INTELLIGENT SYSTEMS AND CONTROL
PRACTICAL FILE

EXPERIMENT 1

AIM: To generate triangular, trapezoidal, and sigmoidal membership function using MATLAB tool-box

SOFTWARE USED: MATLAB R2013a

THEORY:

1. Trapezoidal membership function

`y = trapmf(x,params)` returns fuzzy membership values computed using the following trapezoidal membership function

$$f(x;a,b,c,d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$$

To specify the parameters, a, b, c, and d, use `params`.

Membership values are computed for each input value in `x`.

2. Sigmoidal membership function

`y = sigmf(x,params)` returns fuzzy membership values computed using the sigmoidal membership function given by:

$$f(x;a,c) = \frac{1}{1+e^{-a(x-c)}}$$

To specify the a and c parameters, use `params`.

Membership values are computed for each input value in `x`.

3. Triangular membership function

`y = trimf(x,params)` returns fuzzy membership values computed using the following triangular membership function:

$$f(x;a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$

or, more compactly

$$f(x;a,b,c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

To specify the parameters, a, b, and c, use `params`.

Membership values are computed for each input value in `x`.

MATLAB CODE:

1. Trapezoidal membership function

```
x = 0:0.1:10;
y = trapmf(x,[3 4 5 6]);
plot(x,y)
xlabel('trapmf, P = [3 4 5 6]')
ylim([-0.05 1.05])
```

2. Sigmoidal membership function

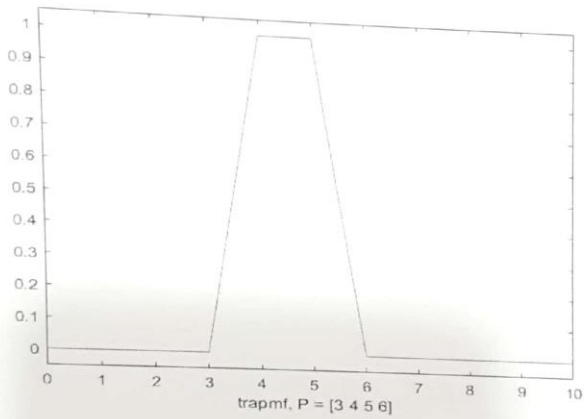
```
x = 0:0.1:10;
y = sigmf(x,[3 6]);
plot(x,y)
xlabel('sigmf, P = [3 6]')
ylim([-0.05 1.05])
```

3. Triangular membership function

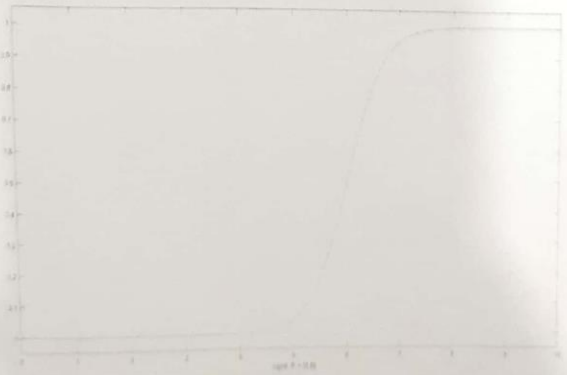
```
x = 0:0.1:10;
y = trimf(x,[2 4 6]);
plot(x,y)
xlabel('trimf, P = [2 4 6]')
ylim([-0.05 1.05])
```

OUTPUT:

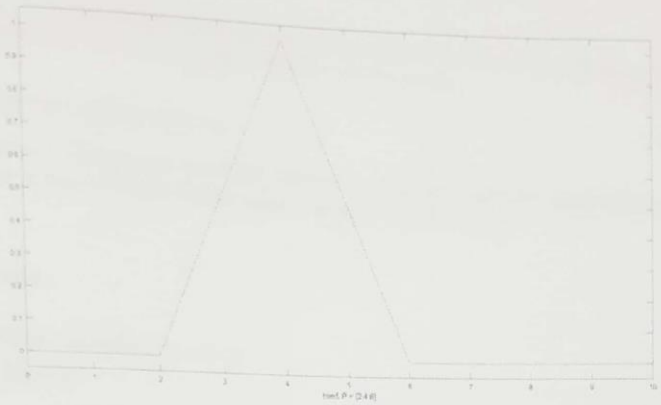
1. Trapezoidal membership function



2. Sigmoidal membership function



3. Triangular membership function



CONCLUSION: The experiment was performed carefully and successfully. Triangular, trapezoidal, and sigmoidal membership function were successfully generated and plotted using MATLAB tool-box

EXPERIMENT 2

AIM: To write a MATLAB program to generate the union and intersection of trapezoidal, triangular and sigmoidal member functions.

SOFTWARE USED: MATLAB R2013a

THEORY:

1. Max function (union)

   `C = max(A,B)` returns an array the same size as `A` and `B` with the largest elements taken from `A` or `B`. The dimensions of `A` and `B` must match, or they may be scalar.

2. Min function (intersection)

   `C = min(A,B)` returns an array the same size as `A` and `B` with the smallest elements taken from `A` or `B`. The dimensions of `A` and `B` must match, or they may be scalar.

MATLAB CODE:

1. Trapezoidal member function
   - Union:

```
x = 0:0.1:10;
y = trapmf(x,[3 5 7 9]);
z = trapmf(x,[2 4 6 8]);
A = max(y,z);
plot(A)
xlabel('Union')
ylim([-0.05 1.05])
```

   - Intersection:

```
x = 0:0.1:10;
y = trapmf(x,[3 5 7 9]);
z = trapmf(x,[2 4 6 8]);
A = min(y,z);
plot(A)
xlabel('Intersection')
ylim([-0.05 1.05])
```

2. Sigmoidal member function:
   - Union:

```
x = 0:0.1:10;
y = sigmf(x,[2 4]);
z = sigmf(x,[3 5]);
A = max(y,z);
plot(A)
xlabel('Union')
ylim([-0.05 1.05])
```

   - Intersection:

```
x = 0:0.1:10;
y = sigmf(x,[2 4]);
z = sigmf(x,[3 5]);
A = min(y,z);
plot(A)
```

```
xlabel('intersection')
ylim([-0.05 1.05])
```

3. Triangular member function:
   - Union:

```
x = 0:0.1:10;
y = trimf(x,[2 4 6]);
z = trimf(x,[3 5 7]);
A = max(y,z);
plot(A)
xlabel('x-axis')
ylim([-0.05 1.05])
```

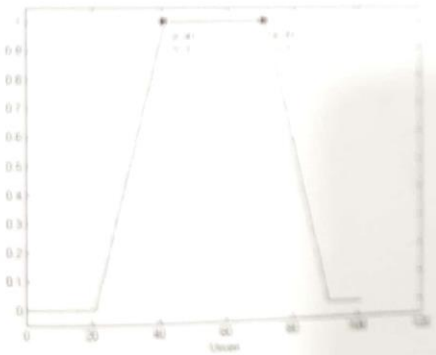
   - Intersection:

```
x = 0:0.1:10;
y = trimf(x,[2 4 6]);
z = trimf(x,[3 5 7]);
A = min(y,z);
plot(A)
xlabel('x-axis')
ylim([-0.05 1.05])
```

