Question

If the solar radiation level is 75 mW/cm^2 and the temperature is 25°C, find the power output of this fuzzy rule-based system in watts (W).
a) Explain each step in detail.
b) Compare the result you obtained in part a with the result obtained using MATLAB. Use the center of gravity (centroid) method for defuzzification in Mamdani Fuzzy Modeling.

Answer

To answer this question, we first need to outline the steps. Mamdani Fuzzy Modeling consists of the following stages:

- Identify the inputs and outputs of the problem.
- Determine the membership functions for the inputs and outputs.
- Create the rule base.

When an input is provided to the system, the following occurs:

- Determine the active rules based on the input using the selected rules.
- The input undergoes fuzzification.
- Process the rules.
- Perform fuzzy inference.
- To convert the result into a format that the actual system can understand, the defuzzification step is applied. As mentioned in the question, the center of gravity (centroid) method will be used for defuzzification.

Solution for Option A

Let's first manually go through the steps mentioned above.

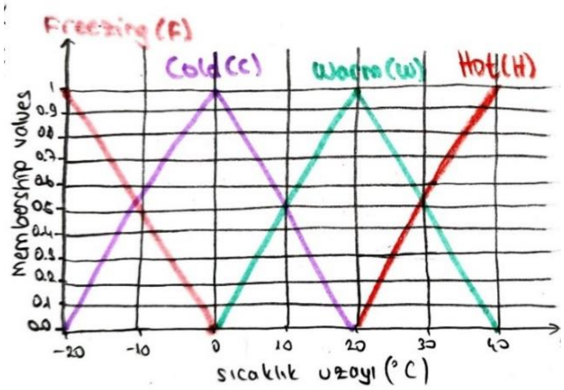
For this part, let's use previous examples. In a previous example related to this question, the following was explained:

The power generated by photovoltaic (PV) panels varies as a function of solar radiation and operating temperature. Thus, the power generated increases as the solar radiation level increases, but decreases as the operating temperature rises. Therefore, meteorological data such as solar radiation, temperature, wind, and waves are measured daily, monthly, and annually for use in renewable energy projects. These measurements show that 50 mW/cm^2 is considered low, while 200 mW/cm^2 is considered well above average and assumed to be good. Similarly, the ambient temperature ranges from –20°C to 40°C. Based on these observations, a precise solar radiation universe (S) can be defined within the range of {50,200} mW/cm^2, and a precise temperature universe (T) can be defined within the range of {–20,40}°C.

(a) Divide the S universe into the fuzzy subsets of low (L), sufficient (Su), good (G), and high (Hi), represented by equally distributed triangular membership functions.

(b) Divide the T universe into the fuzzy subsets of freezing cold (F), cold (C), warm (W), and hot (Ho), represented by equally distributed triangular membership functions.

As a solution, the following sets with triangular membership functions were formed:

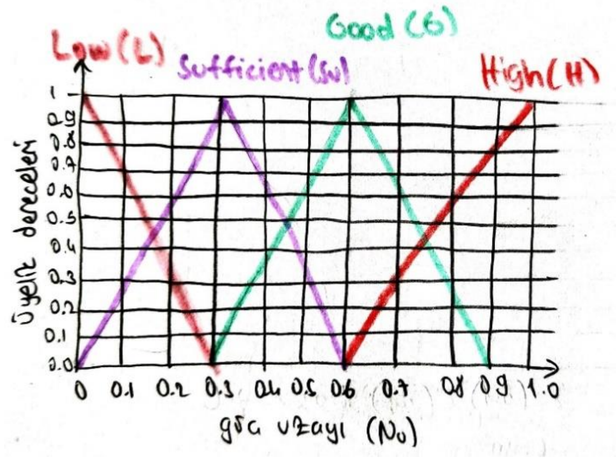


The sets created serve as **input sets** for the problem.

According to another example, by forming a power relationship based on the following explanation, an output space can be created, and by partitioning these, **output sets** can be formed.

The power generated by photovoltaic (PV) panels varies as a function of solar irradiance and operating temperature. Thus, the generated power increases as the solar irradiance level increases, but decreases as the operating temperature rises.

Since power is directly proportional to the level of solar irradiance, an attempt was made to create a space and sets using the values of solar irradiance levels.



The input and output sets of the problem, as well as the membership functions, which are the first stages of Mamdani, have been determined. After these stages, a rule table is created.

Based on the example we used for the output sets, we can establish the **rule table** as follows.

K	LI	SuI	GI	HI
FT	LP	SuP	GP	HP
CT	LP	SuP	HP	HP
WT	LP	SuP	HP	HP
HT	LP	SuP	HP	HP

Solar Radiation Sets

LI = low irradation

SuI = sufficient irradation

GI = good irradation

HI = high irradation

Temperature Sets

FT = freezing temperature

CT = cold temperature

WT = warm temperature

HT = hot temperature

Power Sets

LP = low power

SuP = sufficient power

GP = good power

HP = high power

For the main part of the problem, **active rules** are determined based on the inputs to our system:

System inputs = Solar irradiance level of 75 mW/cm² and temperature of 25°C

The value of 75 mW/cm² for solar irradiance corresponds to the Low and Sufficient sets in the irradiance space. The value of 25°C for temperature corresponds to the Warm and Hot sets in the temperature space. Therefore, the **active rules** in the table are,

K	LI	SuI	GI	HI
FT	LP	SuP	GP	HP
CT	LP	SuP	HP	HP
WT	LP	SuP	HP	HP
HT	LP	SuP	HP	HP

- *If temperature is WARM and irradation is LOW then power is LOW*
- *If temperature is WARM and irradation is SUFFICIENT then power is SUFFICIENT*
- *If temperature is HOT and irradation is LOW then power is LOW*
- *If temperature is HOT and irradation is SUFFICIENT then power is SUFFICIENT*

As the next step, **the inputs are fuzzified**:

For the input solar irradiance level of 75 mW/cm², the fuzzy value (membership value) is found as 0.5 for both the Low and Sufficient sets using a triangular membership function or triangular similarity.

