به نام خدا

تمرین شماره ۳

درس هوش مصنوعی و سیستم های خبره

تهیه کننده: علیرضا امیری

شماره دانشجویی: ۴۰۲۰۲۴۱۴

استاد درس: دکتر نجفی

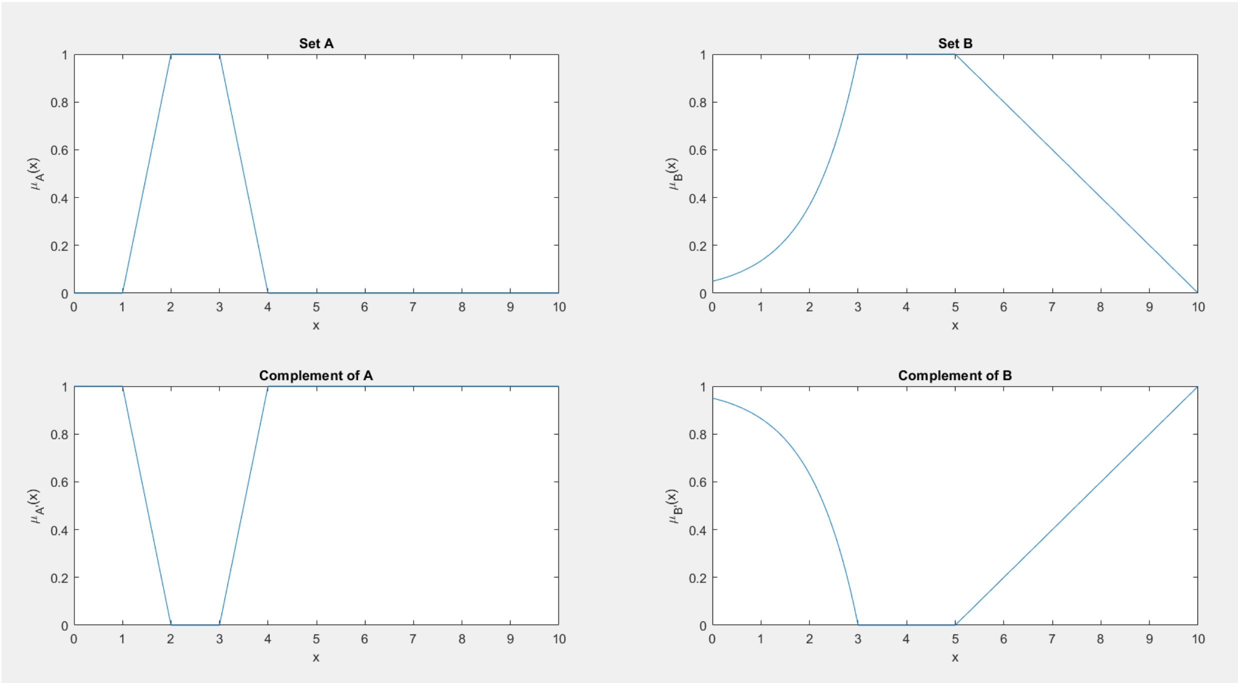
بهار ۱۴۰۲

Answer of Problem 1:

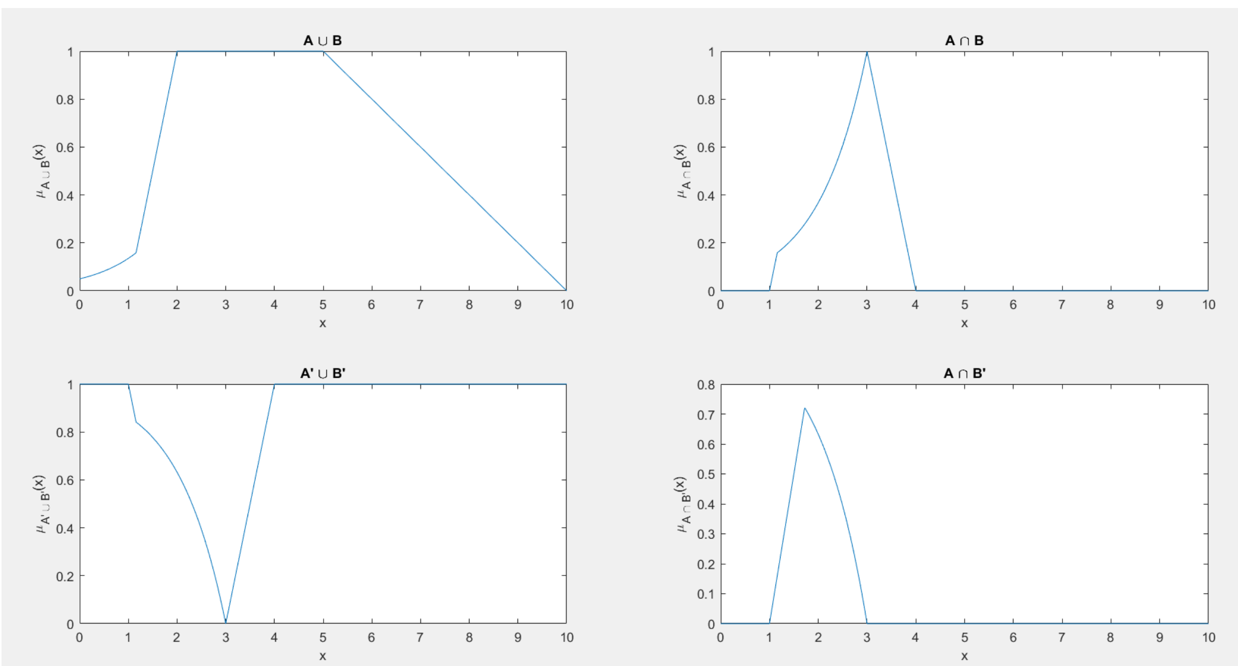
First we need to define Sets of A and B. this is done using the code below.

```
% Define the range of x
x_values = 0:0.01:10;

% Define the membership functions for Set A
set_A = zeros(size(x_values));
set_A(x_values >= 1 & x_values < 2) = x_values(x_values >= 1 & x_values < 2) - 1;
set_A(x_values >= 2 & x_values < 3) = 1;
set_A(x_values >= 3 & x_values < 4) = 4 - x_values(x_values >= 3 & x_values < 4);

% Define the membership functions for Set B
set_B = zeros(size(x_values));
set_B(x_values < 3) = exp(x_values(x_values < 3) - 3);
set_B(x_values >= 3 & x_values < 5) = 1;
set_B(x_values >= 5 & x_values <= 10) = 1 - (x_values(x_values >= 5 & x_values <= 10) - 5) /
5;
% Define the complement of sets A and B
set_A_complement = 1 - set_A;
set_B_complement = 1 - set_B;

% Plot the sets A and B
figure;
subplot(2, 2, 1);
plot(x_values, set_A);
title('Set A');
xlabel('x');
ylabel('\mu_A(x)');

subplot(2, 2, 2);
plot(x_values, set_B);
title('Set B');
xlabel('x');
ylabel('\mu_B(x)');

% Plot the complements of sets A and B
subplot(2, 2, 3);
plot(x_values, set_A_complement);
title('Complement of A');
xlabel('x');
ylabel('\mu_{A''}(x)');

subplot(2, 2, 4);
plot(x_values, set_B_complement);
title('Complement of B');
xlabel('x');
ylabel('\mu_{B''}(x)');
```