OUTPUT:

1. Trapezoidal member function:

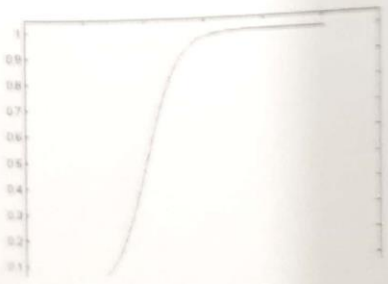
Union:



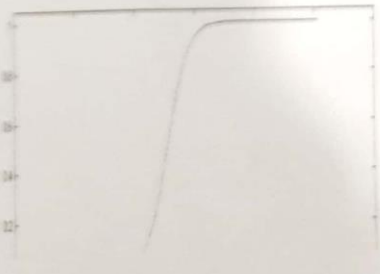
Intersection:

2. Sigmoidal member function:

Union:

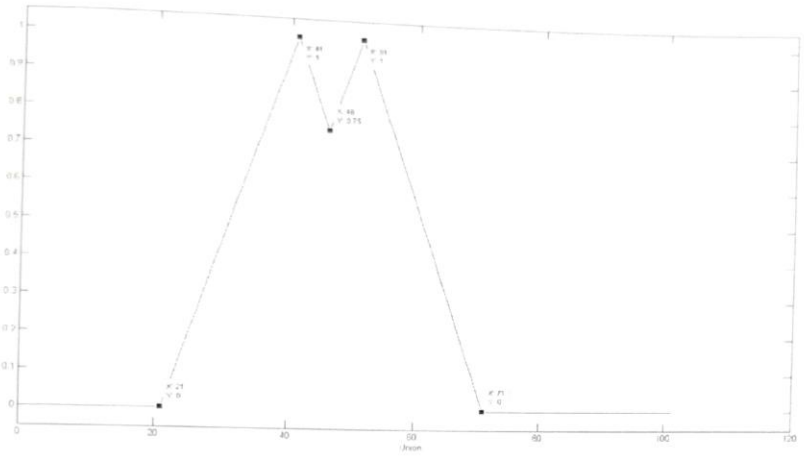


Intersection:

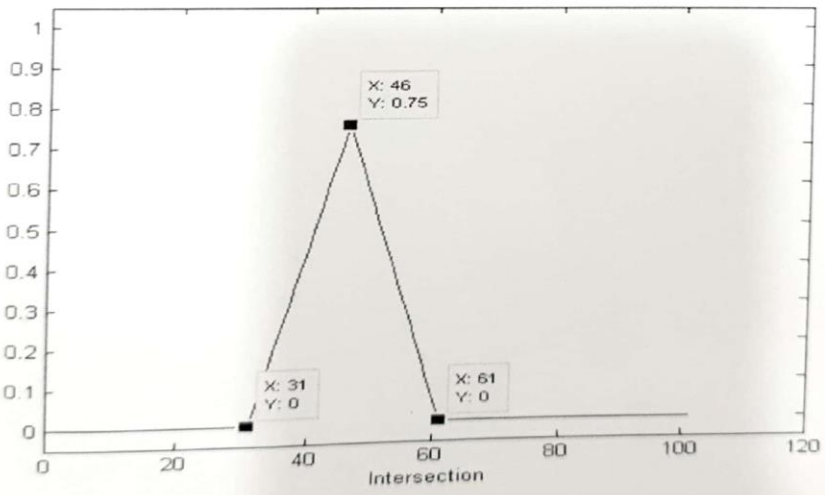


3. Triangular member function:

Union:



Intersection:



EXPERIMENT 3

AIM: To implement fuzzy controller for temperature control system using MATLAB software

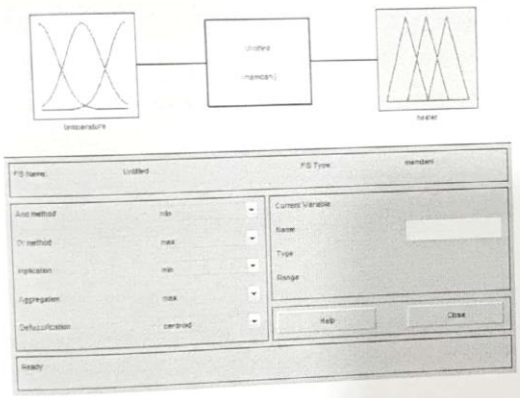
SOFTWARE USED: MATLAB R2013a

THEORY:

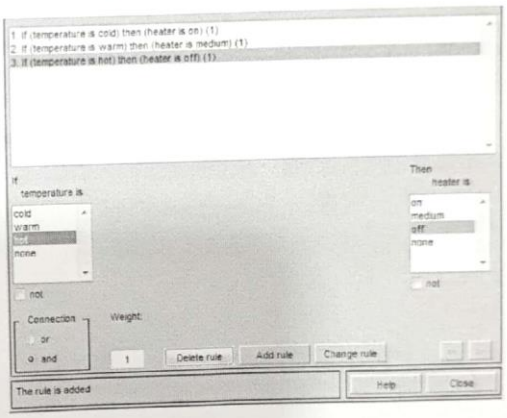
The FIS Editor displays high-level information about a Fuzzy Inference System. At the top is a diagram of the system with each input and output clearly labelled. By double-clicking on the input or output boxes, you can bring up the Membership Function Editor. Double-clicking on the fuzzy rule box in the centre of the diagram will bring up the Rule Editor.

Just below the diagram is a text field that displays the name of the current FIS. In the lower left of the window are a series of popup menus that allow you to specify the various functions used in the fuzzy implication process. In the lower right are fields that provide information about the current variable. The current variable is determined by clicking once on one of the input or output boxes

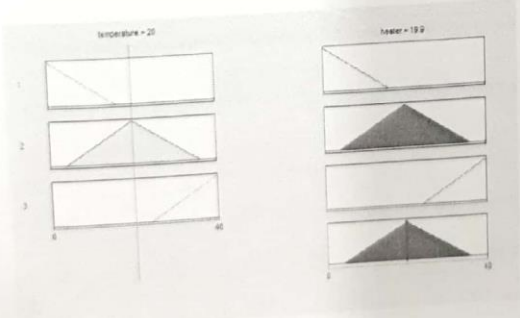
INPUT:

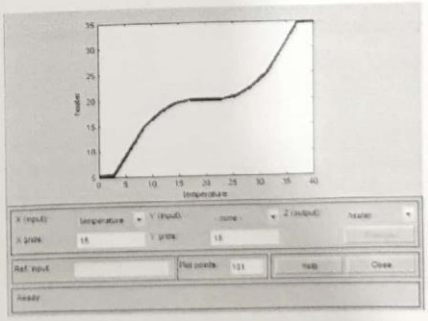


MATLAB CODE:



OUTPUT:





8(14)

Experiment – 4

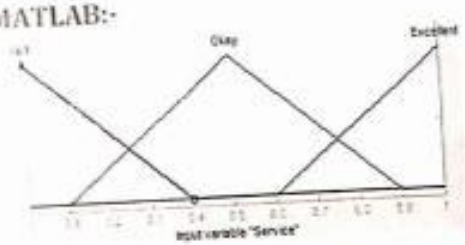
Aim:-

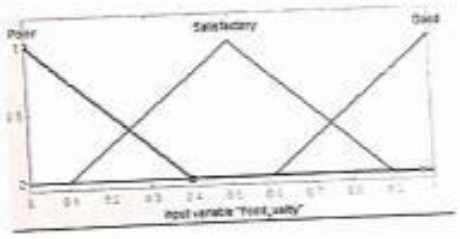
To implement Mamdani type of Fuzzy Controller and to study the control of different input combinations.

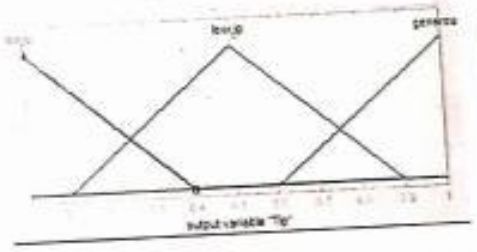
Input:-

Service Quality	Food Quality	Tip
Bad	Poor	NoTip
Bad	Satisfactory	NoTip
Bad	Good	NoTip
Okay	Poor	LowTip
Okay	Satisfactory	LowTip
Okay	Good	LowTip
Excellent	Poor	Generous
Excellent	Satisfactory	Generous
Excellent	Good	Generous

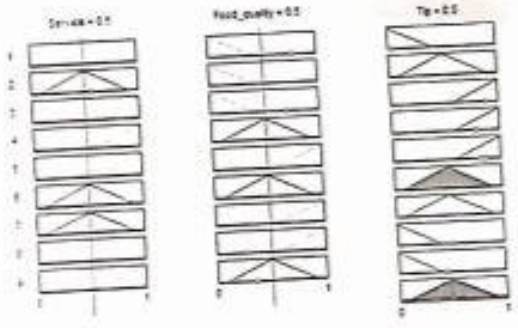
MATLAB:-

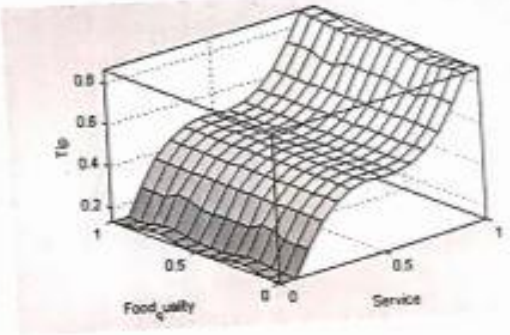






1. If (Service is Bad) and (Food_quality is Poor) then (Tip is No_tip) (1)
2. If (Service is Okay) and (Food_quality is Poor) then (Tip is low_tip) (1)
3. If (Service is Excellent) and (Food_quality is Poor) then (Tip is generous) (1)
4. If (Service is Excellent) and (Food_quality is Satisfactory) then (Tip is generous) (1)
5. If (Service is Excellent) and (Food_quality is Good) then (Tip is generous) (1)
6. If (Service is Okay) and (Food_quality is Satisfactory) then (Tip is low_tip) (1)
7. If (Service is Okay) and (Food_quality is Good) then (Tip is low_tip) (1)
8. If (Service is Bad) and (Food_quality is Good) then (Tip is No_tip) (1)





Experiment – 5

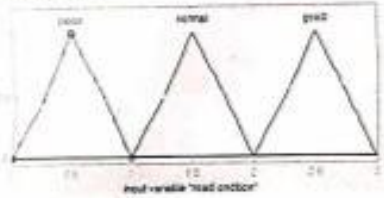
Aim:-

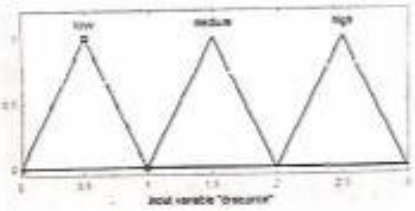
To implement Mamdani type of Fuzzy Controller to implement a vehicular control system.

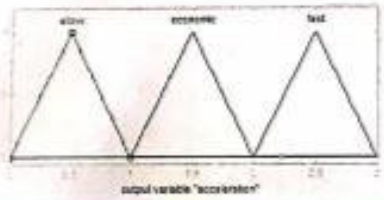
Input:-

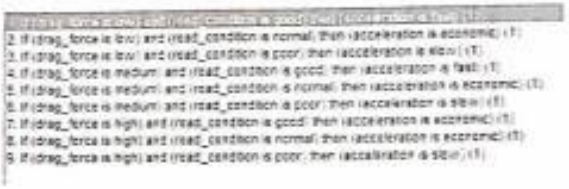
Drag_Force	Road_Condition	Acceleration
Low	Poor	Slow
Low	Normal	Economic
Low	Good	Fast
Medium	Poor	Slow
Medium	Normal	Economic
Medium	Good	Fast
High	Poor	Slow
High	Normal	Economic
High	Good	Economic

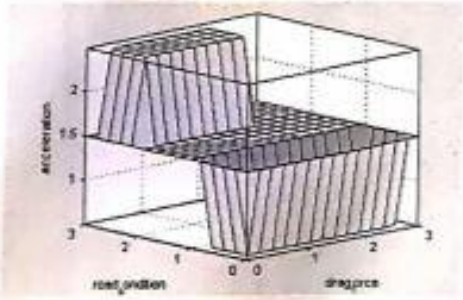
MATLAB:-

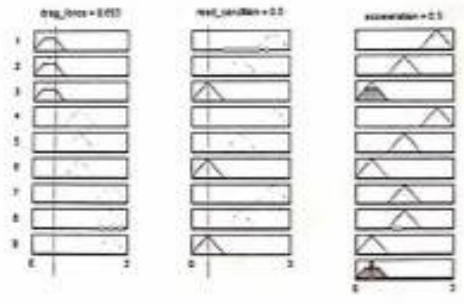












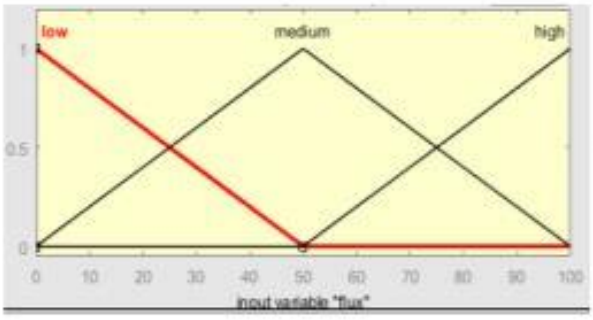
EXPERIMENT - 6

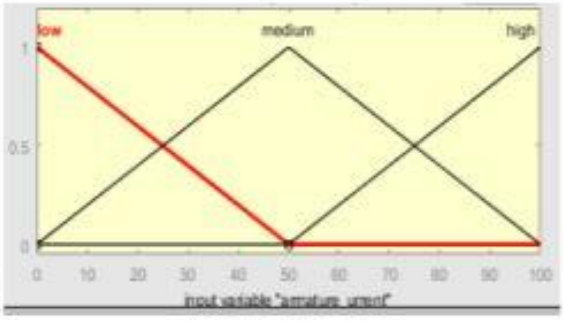
AIM :- To Implement a Sugeno type fuzzy controller for a electrical machine whose output(i.e. speed) can be varied by two inputs viz armature current and flux.

