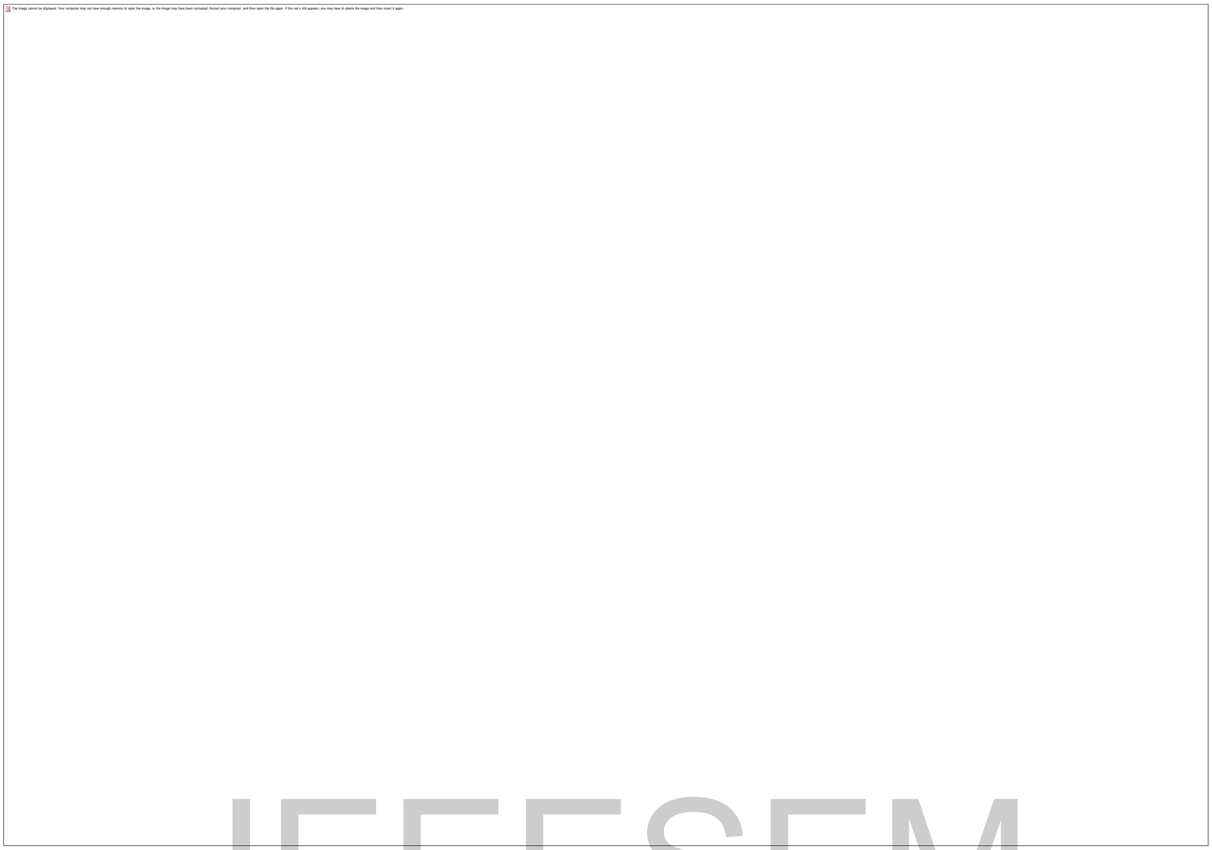


Figure 1The Block Diagram of the Different Modeling Stages for the Two Input-One Output Interval Type II Fuzzy System Using the Optimal Lookup Table Method

Step Four: In accordance to the aforesaid explanations, the strength is calculated for each rule according to table 18, and each set's rules are normalized.

All the strong rules are considered according to table ... after normalizations to be used for modeling the fuzzy system in question, and the results are compared to the previous state in chapter four.

Step Five: Afterward, 90% of the rules with the highest strength are selected from each set to model the system using the optimal lookup table method, which is used to better and more accurately model the system. The operations are run in Matlab and the resulting output along with the system's accuracy and error is calculated and compared to the design using the normal lookup table method, all of which are extensively presented in chapter four.

Chapter 4 – Result and discussion

Chapter 3 presented evaluations of type I and interval type II fuzzy system modeling methods using the lookup table method for different one input-one output and two inputs-one output structures with training data and test data, and examined the implementation stages. This section simulates various modeled systems in Matlab and compares different structures on accuracy and error according to the normal and proposed optimized lookup table methods and presents the results.

The simulation results for each structure modeled for the proposed fuzzy systems are evaluated according to table 19:

Table 1Different Modeled Fuzzy Systems

System Type	Rule Extracting	# of Input/output
Type 1 fuzzy system	One Rule for each set with equal antecedents	One-Input One-Output
	One Rule for each set with equal antecedents	Two-Input One-Output
	3/4 Rule for each set with equal antecedents	One-Input One-Output
	3/4 Rule for each set with equal antecedents	Two-Input One-Output
Interval Type 2 Fuzzy System	One Rule for each set with equal antecedents	One-Input One-Output
	One Rule for each set with equal antecedents	Two-Input One-Output
	3/4 Rule for each set with equal antecedents	One-Input One-Output
	3/4 Rule for each set with equal antecedents	Two-Input One-Output

The simulation results for each fuzzy system modeled using the lookup table method in Matlab will be presented as follows.

4.1. Simulating One Input-One Output Type I Fuzzy System Using the Normal Lookup Table Method in Matlab

As explained in Chapter 3, a set of training data are used to simulate this system as shown in equation (23) in order to generate input and output data shown in table 20:

(23)y=1+0.1*sin(x)

Table 2The y=1+0.1*sin(x) Input and Output Equation

System Type	Rule Extracting	# of Input/output
Type 1 fuzzy system	One Rule for each set with equal antecedents	One-Input One-Output
	One Rule for each set with	Two-Input One-Output

	equal antecedents	
	3/4 Rule for each set with equal antecedents	One-Input One-Output
	3/4 Rule for each set with equal antecedents	Two-Input One-Output
Interval Type 2 Fuzzy System	One Rule for each set with equal antecedents	One-Input One-Output
	One Rule for each set with equal antecedents	Two-Input One-Output
	3/4 Rule for each set with equal antecedents	One-Input One-Output
	3/4 Rule for each set with equal antecedents	Two-Input One-Output

In order to simulate this system in Matlab, the input and output variance intervals are set according to table 21 and equation **(23)**, which is used to create Gaussian functions, are considered according to table 21:

(24) U=exp(-0.5*((F-G)/K).^2)

Table 3One Input Type I Fuzzy System's Input and Output Gaussian Functions

Dedicated Gaussian MFs for Inputs	**Gaussian MFs for Output**
x=1:1:32;	y=0.9:0.0065:1.1
A1= exp(-0.5*((x-0)/2.5).^2)	B1=exp(-0.5*((y-0.9)/0.02).^2)
A2= exp(-0.5*((x-8)/2.5).^2)	B2=exp(-0.5*((y-0.95)/0.02).^2)
A3= exp(-0.5*((x-16)/2.5).^2)	B3=exp(-0.5*((y-0.1)/0.02).^2)
A4= exp(-0.5*((x-24)/2.5).^2)	B4=exp(-0.5*((y-1.05)/0.02).^2)
A5= exp(-0.5*((x-32)/2.5).^2)	B5=exp(-0.5*((y-1.1)/0.02).^2)

According to the equations presented in table 21, the structure of the program written to simulate these functions in Matlab is as follows:

Table 4The Programming Structure of the One Input Type I Fuzzy System's Input and Output Functions

Program Structure	**Description**
x=1:1:32;	Input Range

A1=exp(-0.5*((x-0)/2.5).^2);	Gaussian MFs for total Input range
A2=exp(-0.5*((x-8)/2.5).^2);	
A3=exp(-0.5*((x-16)/2.5).^2);	
A4=exp(-0.5*((x-24)/2.5).^2);	
A5=exp(-0.5*((x-32)/2.5).^2);	
> y=0.9:0.0065:1.1;	Output Range
B1=exp(-0.5*((y-0.9)/0.02).^2);	Gaussian MFs for total Output range
B2=exp(-0.5*((y-0.95)/0.02).^2);	
B3=exp(-0.5*((y-0.1)/0.02).^2);	
B4=exp(-0.5*((y-1.05)/0.02).^2);	
B5=exp(-0.5*((y-1.1)/0.02).^2);	
plot(y,B1,y,B2,y,B3,y,B4,y,B5)	Illustrating of Output
plot(x,A1,x,A2,x,A3,x,A4,x,A5)	Input Illustrations

According to table 22, the structure of the input and output functions simulated in Matlab are respectively shown in figures 41 and 42:

Figure 2Simulating The One Input-One Output Type I Fuzzy System’s Five Gaussian Input Functions in Matlab