The diagram of these sets will be as shown below.



Next we plot the required sets using the code below.



To Define if these sets are convex or not, the code below is used. The result is then shown.

```
% Check if sets A and B are convex
is_A_convex = all(diff(set_A) >= 0) || all(diff(set_A) <= 0);
is_B_convex = all(diff(set_B) >= 0) || all(diff(set_B) <= 0);

% Display the results
fprintf('Is Set A a convex set? %d\n', is_A_convex);
```

```
fprintf('Is Set B a convex set? %d\n', is_B_convex);
```

```
>> t12
Is Set A a convex set? 0
Is Set B a convex set? 0
```

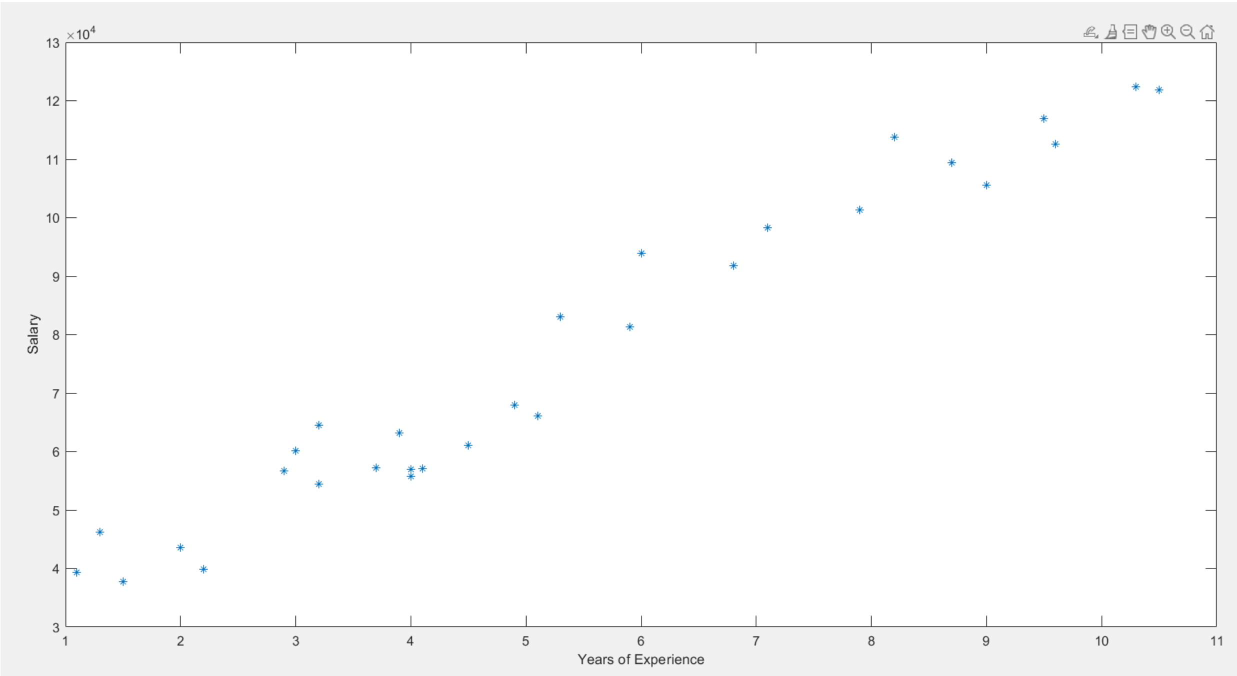
Problem 2: Imagine that you work in a company and you are asked to design a system which can predict the probable salary of an employee based on their years of experience. For more information, you are given a dataset which contains information about years of experience and salary of 30 employees. Without using the Simulink package for fuzzy logic systems and by using pure MATLAB script which satisfies the following conditions:

- The function takes the years of experience as the input and outputs the estimated salary.
- Has 5 membership functions for the input.
- Has 5 membership functions for the output
- Rules are written inside the function.