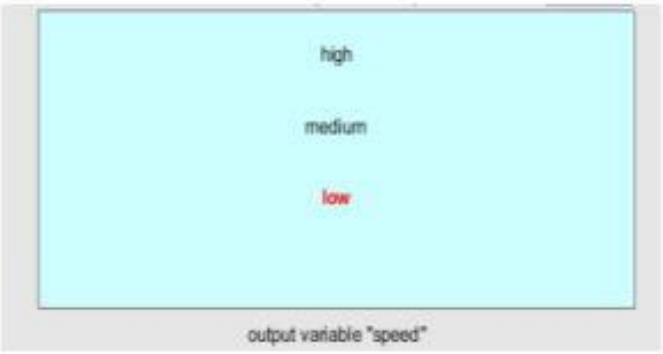
SOFTWARE USED: - MATLAB R2016b

INPUT: -

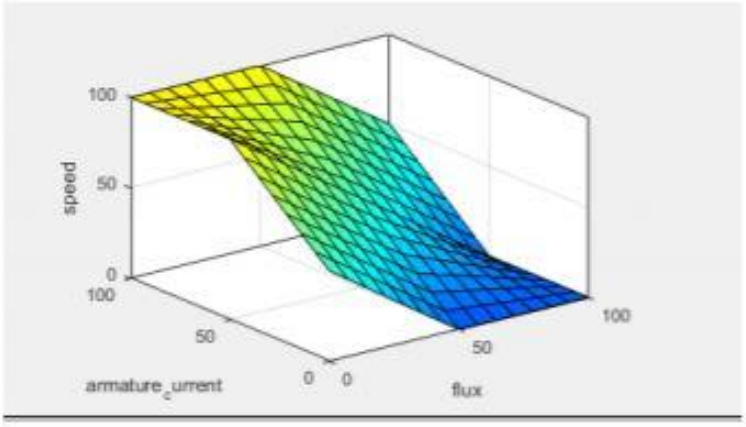
FLUX	ARMATURE CURRENT	SPEED
LOW	LOW	MEDIUM
LOW	MEDIUM	HIGH
LOW	HIGH	HIGH
MEDIUM	LOW	LOW
MEDIUM	MEDIUM	MEDIUM
MEDIUM	HIGH	HIGH
HIGH	LOW	LOW
HIGH	MEDIUM	LOW
HIGH	HIGH	MEDIUM

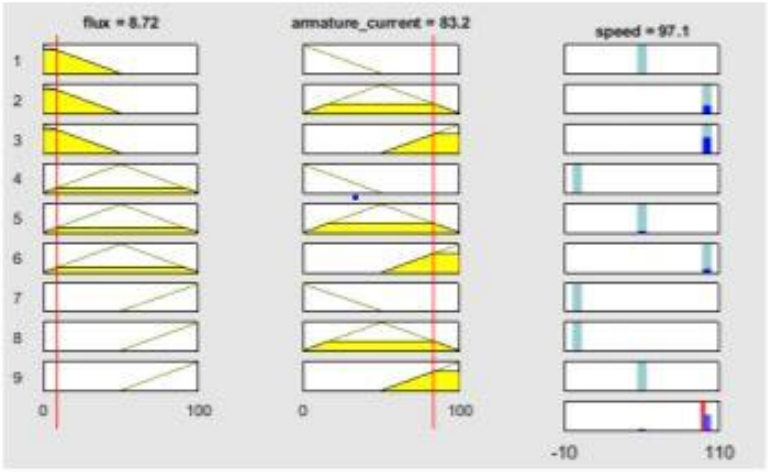






1. If (flux is low) and (armature_current is low) then (speed is medium) (1)
2. If (flux is low) and (armature_current is medium) then (speed is high) (1)
3. If (flux is low) and (armature_current is high) then (speed is high) (1)
4. If (flux is medium) and (armature_current is low) then (speed is low) (1)
5. If (flux is medium) and (armature_current is medium) then (speed is medium) (1)
6. If (flux is medium) and (armature_current is high) then (speed is high) (1)
7. If (flux is high) and (armature_current is low) then (speed is low) (1)
8. If (flux is high) and (armature_current is medium) then (speed is low) (1)
9. If (flux is high) and (armature_current is high) then (speed is medium) (1)



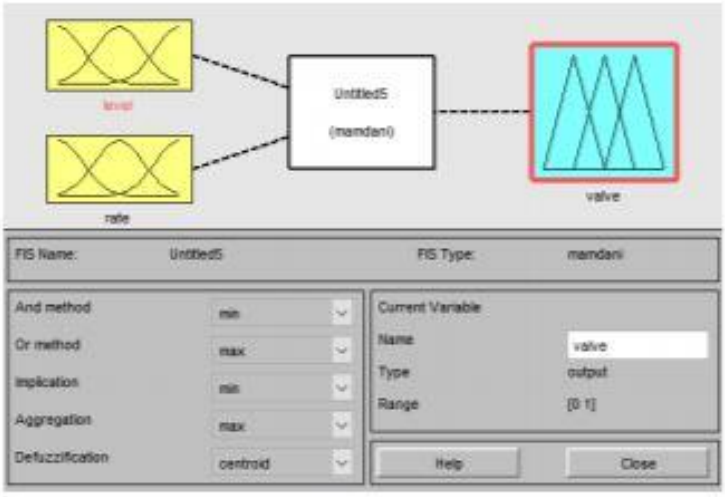


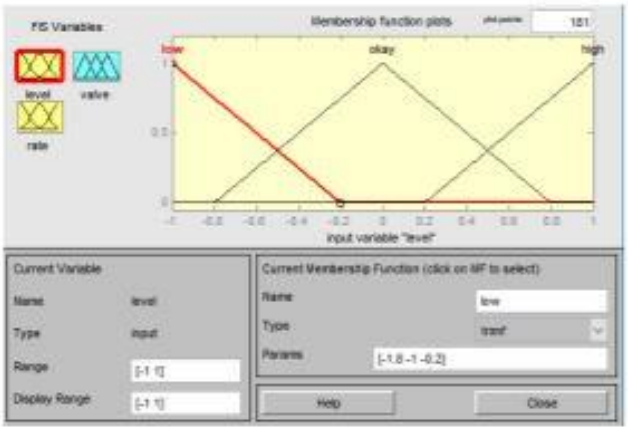
CONCLUSION: - Sugeno type Fuzzy Controller for an electrical machine whose speed was varied by armature current and flux was successfully implemented.