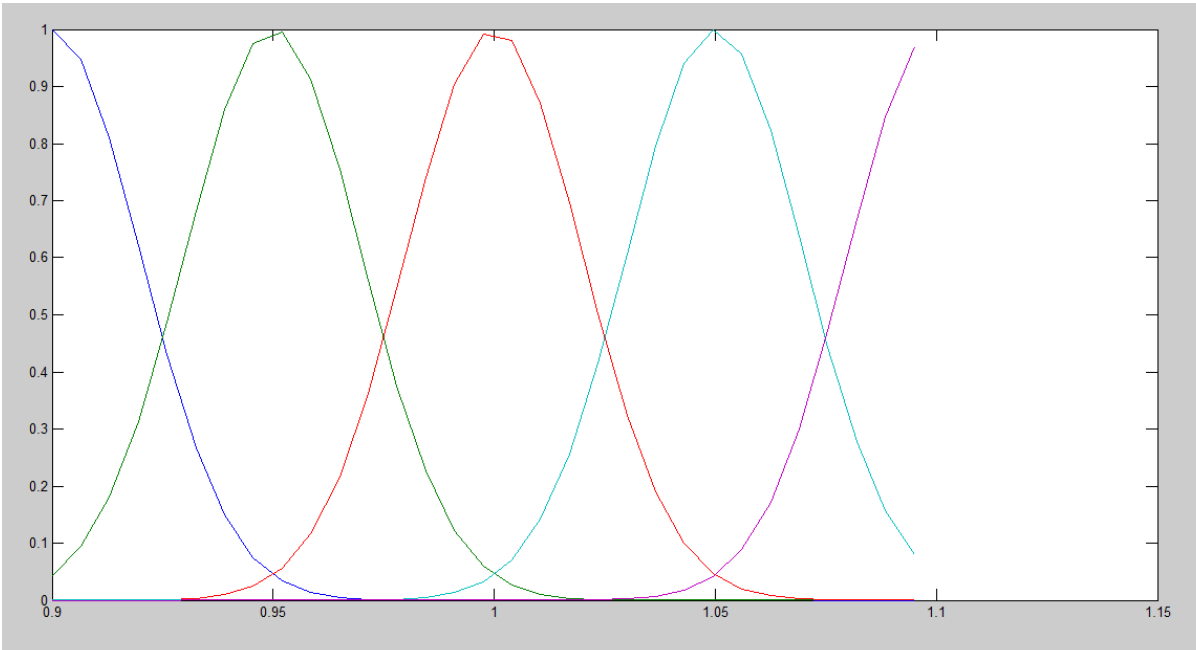


Figure 3Simulating The One Input-One Output Type I Fuzzy System's Five Gaussian Output Functions in Matlab

In accordance with chapter 3, after naming the input and output functions of figures 41 and 42 as A1, A2, A3, A4, A5 and B1, B2, B3, B4, B5, the general rules of the system for calculating the strength of each rule in table 23 were considered by categorizing the rules with the same antecedent and different consequents as follows:

Table 5The Rules of the One Input-One Output Type I Fuzzy System

Set #	Rules
1	If X is A1 then y is B1
	If X is A1 then y is B2
	If X is A1 then y is B3
	If X is A1 then y is B4
	If X is A1 then y is B5
2	If X is A2then y is B1
	If X is A2then y is B2
	If X is A2 then y is B3
	If X is A2then y is B4
	If X is A2 then y is B5
3	If X is A3 then y is B1
	If X is A3 then y is B2
	If X is A3 then y is B3
	If X is A3 then y is B4

	If X is A3 then y is B5
4	If X is A4 then y is B1
	If X is A4 then y is B2
	If X is A4 then y is B3
	If X is A4 then y is B4
	If X is A4 then y is B5
5	If X is A5 then y is B1
	If X is A5 then y is B2
	If X is A5 then y is B3
	If X is A5 then y is B4
	If X is A5 then y is B5

After specifying the rules used in the system according to table 24, the strength of each rule should be determined at this stage according to the lookup table method's guidelines explained extensively in chapters 2 and 3. They are calculated through multiplying the membership degree of each input membership function per its different (μ_x) input values by the membership degree of each output function per its various (μ_y) output values. After defining input and output membership functions, the programming structure of these operations in Matlab are as follows:

Table 6The Programming Structure for Determining the Strength of Various Rules in the One Input One-output Type I Fuzzy System

Program Structure	Descriptions
x=1:1:32;	Input Range
A1=exp(-0.5*((x-0)/2.5).^2);	Gaussian MFs for total Input range
A2=exp(-0.5*((x-8)/2.5).^2);	
A3=exp(-0.5*((x-16)/2.5).^2);	
A4=exp(-0.5*((x-24)/2.5).^2);	
A5=exp(-0.5*((x-32)/2.5).^2);	
y=0.9:0.0065:1.1;	Output Range
B1=exp(-0.5*((y-0.9)/0.02).^2);	Gaussian MFs for total Output range
B2=exp(-0.5*((y-0.95)/0.02).^2);	
B3=exp(-0.5*((y-0.1)/0.02).^2);	
B4=exp(-0.5*((y-1.05)/0.02).^2);	
B5=exp(-0.5*((y-1.1)/0.02).^2);	
plot(y,B1,y,B2,y,B3,y,B4,y,B5)	Illustrating of Output
plot(x,A1,x,A2,x,A3,x,A4,x,A5)	Input Illustrations
for i=1:1:32	Power Calculations w=1:1:25 ;m=1:1:5; n=1:1:5
Power(Rulew(i))=Am(i)*Bn(i)	
end	

The strength of various rules according to table 24 in the Matlab programming interface, e.g. the first rule, is shown in figure 43:

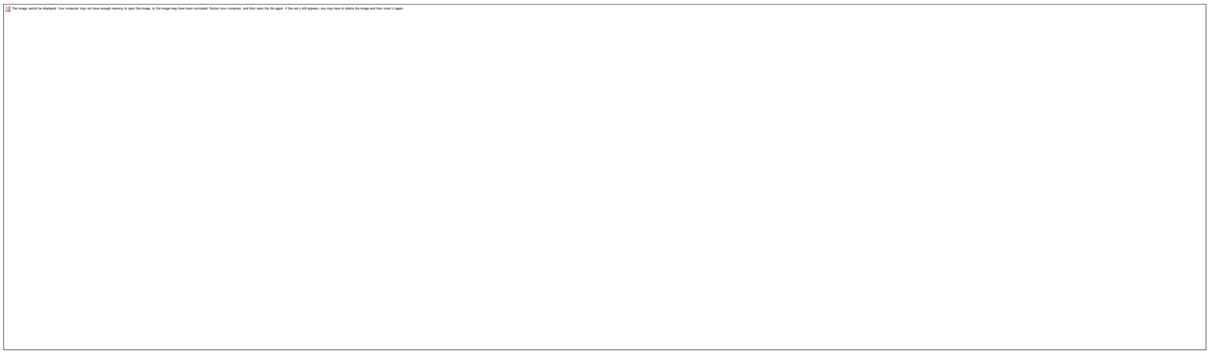


Figure 4Programming the One Input-One Output Type I Fuzzy System's Various Rules in Matlab

After executing the command shown in Figure 43, the strength of various rules will be calculated as shown in table 25:

Table 7: The Strength of Various Rules in the One Input one output Type I Fuzzy System

Set #	Rules	Power
1	If X is A1 then y is B1	2.2554
	If X is A1 then y is B2	0.4125
	If X is A1 then y is B3	9.4992e-004
	If X is A1 then y is B4	4.7647e-008
	If X is A1 then y is B5	5.5337e-014
2	If X is A2 then y is B1	1.0161
	If X is A2 then y is B2	4.7899
	If X is A2 then y is B3	0.5196
	If X is A2 then y is B4	0.0013
	If X is A2 then y is B5	6107e-008
3	If X is A3 then y is B1	0.0038
	If X is A3 then y is B2	0.8895
	If X is A3 then y is B3	4.8407
	If X is A3 then y is B4	0.6105
	If X is A3 then y is B5	0.0018
4	If X is A4 then y is B1	2.3904e-007
	If X is A4 then y is B2	0.0028
	If X is A4 then y is B3	0.7686
	If X is A4 then y is B4	4.8624
	If X is A4 then y is B5	0.7052
5	If X is A5 then y is B1	2.5709e-013
	If X is A5 then y is B2	1.5192e-007
	If X is A5 then y is B3	0.0021

	If X is A5 then y is B4	0.6050
	If X is A5 then y is B5	2.0186

In order to simulate the one input-one output type I fuzzy system according to the lookup table method, the rules with the highest strength are selected from each category according to table 25 and 26:

Table 8The Rules Selected for the One Input one output Common search Type I Fuzzy System

Set #	Rules	Power
1	If X is A1 then y is B1	2.2554
2	If X is A2 then y is B2	4.7899
3	If X is A3 then y is B3	4.8407
4	If X is A4 then y is B4	4.8624
5	If X is A5 then y is B5	2.0186

Then, the system's optimal and intended output are calculated according to the 5 rules extracted using the normal lookup table as shown in table 26, so as to compare them with the system's current output presented in table 21 and calculate the modeled system's error. To this end, Matlab is launched and the term 'Fuzzy' is typed in the Command Window to use the fuzzy toolbox, after which a window titled FIS editor will open as shown in figure 4.4:

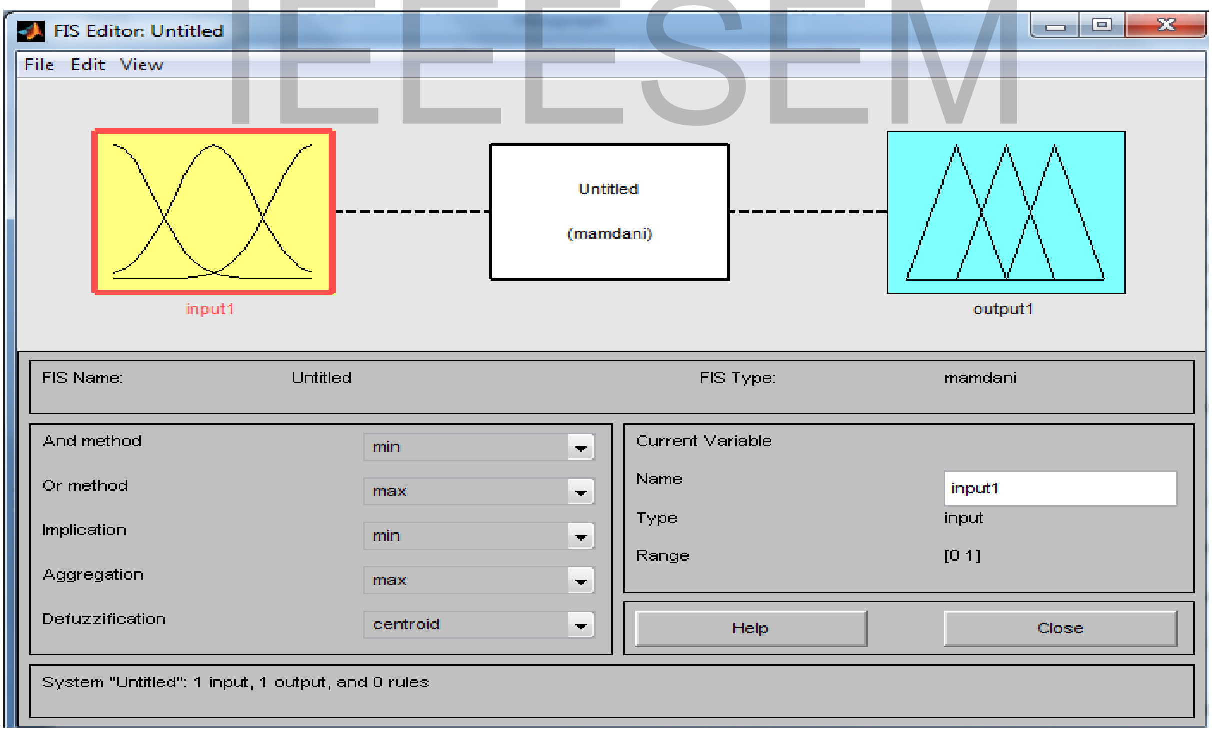


Figure 5The FIS Editor Window's Structure in Matlab

In the window shown in figure 44, system type, i.e. Mamdani, is selected from File and the FIS Editor, and the type of operation in this system is selected according to the rules in table 25 below the same window as min for the AND method and centroid for Defuzzification. In the same page, clicking on input1 will open a new window shown in figure 45:

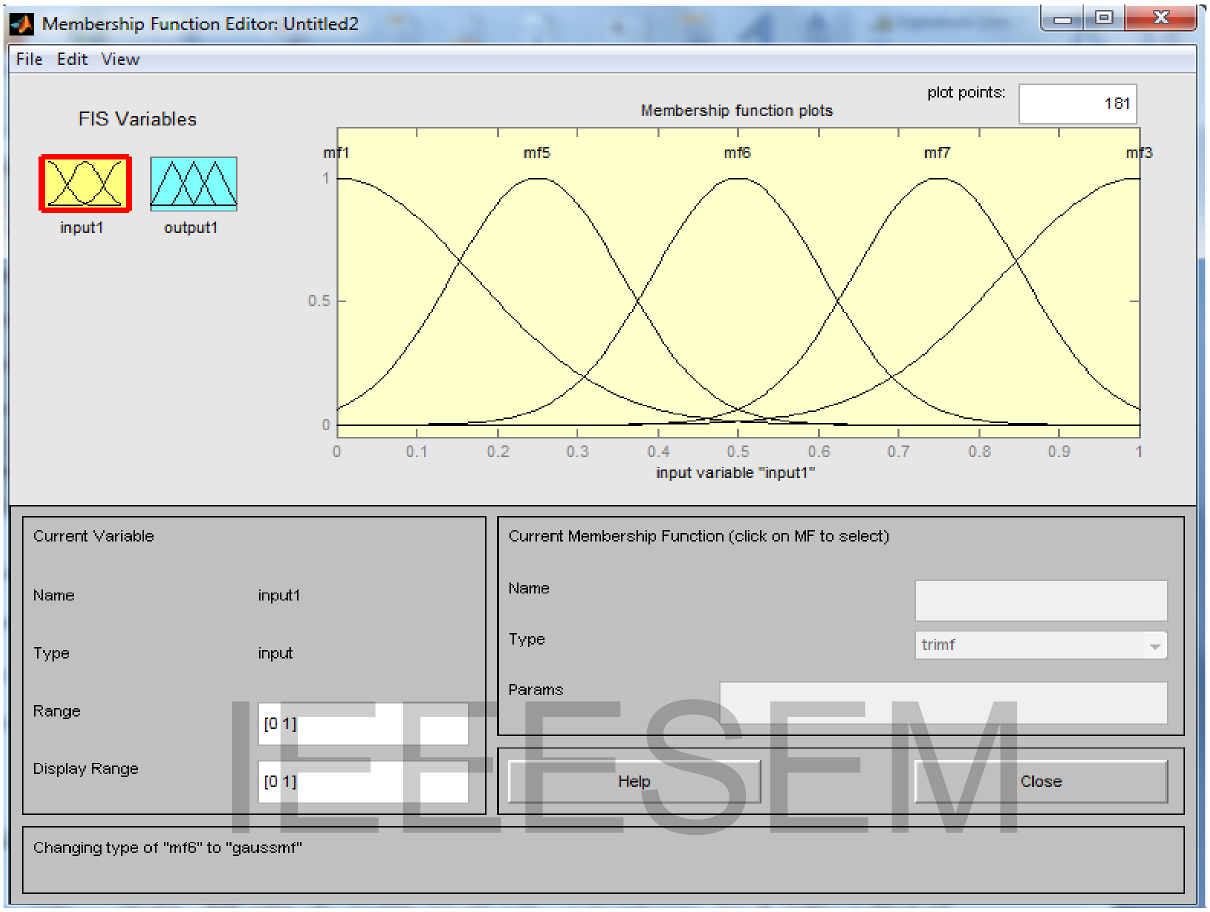


Figure 6The FIS Editor’s Input Selection Structure in Matlab

In the same page, clicking on output1 will open a new window shown in figure 46:

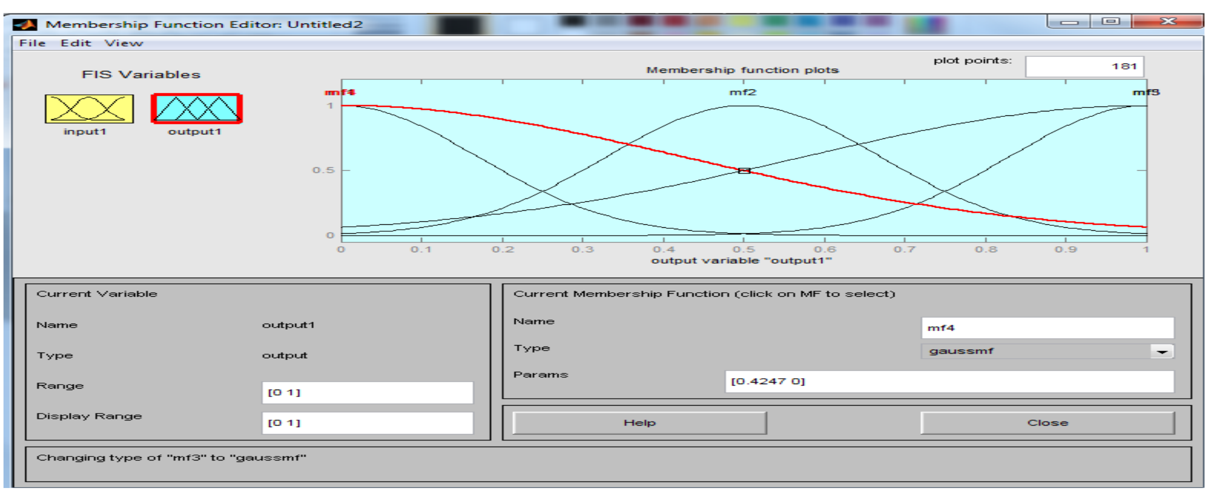


Figure 7The FIS Editor’s Output Selection Structure in Matlab

The window shown in figure 46 is used for selecting input membership functions, and the input membership function with 5 Gaussian structures in the x=1:1:32 interval are simulated and its structure is shown in figure 47:

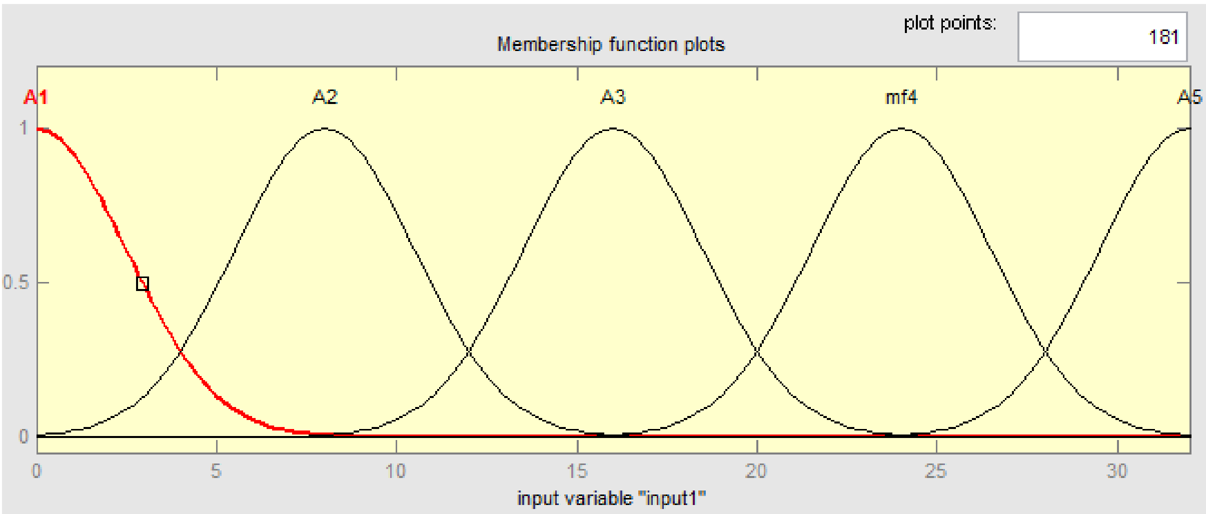


Figure 8The Structure of the One Input-One Output Type I Fuzzy System's 5 Gaussian Input Membership Functions in Matlab

Similarly, simulating the output membership function structure that consists of 5 Gaussian structures in the y=0.9:0.0065:1.1 interval is shown in figure 48:

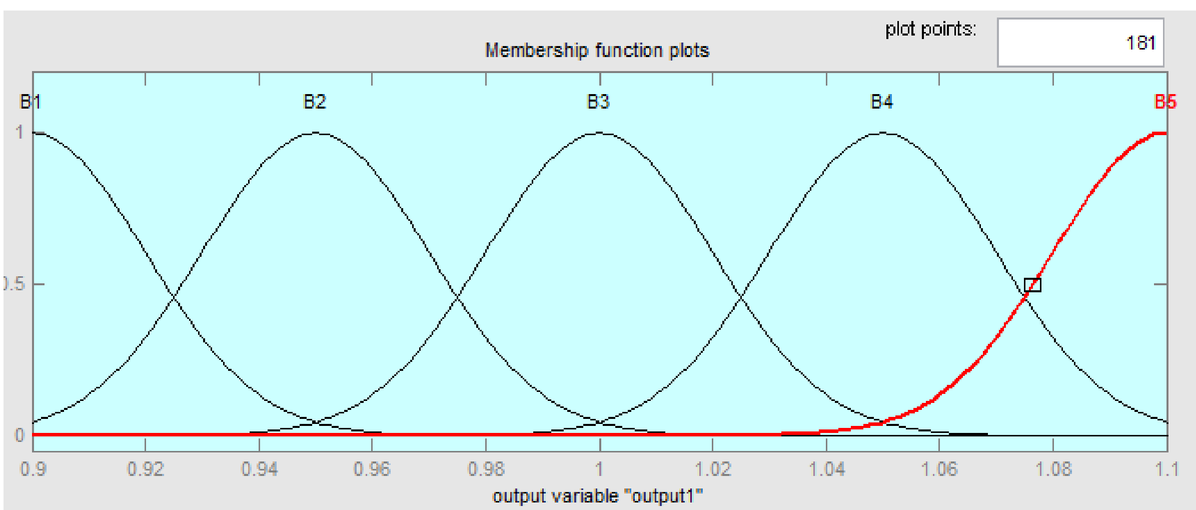


Figure 9The Structure of the One Input-One Output Type I Fuzzy System's 5 Gaussian Output Membership Functions in Matlab

After determining the input and output membership functions in Matlab's FIS Editor, the superior rules selected using the lookup table method should be applied to the system. To this end, the rules shown in table 48 should be created in the Rule section of the Membership Function Editor in the Edit menu. The structure of the rules created is shown in figure 49:

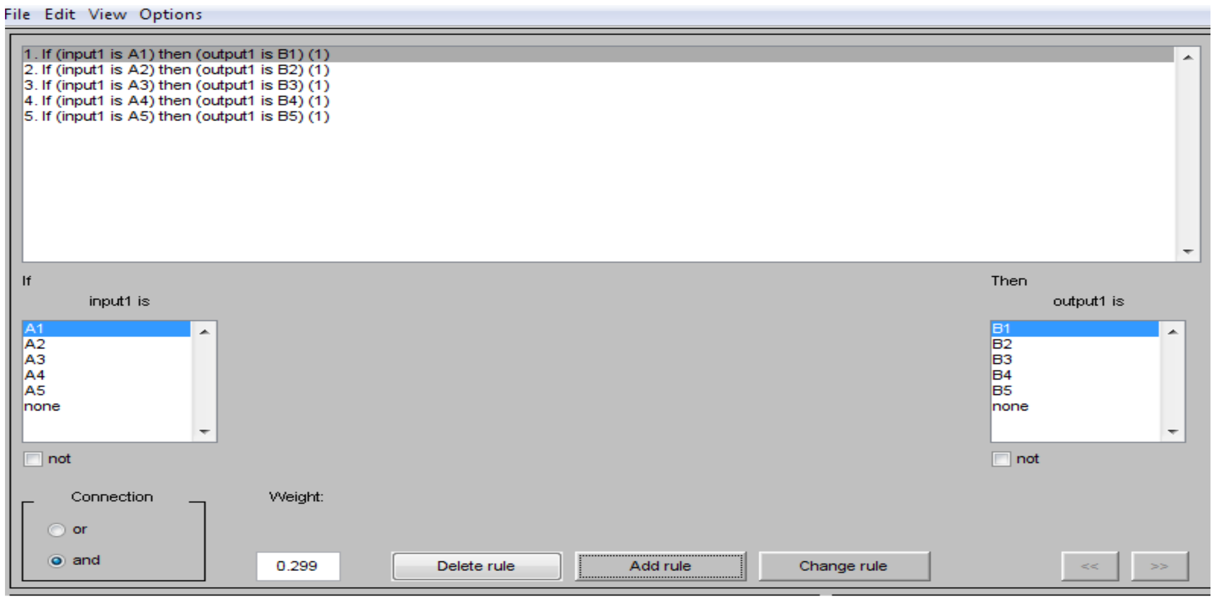


Figure 10Creating the One Input-One Output Type I Fuzzy System's Rules Using the Lookup Table Method in Matlab

After creating the input and output membership functions and applying the rules to the system, the system's output structure for various input values are shown in the View section as shown in figure 50:

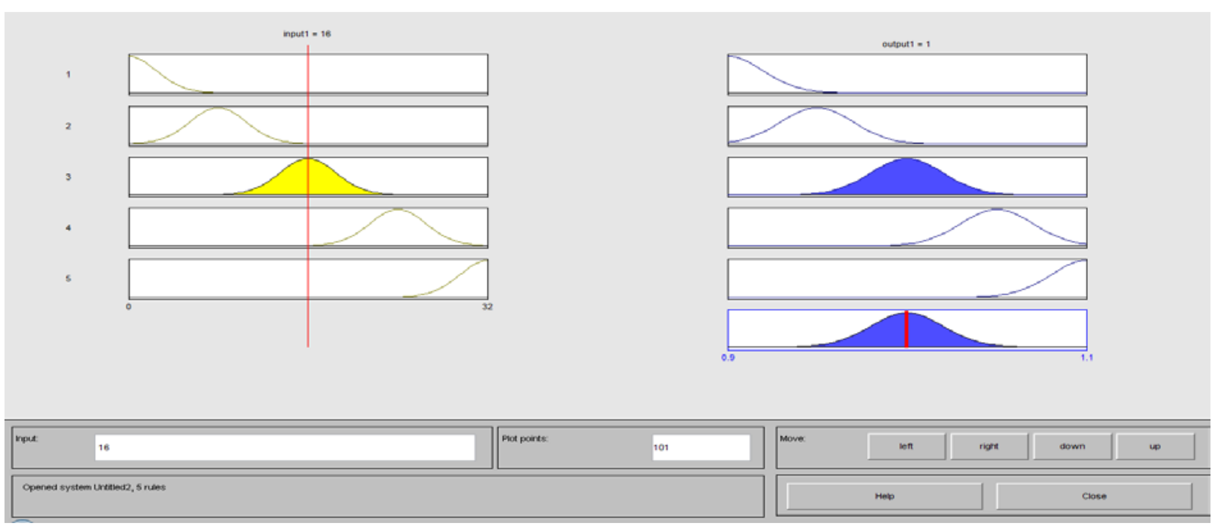


Figure 11The One Input-One Output Type I Fuzzy System's Output Using the Normal Lookup Table Method in Matlab