Answer of Problem 2:

In order to define membership functions of input and outputs and set the range of their values, we need to analyze the given dataset.

By plotting Salary vs Years of Experience, we get an intuition of the range of values as shown in figure below.



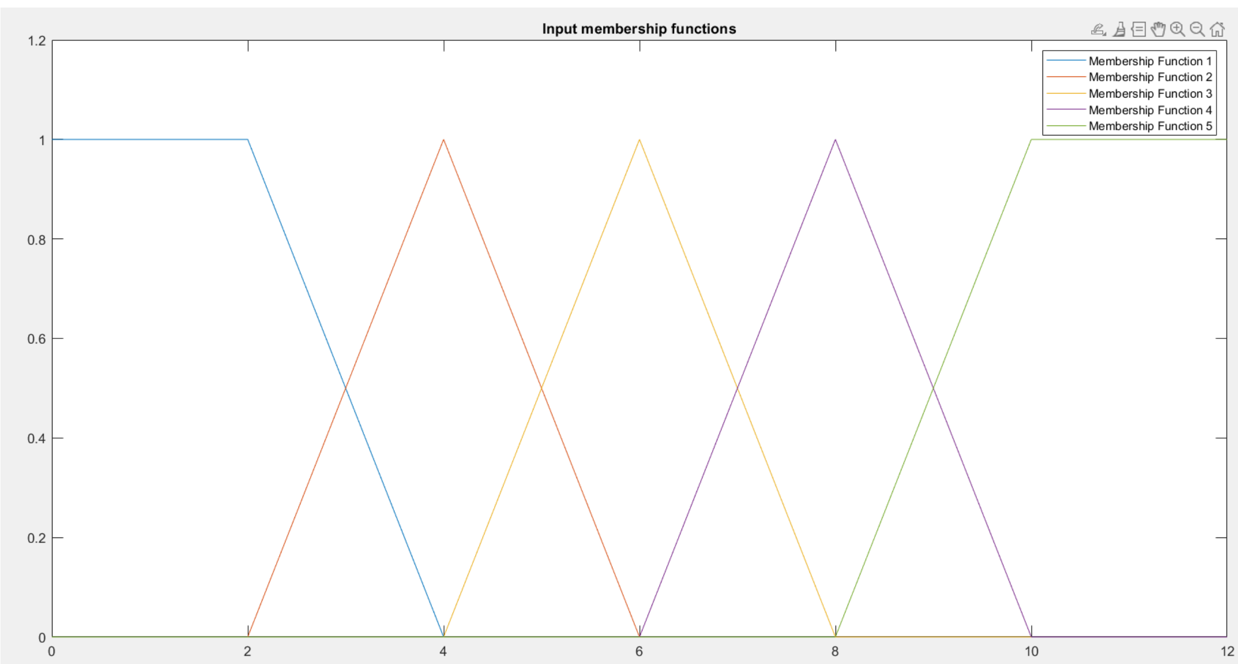
Then, We conclude that the input values for years of experience ranges from 1 to 11 and the values of the salaries ranges from 3000 to 13000. Given this information, we can divide each of them to 5 equal sections and design our membership functions.

Since our ultimate goal is to develop a function that inputs a single number as year of experience and outputs the estimated salary, we begin by creating a function.

```
function [Estimated_Sallary] = Salary_Estimator(INPUT)
%% Define input membership functions: Years of experience
xp = 0:0.01:12;


mf1 = trapmf(xp, [0,0,2,4]);
mf2 = trimf(xp, [2,4,6]);
mf3 = trimf(xp, [4,6,8]);
mf4 = trimf(xp, [6,8,10]);
mf5 = trapmf(xp, [8,10,12,12]);


figure(1)
plot(xp, mf1);
hold on
plot(xp, mf2);
hold on
plot(xp, mf3);
hold on
```

```
plot(xp, mf4);
hold on
plot(xp, mf5);
ylim([0, 1.2]);
```

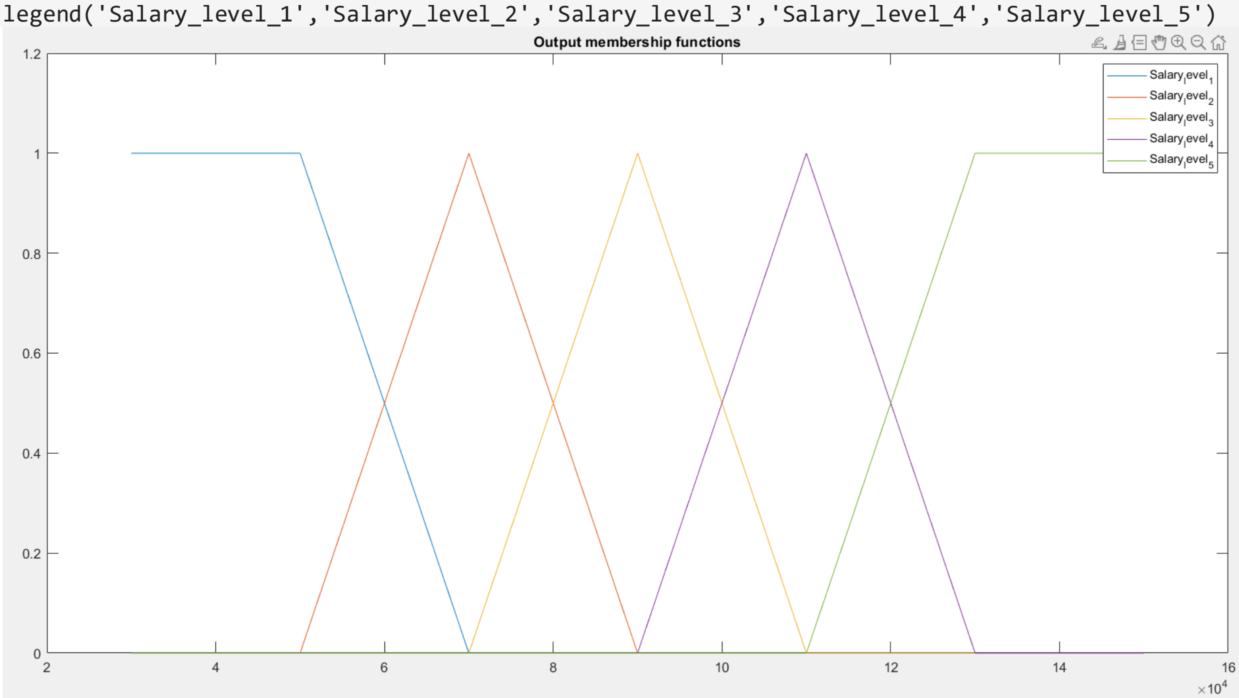
In this code, 5 membership was defined for input. The resulting functions are shown in the figure below.