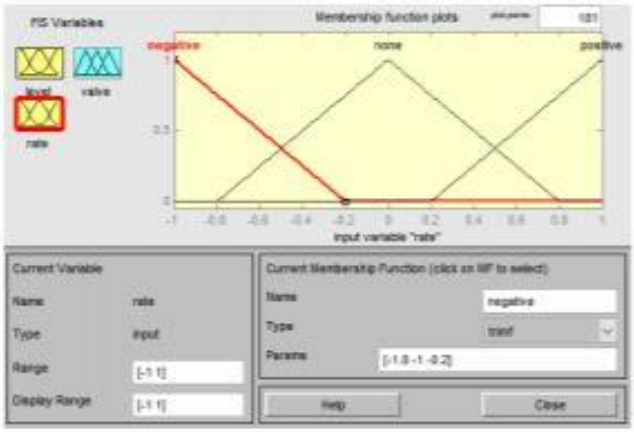
Experiment- 7

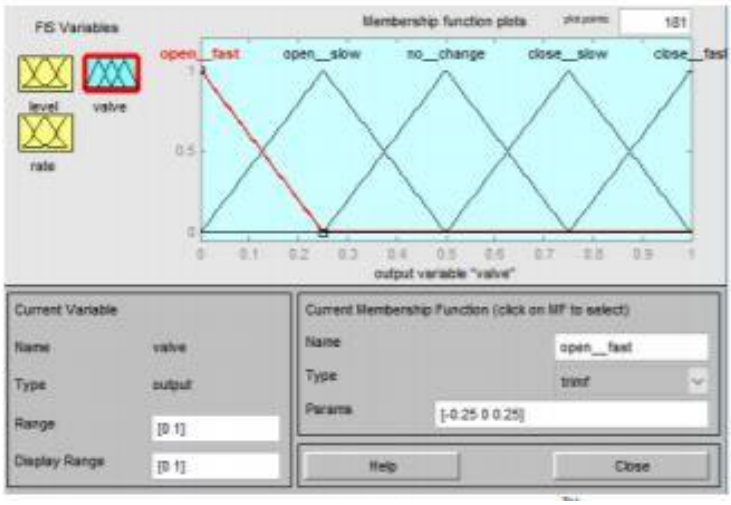
Aim: To implement Fuzzy Controller for 1st order system using MATLAB.

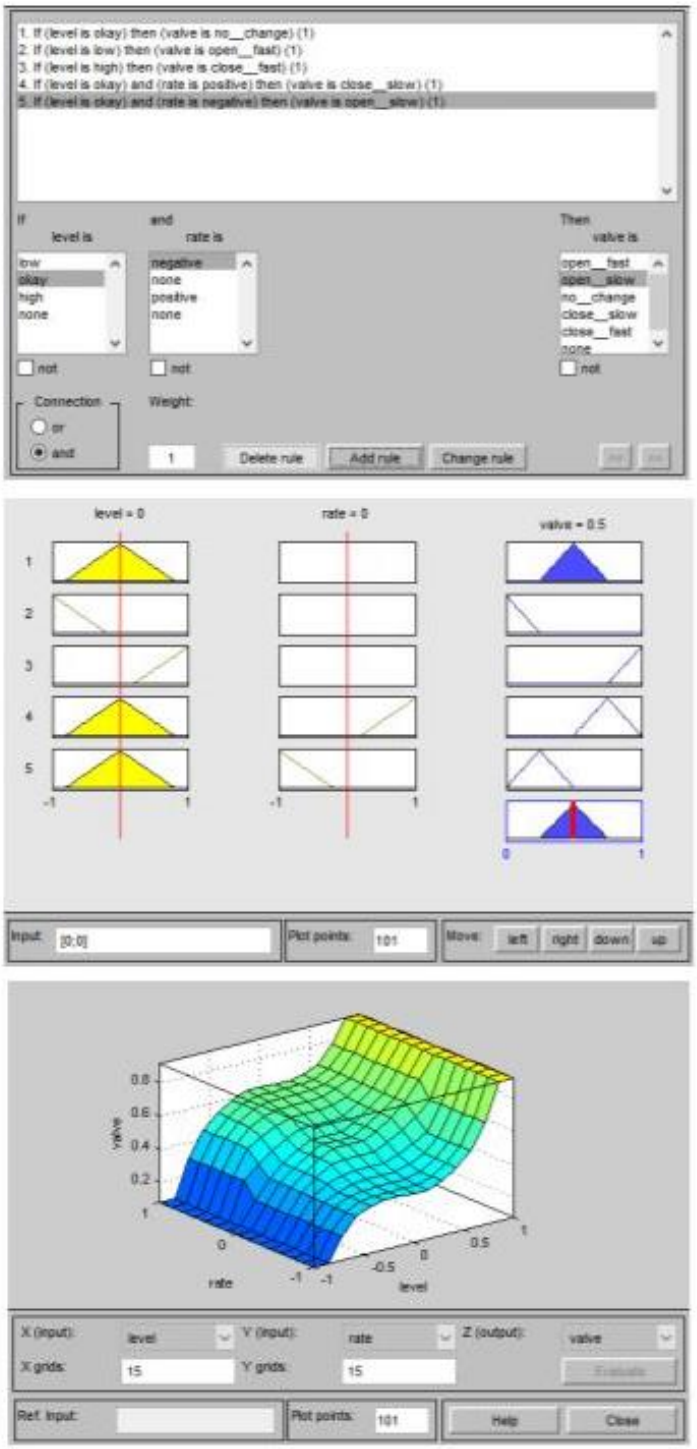
We consider first order water tank system with inputs level of water and rate to control the opening and closing of tank valve.











Conclusion: First order control system was successfully implemented using Fuzzy Logic.

Experiment- 8

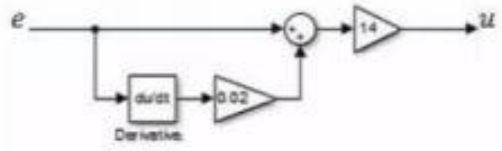
Aim: To implement Fuzzy PID Controller.