This system's output results for various input values are shown in table 27 using the evalfis commandafter saving the process and recalling it using the w=readfis ('extracted file save address):

Table 9One Input one out put base on search common method Type I Fuzzy

Input	Formula Output	Simulation Output	Input	Formula Output	Simulation Output
X	y=1+0.1*sin(x)	y=simulation	X	y=1+0.1*sin(x)	y=simulation
1	1.0841	0.9183	17	0.9039	1.0012
2	1.0909	0.9236	18	0.9249	1.0042
3	1.0141	0.9337	19	1.0150	1.0113
4	0.9243	0.9458	20	1.0913	1.0244
5	0.9041	0.9496	21	1.0837	1.0376
6	0.9721	0.9504	22	0.9991	1.0449
7	1.0657	0.9505	23	0.9154	1.0480
8	1.0989	0.9509	24	0.9094	1.0491
9	1.0412	0.9520	25	0.9868	1.0495
10	0.9456	0.9551	26	1.0763	1.0496
11	0.9000	0.9624	27	1.0956	1.0504
12	0.9463	0.9756	28	1.0271	1.0542
13	1.0420	0.9887	29	0.9336	1.0663
14	1.0991	0.9958	30	0.9012	1.0764
15	1.0650	0.9988	31	0.9596	1.0817
16	0.9712	1.0000	32	1.0551	1.0838

Given this circuit's simulation output, the specific curve of input changes for the resulting output is shown in figure 51:

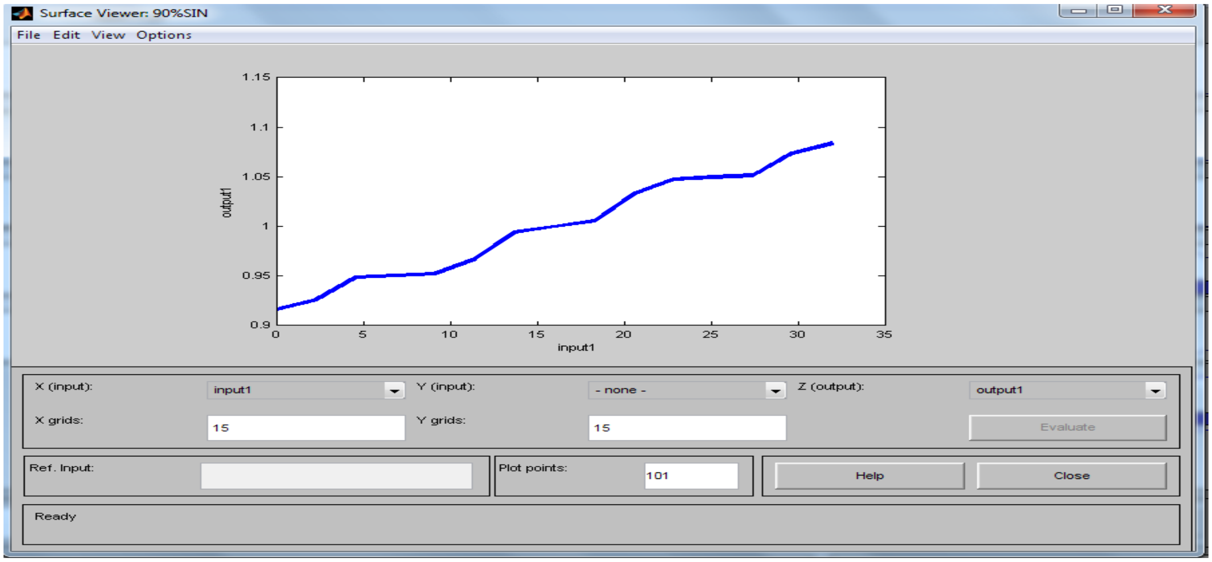


Figure 12The One Input-One Output Type I Fuzzy System's Specific Input Variance Curve According to Output Using the Normal Lookup Table Method in Matlab

Given the output from the modeled fuzzy system and the initial output shown in table 27, the system error can be obtained according to equation (25):

(25) $$\text{Error}=\frac{\sqrt{(y_{Real}-y_{Result})^2}}{n}\%$$

In this equation, y_{Real} is the initial value of the y=1+0.1*sin(x) output, y_{Result} are the values obtained by modeling this fuzzy system according to the lookup table method, and n represents the number of data considered:

After these calculation, the one input-one output type I fuzzy system's error according to the normal lookup table method is %1.6, which means that the modeled system approximates the intended output with an accuracy of over 98.4%.

4.2 Simulating the One Input-One Output Type I Fuzzy System Using the Normal Lookup Table Method in Matlab

As explained in Chapter 3, this structure's overall design and modeling process is identical to the design of a one input-one output type I fuzzy system using the normal lookup table method, and the only difference is in applying system rules after deriving and calculating the strength of each rule. In order to improve the system's accuracy in output modeling, unlike the normal search table method which only selects one rule from every set with the same antecedents and different consequents, over 90% of superior rules in each category are used to calculate the system output. Hence, this system's modeling stages are quite similar to simulating a one input-one output type I fuzzy system using the normal search table method, including programming and simulating the input and output membership functions as well as the strength of different rules, while 90% of superior rules should be selected from each set and normalized in the rule selection stage.

Following table 25 which considered the computed strength of each rule, in this stage, the rules in each set are normalized according to equation (26):

(26) $\frac{\textbf{Power of each Rule}}{\textbf{Mix Rule Power}}$= the equation for normalizing rules in each set

Example:.$\frac{\mathbf{2.2554}}{\mathbf{2.2554}} = 1$ = normalizing R1 in the first set

$\frac{\mathbf{0.4125}}{\mathbf{2.2554}} = 0.183$ = normalizing R2 in the first set

Likewise, the other rules of various sets are normalized and the results are shown in table 28:

Table 10Normalizing the Strengths of Various Rules in the One Input-One Output Type I Fuzzy System with the Optimal Lookup Table Method

Set #	Rules	Power	Normalization
1	R1: If X is A1 then y is B1	2.2554	1
	R2:If X is A1 then y is B2	0.4125	0,183
	R3:If X is A1 then y is B3	0	0
	R4:If X is A1 then y is B4	0	0
	R5:If X is A1 then y is B5	0	0
2	R6:If X is A2 then y is B1	1.0161	0,212
	R7:If X is A2 then y is B2	4.7899	1
	R8:If X is A2 then y is B3	0.5196	0,108
	R9:If X is A2 then y is B4	0.0013	0,0002
	R10:If X is A2 then y is B5	0	0
3	R11:If X is A3 then y is B1	0.0038	0,0007
	R12:If X is A3 then y is B2	0.8895	0,183
	R13:If X is A3 then y is B3	4.8407	1
	R14:If X is A3 then y is B4	0.6105	0,126
	R15:If X is A3 then y is B5	0.0018	0,00037
4	R16:If X is A4 then y is B1	0	0
	R17:If X is A4 then y is B2	0.0028	0,00057
	R18:If X is A4 then y is B3	0.7686	0,158
	R19:If X is A4 then y is B4	4.8624	1

	R20:If X is A4 then y is B5	0.7052	0,145
5	**R21:If X is A5 then y is B1**	0	0
	R22:If X is A5 then y is B2	0	0
	R23:If X is A5 then y is B3	0.0021	0,001
	R24:If X is A5 then y is B4	0.6050	0,299
	R25:If X is A5 then y is B5	2.0186	1

In order to simulate the system, according to table 28 which includes the strength of different rules as well as their normalization process, 90% of superior rules should be determined for simulation according to table 29:

Table 11The Rules Selected for the One Input one output base on optimal search Type I Fuzzy System

Set #	**Rules**	**Power**	**Normalization**	**75% 0f Rules**
1	**R1: If X is A1 then y is B1**	2.2554	1	*
	R2:If X is A1 then y is B2	0.4125	0,183	*
2	**R6:If X is A2 then y is B1**	1.0161	0,212	*
	R7:If X is A2 then y is B2	4.7899	1	*
	R8:If X is A2 then y is B3	0.5196	0,108	*
3	**R12:If X is A3 then y is B2**	0.8895	0,183	*
	R13:If X is A3 then y is B3	4.8407	1	*
	R14:If X is A3 then y is B4	0.6105	0,126	*
4	**R18:If X is A4 then y is B3**	0.7686	0,158	*
	R19:If X is A4 then y is B4	4.8624	1	*
	R20:If X is A4 then y is B5	0.7052	0,145	*
5	**R24:If X is A5 then y is B4**	0.6050	0,299	*
	R25:If X is A5 then y is B5	2.0186	1	*