Similarly, membership functions of the output are designed and shown in the below images.

```
%% Define output membership functions: Salary

Salary = 30000:150000;
```

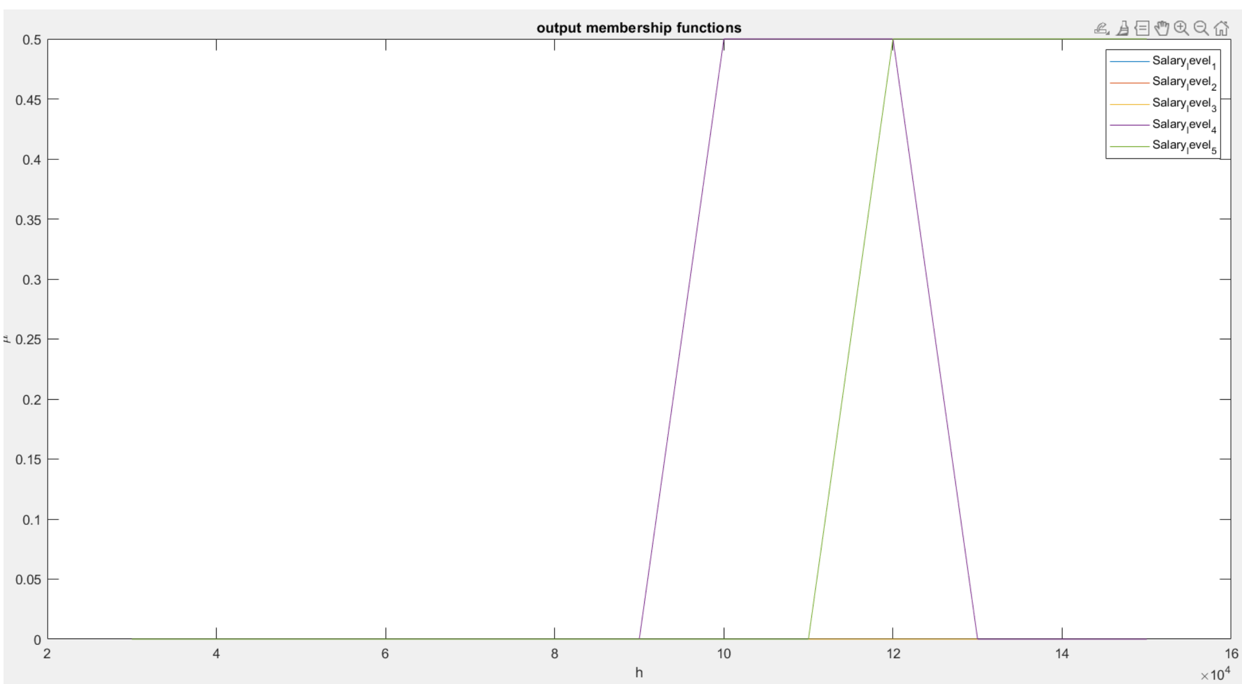
```
Salary_level_1 = trapmf(Salary , [30000,30000,50000,70000]);
Salary_level_2 = trimf(Salary,[50000,70000,90000]);
Salary_level_3 = trimf(Salary,[70000,90000,110000]);
Salary_level_4 = trimf(Salary,[90000,110000,130000]);
Salary_level_5 = trapmf(Salary,[110000,130000,150000,150000]);

figure(2)
plot(Salary, Salary_level_1);
hold on
plot(Salary, Salary_level_2);
hold on
plot(Salary, Salary_level_3);
hold on
plot(Salary, Salary_level_4);
hold on
plot(Salary, Salary_level_5);
ylim([0, 1.2]);
title('Output membership functions')
legend('Salary_level_1','Salary_level_2','Salary_level_3','Salary_level_4','Salary_level_5')
```



Next, The rules of the fuzzy system need to be defined. Since we have 5 input MFs and 5 MFs and as we have seen that there is a linear relation between years of experience and salary, we assign MFs respectively.

```
%% Define Rules
```