Assume the open-loop transfer function of the laboratory servo rig is:

$$G(s) = \frac{1.85}{s(1+0.1s)}$$

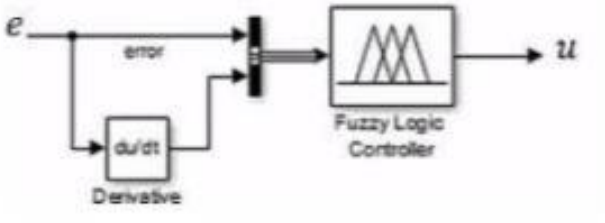
1. Design a conventional PD controller for the servo to give a closed-loop settling time to a step input of approximately 0.5 seconds. The closed-loop phase margin should be around 50°.

For the given specifications, from the root locus the Gain Kc = 14



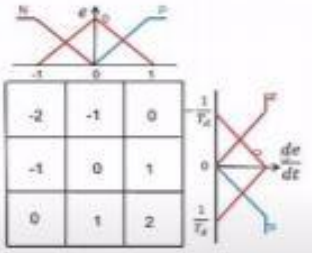
$$C = 14(1 + 0.02s) = K_c(1 + T_d s)$$

Simukink Model



Fuzzy Model

The error path has unit gain and the derivative is modified by T_d. Thus, for the fuzzy PD we have:

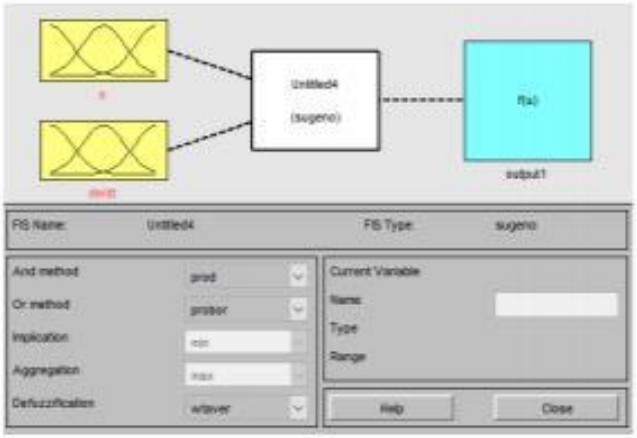


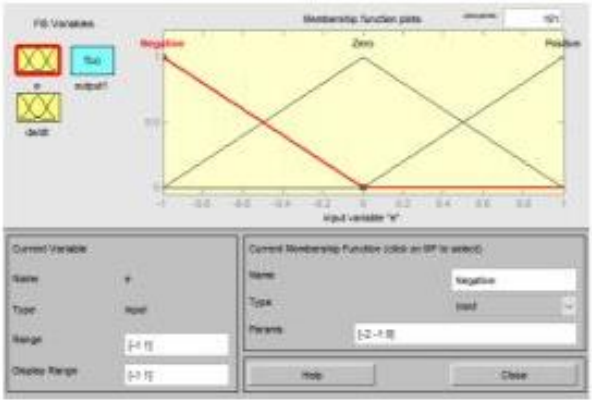
-2	-1	0
-1	0	1
0	1	2

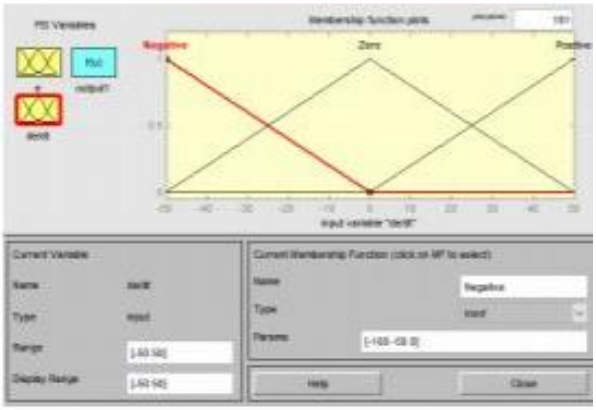
1/Td = 1/0.02 = 50 Kc = 14

At the output, the signal magnitude is adjusted by the gain K_c

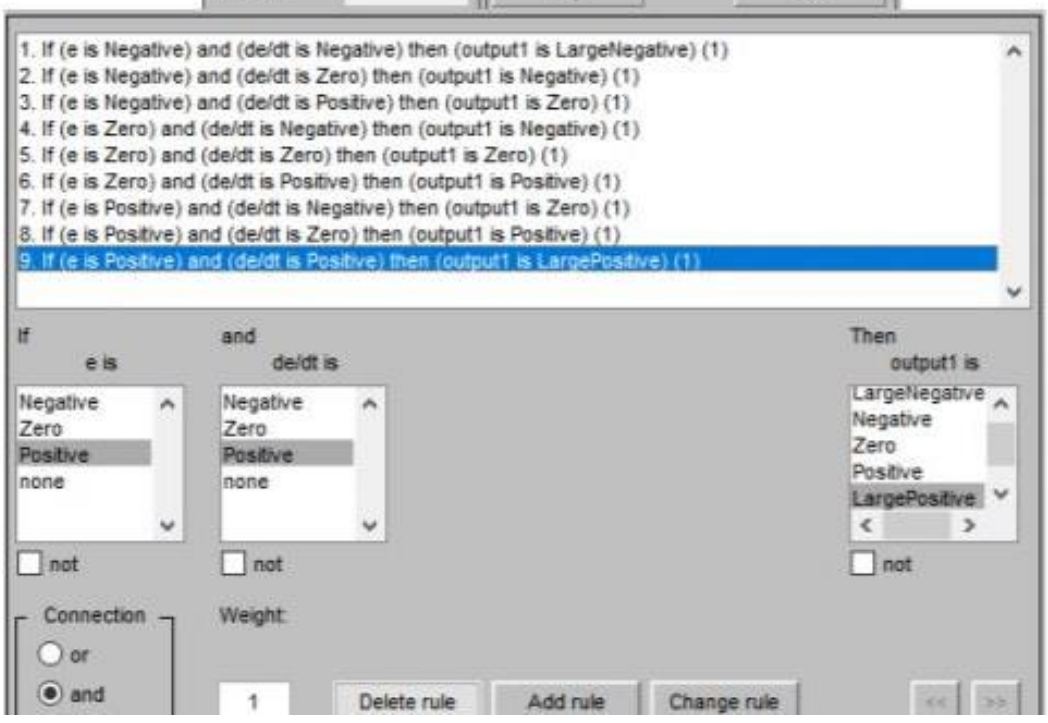
$-2K_c$	$-K_c$	0
$-K_c$	0	K_c
0	K_c	$2K_c$