After specifying the input and output membership functions in the Matlab FIS Editor similarly to the previous state, the 13 rules obtained using the optimal lookup table should be applied to the system. To this end, the superior rules should be applied to the system in Edit and the Membership Function Editor according to their weights obtained after normalization as shown in table 29: The rule generation structure is shown in figure 52:

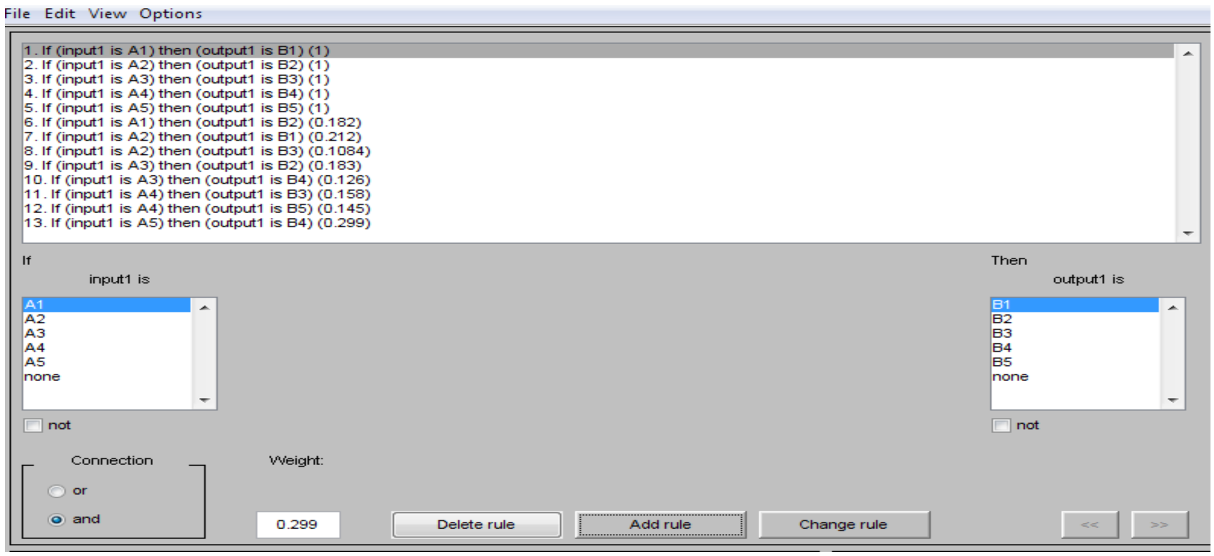


Figure 13Creating the One Input-One Output Type I Fuzzy System's Select Rules Using the Optimal Lookup Table Method in Matlab

After creating the input and output membership functions and applying select rules to the one input-one output type I fuzzy system using the optimal lookup table method, the system's output structure for various input values can be seen in the View section as shown in figure 53:

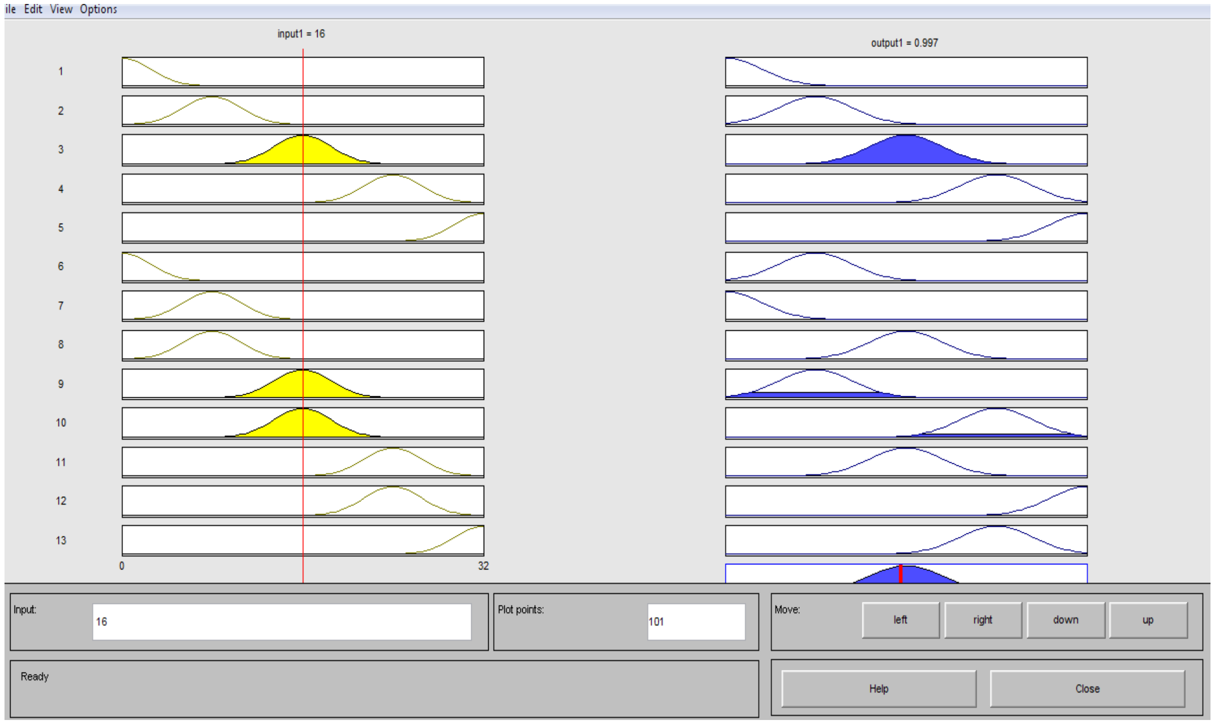


Figure 14 The One Input-One Output Type I Fuzzy System's Output Using the Normal Lookup Table Method in Matlab

Similarly, this system's output results for various input values using the evalfis command and recalling it using w=readfis ('extracted file save address) are shown in table 30 :

Table 12The One Input-One Output Type I Fuzzy System's General Output According to the Optimized Lookup Table Method

Input	Formula Output	Simulation Output	Input	Formula Output	Simulation Output
X	y=1+0.1*sin(x)	y=simulation	X	y=1+0.1*sin(x)	y=simulation
1	1.0841	0.9298	17	0.9039	0.9973
2	1.0909	0.9306	18	0.9249	0.9975
3	1.0141	0.9367	19	1.0150	1.0034
4	0.9243	0.9507	20	1.0913	1.0185
5	0.9041	0.9553	21	1.0837	1.0350
6	0.9721	0.9558	22	0.9991	1.0406
7	1.0657	0.9560	23	0.9154	1.0407
8	1.0989	0.9562	24	0.9094	1.0405
9	1.0412	0.9561	25	0.9868	1.0409
10	0.9456	0.9564	26	1.0763	1.0415

11	0.9000	0.9637	27	1.0956	1.0427
12	0.9463	0.9795	28	1.0271	1.0474
13	1.0420	0.9943	29	0.9336	1.0615
14	1.0991	0.9975	30	0.9012	1.0650
15	1.0650	0.9973	31	0.9596	1.0660
16	0.9712	0.9972	32	1.0551	1.0660

Given this circuit's simulation output, the specific curve of input variances for the resulting output is shown in figure 54:

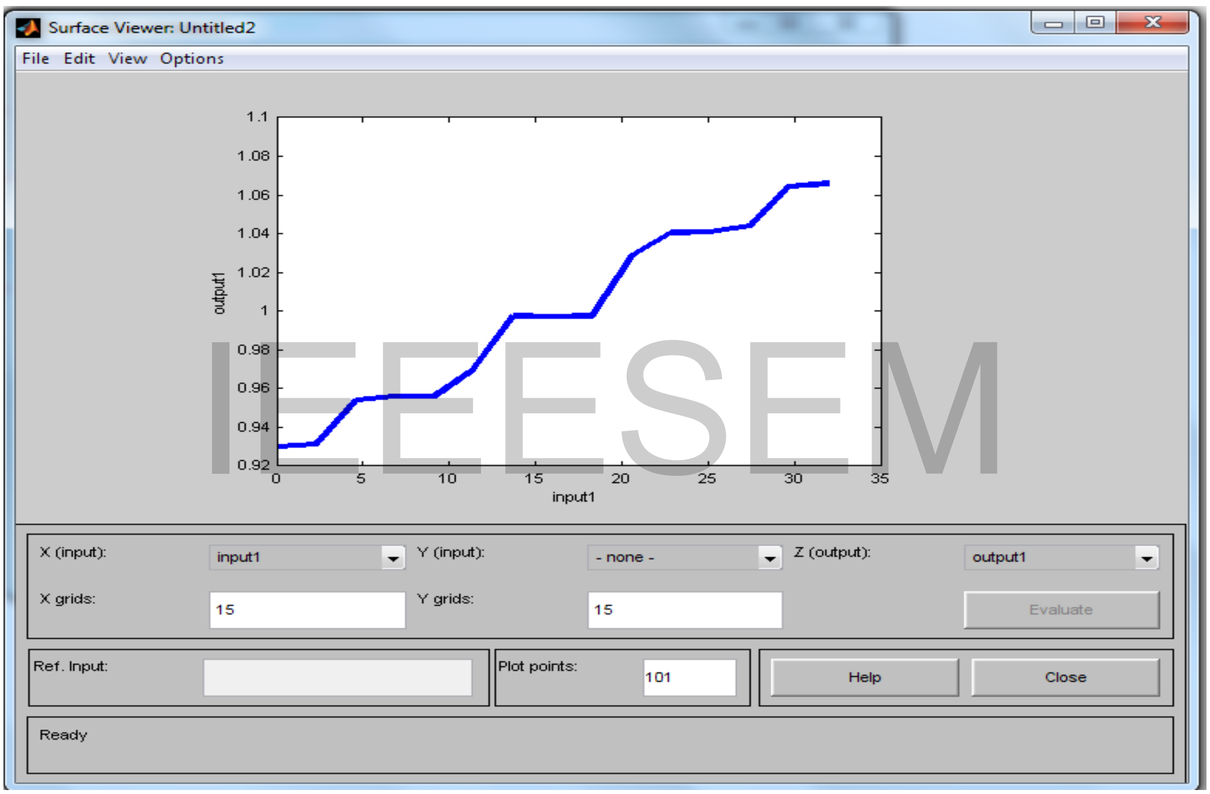


Figure 15The One Input-One Output Type I Fuzzy System's Specific Input Variance Curve According to Output Using the Normal Lookup Table Method in Matlab

Given the output from the modeled fuzzy system and the initial output shown in table 31, the system error can be obtained according to equation (27):

Table 13 T1two-inputone - out fuzzy Model

T1 Two-Input One-Output Fuzzy Model

#	System Inputs		Output System	#	System Inputs		System Output
	input 1	input 2	output		input 1	input 2	output
1	10	2	10	**50**	50,327	16,063	55,57
2	10.823	2,287	10,93	**51**	51,15	16,35	56,5
3	11,646	2,574	11,86	**52**	51,973	16,637	57,43
4	12,469	2,861	12,79	**53**	52,796	16,924	58,36
5	13,292	3,148	13,72	**54**	53,619	17,211	59,29
6	14,115	3,435	14,65	**55**	54,442	17,498	60,22
7	14,938	3,722	15,58	**56**	55,265	17,785	61,15
8	15,761	4,009	16,51	**57**	56,088	18,072	62,08
9	16,584	4,296	17,44	**58**	56,911	18,359	63,01
10	17,407	4,583	18,37	**59**	57,734	18,646	63,94
11	18,23	4,87	19,3	**60**	58,557	18,933	64,87
12	19,053	5,157	20,23	**61**	59,38	19,22	65,8
13	19,876	5,444	21,16	**62**	60,203	19,507	66,73
14	20,699	5,731	22,09	**63**	61,026	19,794	67,66
15	21,522	6,018	23,02	**64**	61,849	20,081	68,59
16	22,345	6,305	23,95	**65**	62,672	20,368	69,52
17	23,168	6,592	24,88	**66**	63,495	20,655	70,45
18	23,991	6,879	25,81	**67**	64,318	20,942	71,38
19	24,814	7,166	26,74	**68**	65,141	21,229	72,31
20	25,637	7,453	27,67	**69**	65,964	21,516	73,24
21	26,46	7,74	28,6	**70**	66,787	21,803	74,17
22	27,283	8,027	29,53	**71**	67,61	22,09	75,1
23	28,106	8,314	30,46	**72**	68,433	22,377	76,03
24	28,929	8,601	31,39	**73**	69,256	22,664	76,96
25	29,752	8,888	32,32	**74**	70,079	22,951	77,89
26	30,575	9,175	33,25	**75**	70,902	23,238	78,82
27	31,398	9,462	34,18	**76**	71,725	23,525	79,75
28	32,221	9,749	35,11	**77**	72,548	23,812	80,68
29	33,044	10,036	36,04	**78**	73,371	24,099	81,61
30	33,867	10,323	36,97	**79**	74,194	24,386	82,54

31	34,69	10,61	37,9	**80**	75,017	24,673	83,47
32	35,513	10,897	38,83	**81**	75,84	24,96	84,4
33	36,336	11,184	39,76	**82**	76,663	25,247	85,33
34	37,159	11,471	40,69	**83**	77,486	25,534	86,26
35	37,982	11,758	41,62	**84**	78,309	25,821	87,19
36	38,805	12,045	42,55	**85**	79,132	26,108	88,12
37	39,628	12,332	43,48	**86**	79,955	26,395	89,05
38	40,451	12,619	44,41	**87**	80,778	26,682	89,98
39	41,274	12,906	45,34	**88**	81,601	26,969	90,91
40	42,097	13,193	46,27	**89**	82,424	27,256	91,84
41	42,92	13,48	47,2	**90**	83,247	27,543	92,77
42	43,743	13,767	48,13	**91**	84,07	27,83	93,7
43	44,566	14,054	49,06	**92**	84,893	28,117	94,63
44	45,389	14,341	49,99	**93**	85,716	28,404	95,56
45	46,212	14,628	50,92	**94**	86,539	28,691	96,49
46	47,035	14,915	51,85	**95**	87,362	28,978	97,42
47	47,858	15,202	52,78	**96**	88,185	29,265	98,35
48	48,681	15,489	53,71	**97**	89,008	29,552	99,28
49	49,504	15,776	54,64	**98**	89,831	29,839	100,21

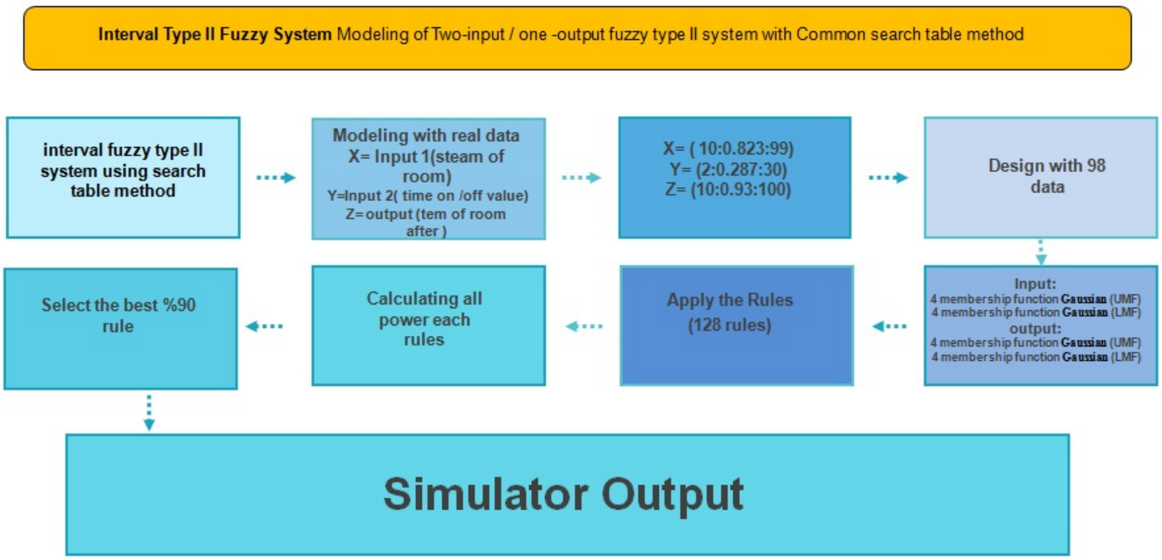
(27) Error=$\frac{\sqrt{(y_{Real} - y_{Result})^2}}{n} = \frac{\sqrt[2]{0.24023833}}{32} = \frac{0.490141136}{32} = 0.014763$

Error $= (0.0014763) * 100 = 1.4\%$

After these calculation, the one input-one output type I fuzzy system's error according to the normal lookup table method is 1.4, which means that the modeled system approximates the intended output with an accuracy of over 98.4%.