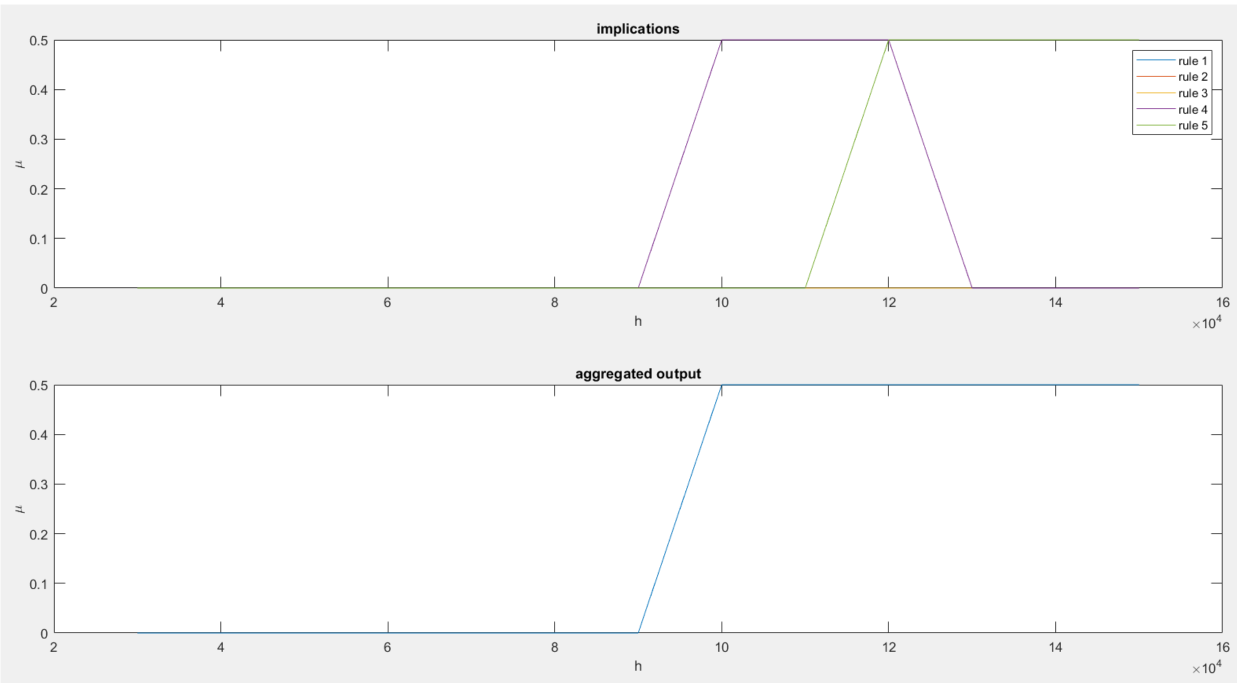
```
%Fuzzify the INPUT
inp1 = trapmf(INPUT, [0,0,2,4]);
inp2 = trimf(INPUT, [2,4,6]);
inp3 = trimf(INPUT, [4,6,8]);
inp4 = trimf(INPUT, [6,8,10]);
inp5 = trapmf(INPUT, [8,10,12,12]);

imp1 = min(Salary_level_1 ,inp1);
imp2 = min(Salary_level_2 ,inp2 );
imp3 = min(Salary_level_3 ,inp3 );
imp4 = min(Salary_level_4 ,inp4 );
imp5 = min(Salary_level_5 ,inp5 );

figure(3)
plot(Salary, imp1, Salary, imp2,Salary, imp3,Salary, imp4,Salary, imp5)

title('output membership functions')
ylabel('\mu')
xlabel('h')
legend(["Salary_level_1" "Salary_level_2" "Salary_level_3" "Salary_level_4" "Salary_level_5"])
```



The next step is aggregation of contacting MFs. We calculate the union of these MFs using Max function.

```
%% Aggregation
aggregated = max(imp1,max(imp2,max(imp3 ,max(imp4 ,imp5))));
```

```
figure(4)
subplot(2,1,1)
plot(Salary, imp1, Salary, imp2,Salary, imp3,Salary, imp4,Salary, imp5)

title('implications')
ylabel('\mu')
xlabel('h')
legend(["rule 1" "rule 2" "rule 3" "rule 4" "rule 5"])

subplot(2,1,2)
plot(Salary, aggregated)

title('aggregated output')
ylabel('\mu')
xlabel('h')
```



As the final step to calculate the estimated salary, we calculate COG(center of gravity) for the aggregated shape.

```
%% Defuzzification

% Calculate COG using equation
defuzzified = sum(aggregated.*Salary)/sum(aggregated);
Estimated_Sallary = defuzzified
```

As an example input, we give the function number 9 as 9 years of experience and the output will be as below

```
>> Salary_Estimator(9)

Estimated_Sallary =

   1.2242e+05
```

Which in comparison to the given dataset, it is 25% higher.

So, to achieve a measure to analyze this model, mean error value of the predicted and true salaries are calculated using the code below.

```
errors = zeros(1,30);
for i=1:30

    input = SalaryData.YearsExperience(i);
    predicted = Salary_Estimator(input);
    errors(i) = SalaryData.Salary(i) - predicted;
end
sum(errors)/30
```

The resulting mean of errors is: -7171

Problem 3: Imagine there is a house which only has two rooms and, in each room, there is an air conditioner. A fuzzy controller should be designed to control temperature in these rooms. This controller takes the difference between desired and measured temperatures in both rooms and outputs the required heat of each air conditioner. The governing equations are:

dT1 dt = −0.05 T1 + 0.01T2 + 0.008 Tout + uac1

dT2 dt = −0.039 T2 + 0.01T1 + 0.0075 Tout + uac2

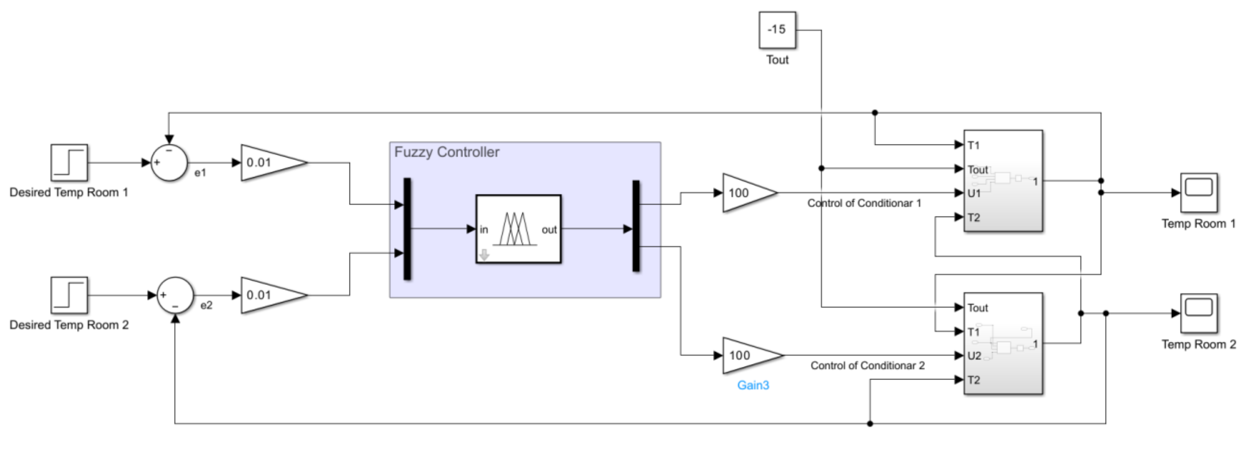
where T1 is the temperature in room 1, T2 the temperature in room 2, Tout the outside temperature, uac1, the required heat for the air conditioner in room 1 and uac2, the required heat for the air conditioner in room 2. (The air conditioner can produce both warm and cold air) Design a fuzzy controller and simulate the result for Tout = −15◦C with various initial conditions.

Answer of Problem 3:

In order to design a controller for this system, first it is needed to get a better understanding of the system inputs and outputs.

The system is given 2 inputs, which are Desired Temperature of both of the rooms. There is a single Fuzzy controller which has to input the errors of both rooms and also outputs 2 control commands to air conditioners of each room. Thus, we have 2 errors, a fuzzy system with 2 inputs and 2 outputs and also we have two air conditioners as our actuators.

The resulting diagram for this system is shown below. The inside parts of each block will be explained later on this report.

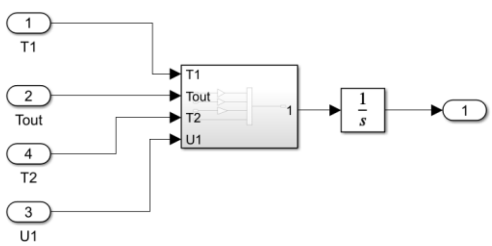


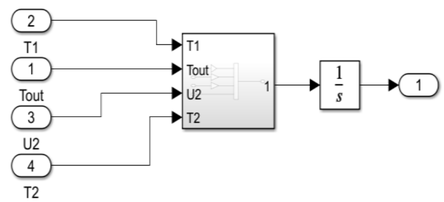
As we can see, there are two control loops with the fuzzy controller acting as the controller for both, and it is worthy of attention that the inputs and outputs of the controller is scaled to a normalized range using two gains.

Model of air conditioners:

Given the equations below for each air conditioner, we attempt to model them in Simulink.

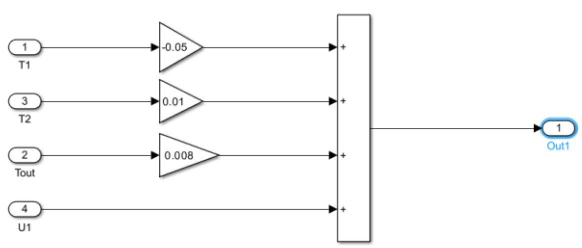
$$\frac{dT_1}{dt} = -0.05\ T_1 + 0.01T_2 + 0.008\ T_{out} + u_{ac1}$$
$$\frac{dT_2}{dt} = -0.039\ T_2 + 0.01T_1 + 0.0075\ T_{out} + u_{ac2}$$

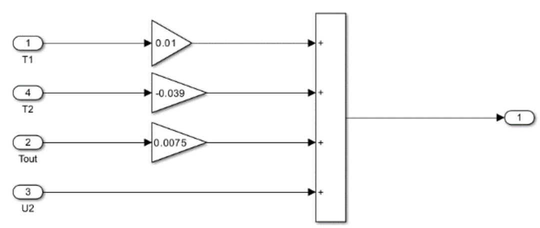




Each model has 4 inputs, which are the temperatures and control command. At the end, since these are the differential equations, we need an integrator to calculate the temperature.

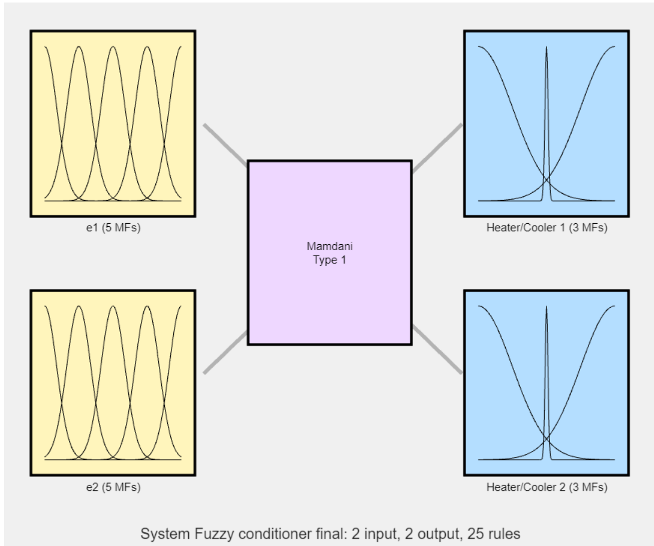
Inside each of these blocks, the gains of each input is considered and calculated.