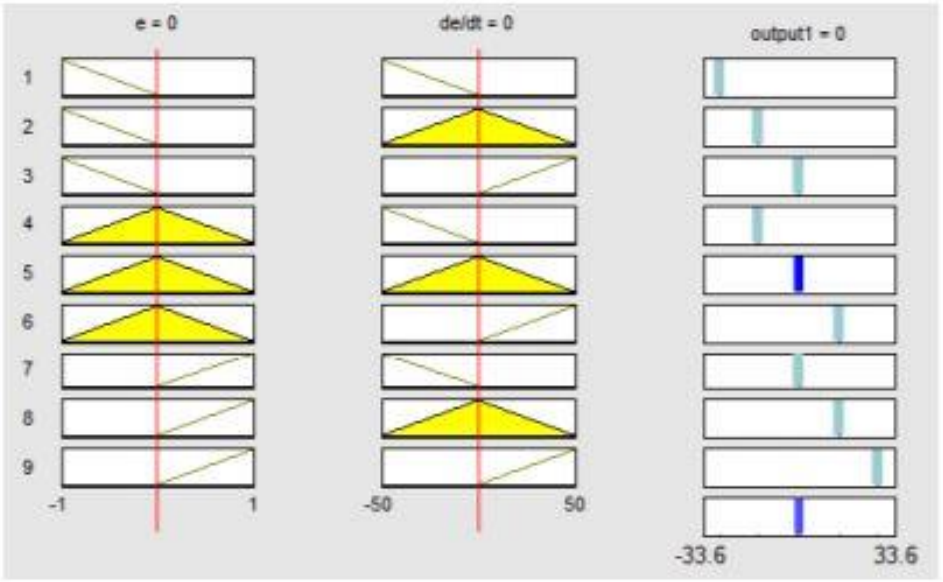


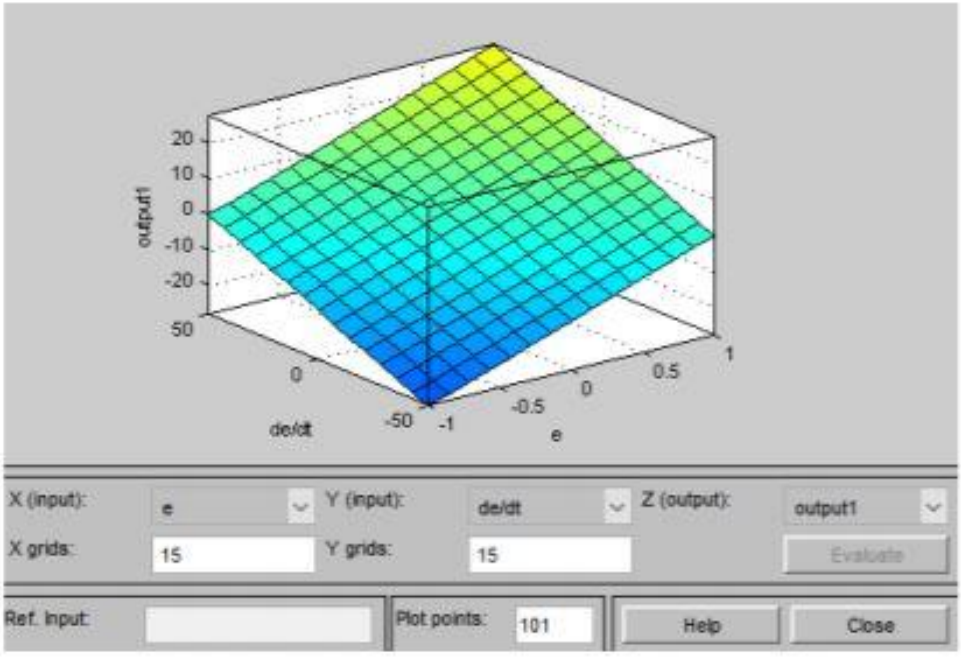

FIS Variables
Membership function plots
Negative
Zero
Positive
e
de/dt
output1
Current Variable
Current Membership Function (click on MF to select)
Name
Type
Range
Display Range
Help
Close


FIS Variables
Membership function plots
Negative
Zero
Positive
e
de/dt
output1
Current Variable
Current Membership Function (click on MF to select)
Name
Type
Range
Display Range
Help
Close


FIS Variables
Membership function plots
LargePositive
Positive
Zero
Negative
LargeNegative
output variable "output1"
e
de/dt
output1
Current Variable
Name
output1
Type
output
Range
[0 1]
Display Range
Current Membership Function (click on MF to select)
Name
LargeNegative
Type
constant
Params
-20
Help
Close


1. If (e is Negative) and (de/dt is Negative) then (output1 is LargeNegative) (1)
2. If (e is Negative) and (de/dt is Zero) then (output1 is Negative) (1)
3. If (e is Negative) and (de/dt is Positive) then (output1 is Zero) (1)
4. If (e is Zero) and (de/dt is Negative) then (output1 is Negative) (1)
5. If (e is Zero) and (de/dt is Zero) then (output1 is Zero) (1)
6. If (e is Zero) and (de/dt is Positive) then (output1 is Positive) (1)
7. If (e is Positive) and (de/dt is Negative) then (output1 is Zero) (1)
8. If (e is Positive) and (de/dt is Zero) then (output1 is Positive) (1)
9. If (e is Positive) and (de/dt is Positive) then (output1 is LargePositive) (1)
If
e is
Negative
Zero
Positive
none
not
and
de/dt is
Negative
Zero
Positive
none
not
Then
output1 is
LargeNegative
Negative
Zero
Positive
LargePositive
not
Connection
or
and
Weight:
1
Delete rule
Add rule
Change rule
<<
>>





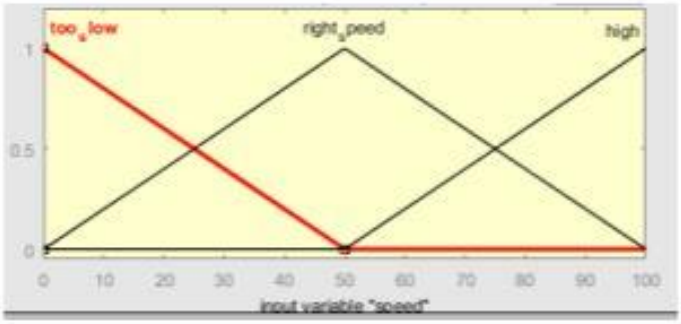
Conclusion: PD Controller was successfully implemented using Fuzzy Logic.

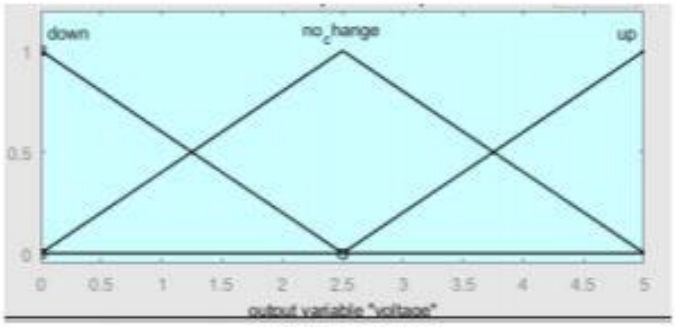
EXPERIMENT – 9

AIM:- Simulaate fuzzy motor speed controller in simulink.