3- **Results:**
4- 5.2.3 Modeling Results for the Two Input-One Output Interval Type II Fuzzy System Using the Normal Lookup Table Method
5-
6- Similarly to the “two input-one output type I fuzzy system modeling using the normal lookup table” section of this chapter, this section is dedicated for modeling the fuzzy system in

Matlabusing real data measured from the steam chamber control process. The structure considered for modeling is shown in figure 94:



- 7

8- Figure94The Block Diagram of the One Input-One Output Interval Type II Fuzzy System Modeling Using the Normal Lookup Table Method

9-

10- The simulation results for this structure are shown in table 76:

11- Table 14Modeling Results for the One Input-One Output Interval Type II Fuzzy System Using the Normal Lookup Table Method

	Type I I fuzzy two-input-one-output fuzzy system
Design Method	**Search table (Commom)**
data type	**real data**
Number of data	**98 data**
Type of input and output membership functions	**Gaussian functions**
Input1 Gaussian Functions(UMF)	**4 Number**
Input 2Gaussian Functions (LMF)	**4 Number**

Output Gaussian Functions (UMF)	**4 Number**
Output Gaussian Functions (LMF)	**4 Number**
Total number of rules	**128 law**
Number of rules applied to the system	**16 law**
System error rate	**3%**
System accuracy	**97%**
Simulation software	**Matlab**

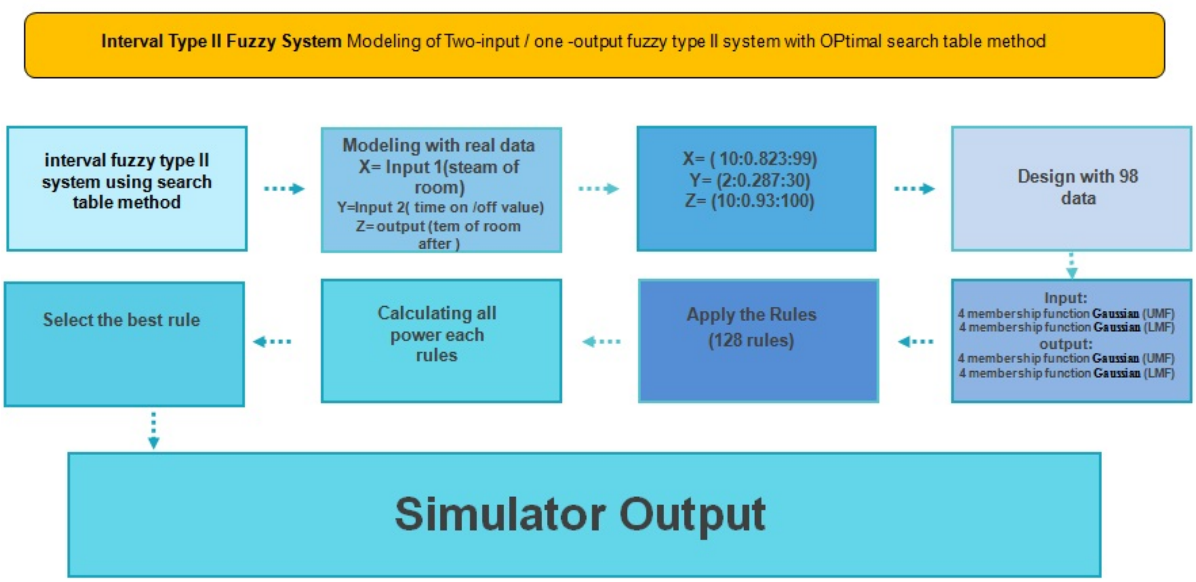
12-

13-

14- 5.2.4 Modeling Results for the Two Input-One Output Interval Type II Fuzzy System Using the Optimal Lookup Table Method

15-

16- Similarly to the previous structure, the structure considered for modeling is shown in figure 95:



\- 17

18- Figure95The Block Diagram of Two Input-One Output Interval Type II Fuzzy System Modeling Using the Optimal Lookup Table Method

19-

20- The simulation results for this structure are shown in table 7:

21- Table 15Simulation Results for the One Input-One Output Interval Type II Fuzzy System Using the Optimal Lookup Table Method

	Type I I fuzzy two-input-one-output fuzzy system
Design Method	**Search table (Optimal)**
data type	**real data**(steam room data)
Number of data	**98 data**
Type of input and output membership functions	**Gaussian functions**
Input1 Gaussian Functions(UMF)	**4 Number**
Input 2Gaussian Functions (LMF)	**4 Number**
Output Gaussian Functions (UMF)	**4 Number**
Output Gaussian Functions (LMF)	**4 Number**
Total number of rules	**128 law**
Number of rules applied to the system	**42 law**
System error rate	**10%**
System accuracy	**90%**
Simulation software	**Matlab**

4- Conclusion

For the first time, this research presents and structurally simulates type I and interval type II fuzzy systems using the lookup table and the concept of uncertainty avoidance. Regarding the proposed type I system, the common design method of using the lookup table, selecting only one superior rule in each category with the same antecedent and different consequent, and a new method based on choosing more rules according to their strength was used and normalized to increase the proposed type I fuzzy system's output accuracy. Furthermore, the lookup table method and the uncertainty avoidance formula were used for the first time in an interval type II fuzzy system with real data to model steam chamber temperature, and the output approximates and simulates the real output with a reasonable accuracy. All proposed structures were simulated in MATLAB and the results are presented in detail along with the programming code.

References:

1. Aminifar, S., &Marzuki, A. (2013). Uncertainty in interval type-2 fuzzy systems. Mathematical Problems in Engineering, 2013.
2. Aminifar, S. (2014). Design and Implementation of Fuzzy Controllers for Handling Uncertainty in an Industrial Application (Doctoral dissertation, UniversitiSains Malaysia).
3. Aminifar, S., &Marzuki, A. B. (2012). Voltage-mode fuzzy logic controller. *Indian Journal of Science and Technology*, *5*(11), 3630-6.
4. Aminifar, S., Khoei, A., Haidi, K., &Yosefi, G. (2006). A digital CMOS fuzzy logic controller chip using new fuzzifier and max circuit. *AEU-International Journal of Electronics and Communications*, *60*(8), 557-566.
5. Aminifar, S., & bin Marzuki, A. (2012, November). A novel fuzzy controller applicable to steaming room. In *2012 IEEE International Conference on Control System, Computing and Engineering* (pp. 171-176). IEEE.
6. Aminifar, S., Kekshar, S. M., &Daneshwar, M. (2018). Look-up table based fuzzy controller in order to control steaming room process. *RevistaInnovaciencia*, *6*(2), 1-12.
7. Aminifar, S., & bin Marzuki, A. (2013). Horizontal and vertical rule bases method in fuzzy controllers. *Mathematical Problems in Engineering*, *2013*.

8. Aminifar, S., &Yosefi, G. (2007, September). Application of adaptive neuro fuzzy inference system (ANFIS) in implementing of new CMOS fuzzy logic controller (FLC) chip. In *AIP Conference Proceedings* (Vol. 936, No. 1, pp. 49-53). American Institute of Physics.

9. Author links open overlay panelHomeiraShahparastEghbal G.Mansoori. (2019). Developing an online general type-2 fuzzy classifier using evolving type-1 rules. *International Journal of Approximate Reasoning* , 336-353.

10. Biglarbegian, M. (2010). Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments. *IEEE transactions on industrial electronics* , 1371-1384.

11. Bilgin, A. (2013). An experience based linear general type-2 fuzzy logic approach for Computing With Words. *IEEE* .

12. Castillo, O. (2011). Design of interval type-2 fuzzy models through optimal granularity allocation. *Elsevier* , 5590-5601.

13. Castillo, O. (2012). *Type-2 fuzzy logic systems.* Berlin: Springer.

14. Celemin, C. E. (2013). A Proposal to Speed up the Computation of the Centroid of an Interval Type-2 Fuzzy Set. *Hindwi* , 17.

15. Daoyuan Zhai, J. M. (2011). Uncertainty measures for general type-2 fuzzy sets. *Elsevier* , 503-518.

16. Ghaemi, M., Hosseini-Sani, S. K., & Khooban, M. H. (2014). Direct adaptive general type-2 fuzzy control for a class of uncertain non-linear systems. *IET Science, Measurement & Technology* , 518 – 527.

17. Gordon, A. (2012). *Obtaining Partitions of a Set of Hard or Fuzzy*. Roma , itly: Univercity st of Andrewa.

18. Greenfield, S. (2011). Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology. *Advances in Intelligent Systems Research* .

19. Greenfield, S. (2013). The structure of the type-reduced set of a continuous type-2 fuzzy set. *Elsevier* .

20. Hamrawi, H. (2010). Measures of uncertainty for type-2 fuzzy sets. *IEEE* , 1-6.

21. Hamrawi, H. (2011). Type-2 fuzzy alpha-cuts. *De Montfort University* .

22. Hao, J. (2012). Power System Load Forecasting Based on Fuzzy Clustering and Gray Target Theory. *Elsevier* , 1852-1859.

23. Hari, C. V. (2010). Interval Type-2 Fuzzy Logic for Software Cost Estimation Using TSFC with Mean and Standard Deviation. *IEEE* .

24. MORDESON, J. N. (2010). A FUZZY MATHEMATICAL MODEL OF NUCLEAR STABILITY. *New Mathematics and Natural Computation* , 119-140.

25. Sadeghian, A. (2013). *Advances in Type-2 Fuzzy Sets and Systems.* TorontoCanada: Ryerson UniversityTorontoCanada.

26. SarahGreenfield. (2012). The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation. *Elsevier* , 77-92.

27. T Paudel, J. R. (2011). Nanocoax solar cells based on aligned multiwalled carbon nanotube arrays. *physica status solidi (a)* , 924-927.

28. Wagner, C., & Hagras, H. (2010). Toward General Type-2 Fuzzy Logic Systems Based on zSlices. *IEEE* , 637 - 660.

29. Yang, G.-H., & Wang, H. (2010). Fault Detection for a Class of Uncertain State-Feedback Control Systems. *IEEE Transactions on Control Systems Technology* , 201 - 212.

30. Zdenko Kovacic, S. B. (2018). *Fuzzy Controller Design: Theory and Applications.* zagreb,crotia: univercity of crotia.