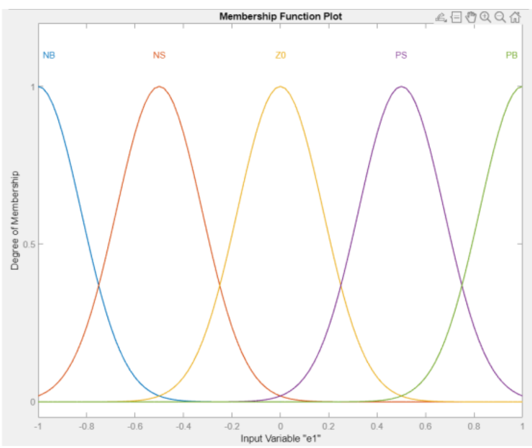


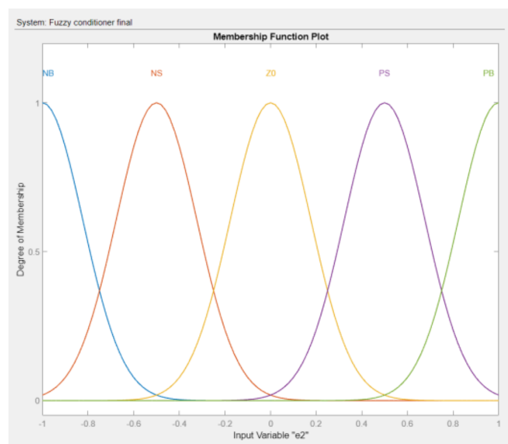
Designing Fuzzy membership functions:

As explained before, we have 2 inputs and 2 outputs for the fuzzy system. We considered to have 5 membership functions for each input error as {Negative Big, Negative Small, Zero, Positive small, Positive Big} and also 3 output membership function for each as {Cooler, Off, Heater}. The resulting controller looks like this:

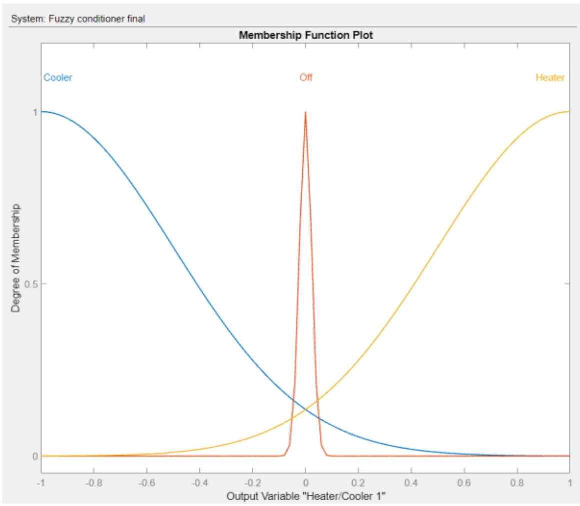


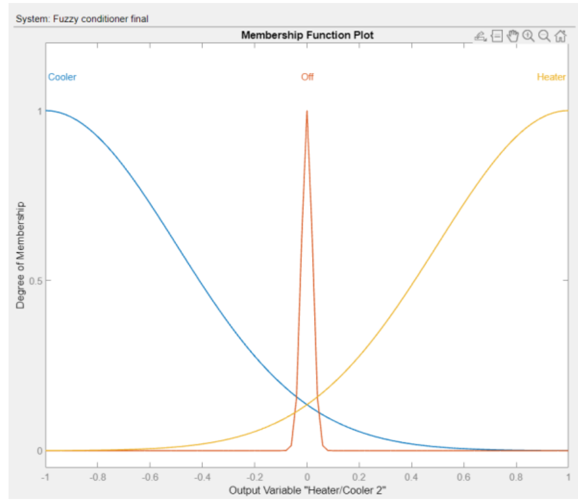
The design of membership functions for inputs consists of 5 gaussian MF in range of -1 and 1. As shown below.





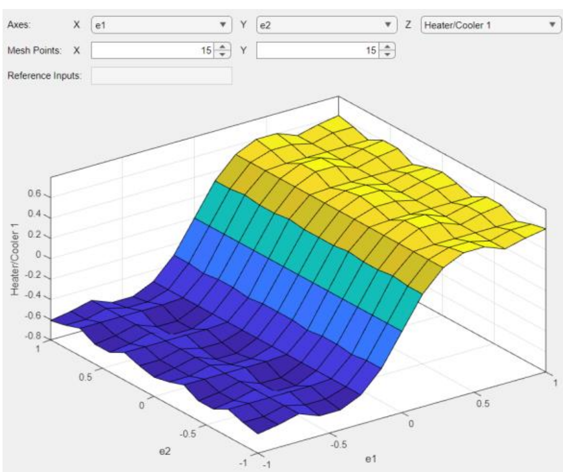
The design of output membership functions required more attention, since with an evenly distributed MFs, the controller would be off around the setpoint and this leads to poor control. Thus, the MF of ZERO is assumed to be much narrower than the others as shown below.

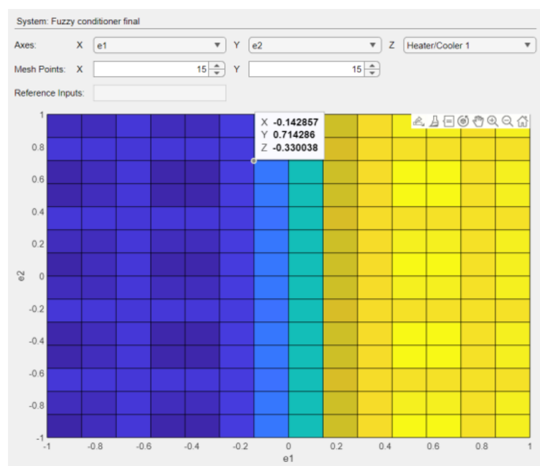




Design Rules:

Having 2 sets of inputs with 5 state for each, results in 25 rules, as shown in the below image.