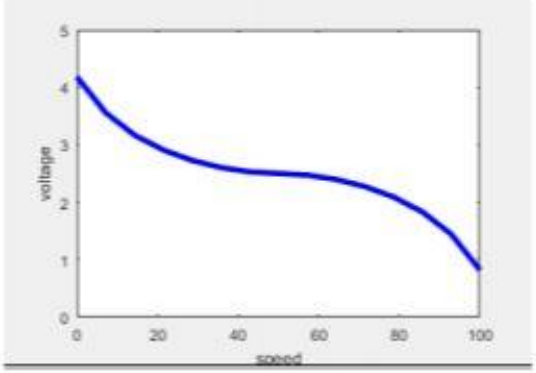
SOFWARE USED:- MATLAB R2016b

INPUT:-

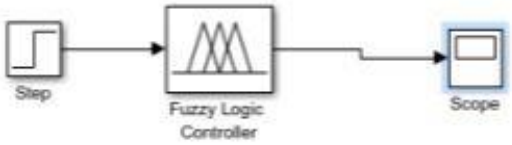




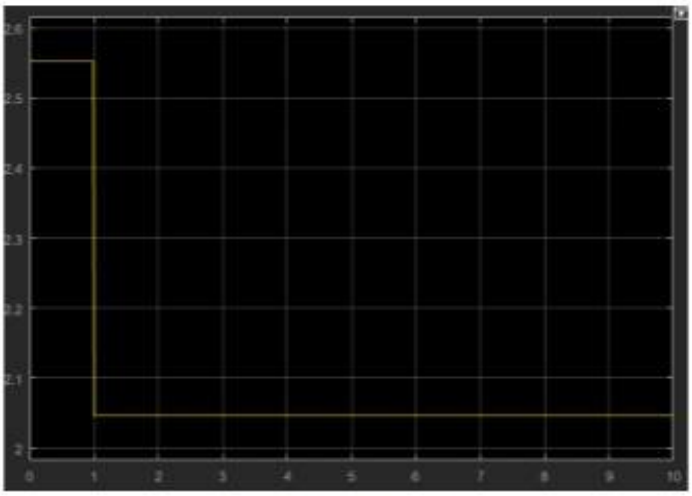
1. If (speed is too_slow) then (voltage is up) (1)
2. If (speed is right_speed) then (voltage is no_change) (1)
3. If (speed is high) then (voltage is down) (1)



Fuzzy controller in Simulink with Step input of initial value= 40 and final value= 80:



BLOCK DIAGRAM



OUTPUT OF SCOPE

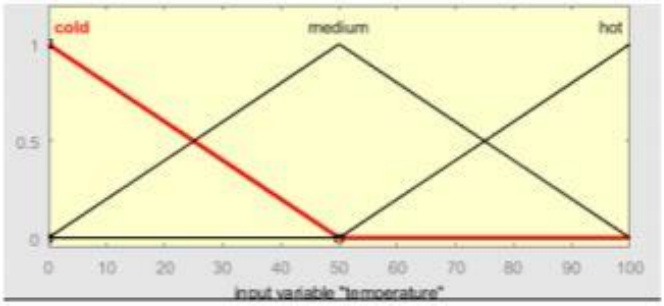
Conclusion: Simulink of fuzzy motor speed controller was successfully performed using Simulink and fuzzy logic designer

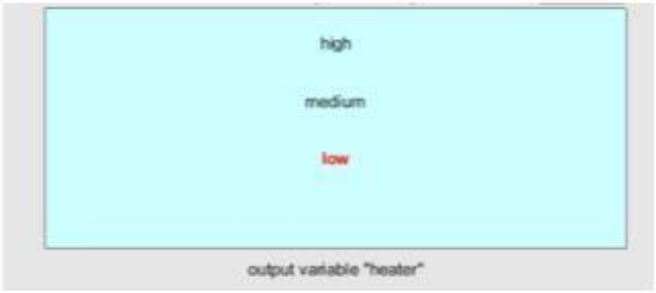
EXPERIMENT – 10

AIM:- To implement Sugeno type Fuzzy Controller for temperature control system using MATLAB.

SOFWARE USED:- MATLAB R2016b

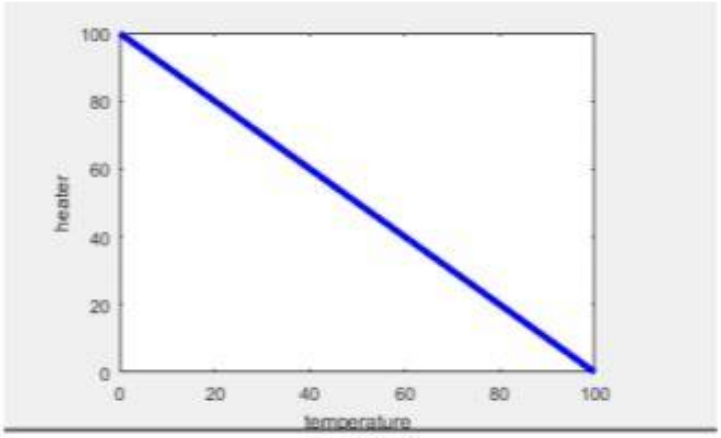
INPUT:-

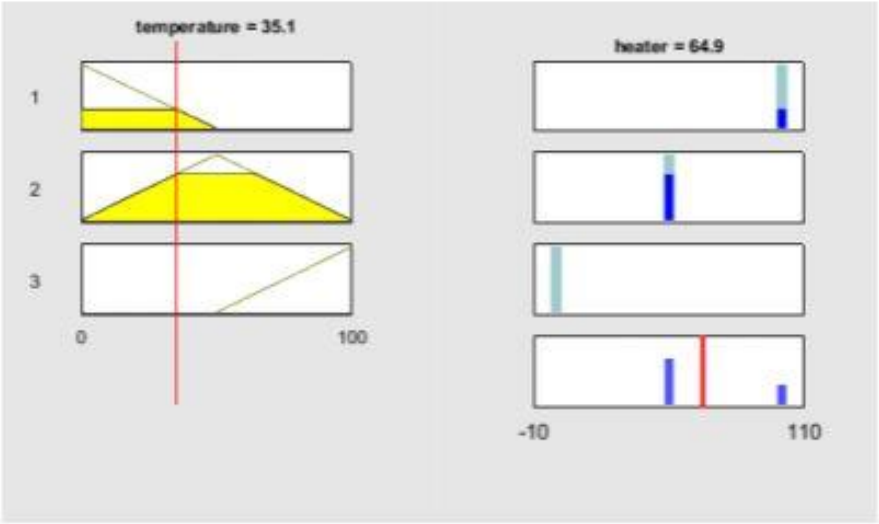




1. If (temperature is cold) then (heater is high) (1)
2. If (temperature is medium) then (heater is medium) (1)
3. If (temperature is hot) then (heater is low) (1)

OUTPUT:-





CONCLUSION: - Surgeno type fuzzy controller for temperature control system was implemented using MATLAB.