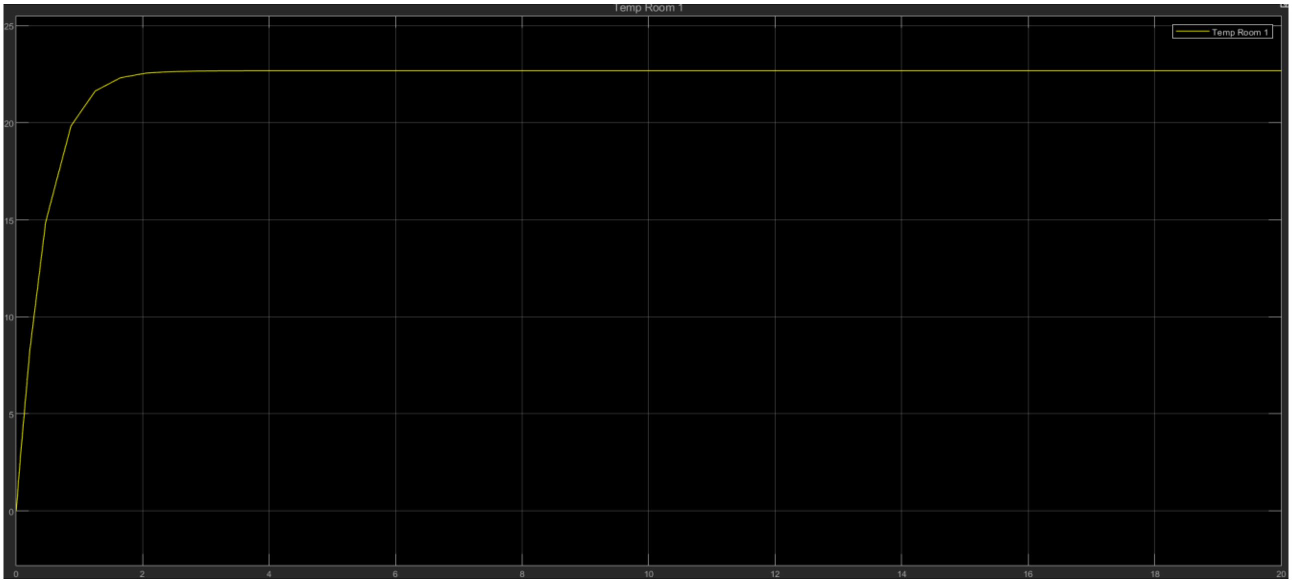


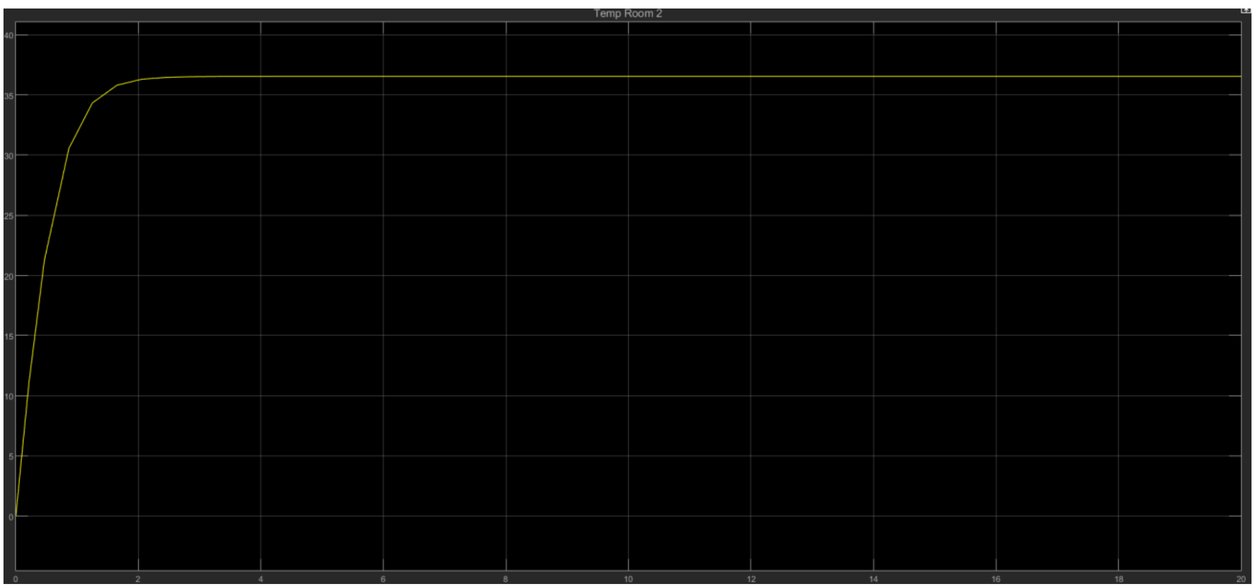


Result:

Having the fuzzy controller designed, we can set initial condition for temperatures of the rooms and assess the step responses.

In test 1, we set Desired temperature of room 1 = 23 and room 2 = 37.


Temp Room 1
Temp Room 1
25
20
15
10
5
0
0
2
4
6
8
10
12
14
16
18
20


Temp Room 2
40
35
30
25
20
15
10
5
0
0
2
4
6
8
10
12
14
16
18
20

Thank you