For the temperature input of 25°C, the fuzzy value is found as 0.25 for the Hot set and 0.75 for the Warm set using the same methods.

Active rules are processed:

WT	And	LI	Then	LP
WT	And	SuI	Then	SuP
HT	And	LI	Then	LP
HT	And	SuI	Then	SuP

Fuzzy inference is performed:
Since there is an 'and' (ve) among the rules, the minimum of the inputs is taken, and the value of the result is written.

WT(0.75)	And	LI(0.5)	Then	LP(0.5)
WT(0.75)	And	SuI(0.5)	Then	SuP(0.5)
HT(0.25)	And	LI(0.5)	Then	LP(0.25)
HT(0.25)	And	SuI(0.5)	Then	SuP(0.25)

The **defuzzification** step is applied for the actual result:

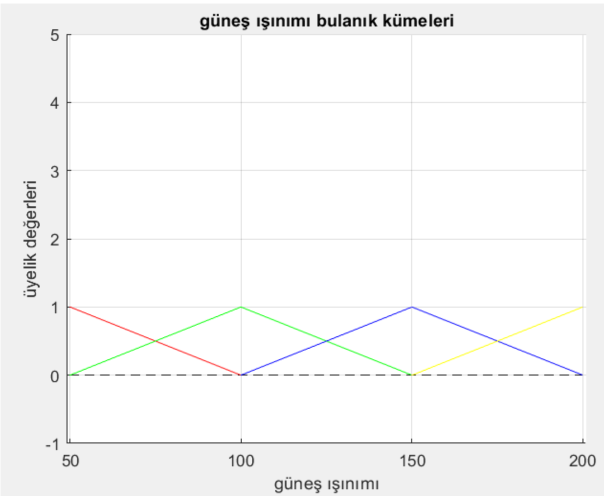
LP(0.5)	0.5	.	0	=	0
SuP(0.5)	0.5	.	0.3	=	0.15
LP(0.25)	0.25	.	0	=	0
SuP(0.25)	0.25	.	0.3	=	0.075
SUM	1.5				0.225

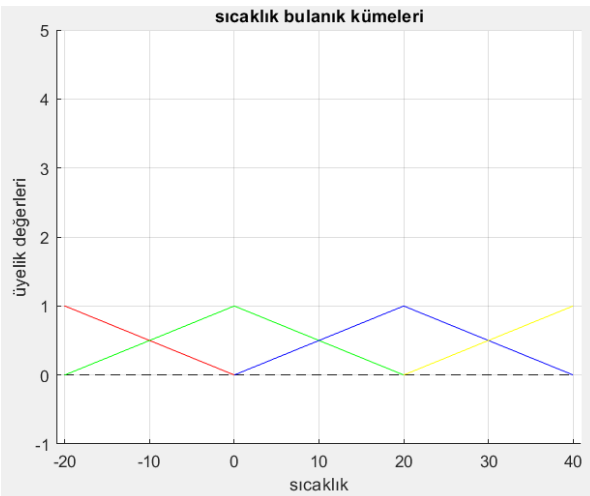
The defuzzification result using the centroid method is 0.225/1.5 = 0.15 pu.

As a result of the steps taken, when the solar irradiance level is 75 mW/cm² and the temperature is 25 °C, the power output is found to be 0.15 pu.

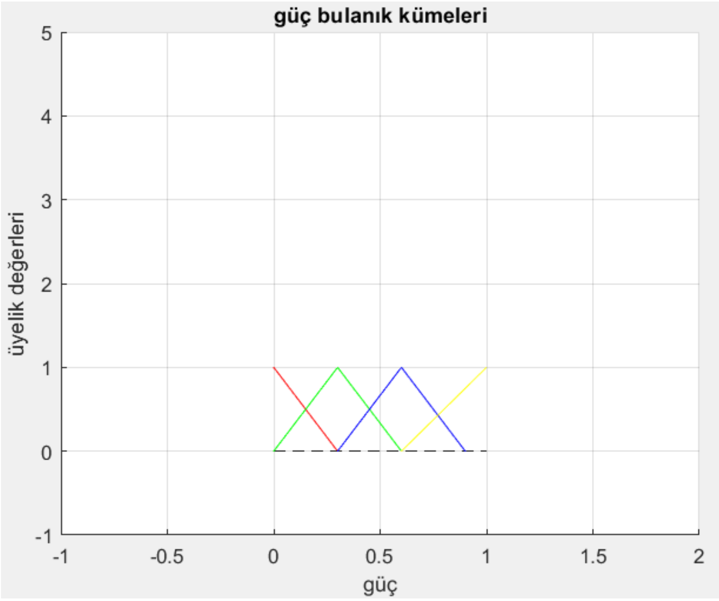
Solution for Part B:
In this section, let's demonstrate what we did manually above by coding it in MATLAB.

Input sets,





Output sets,



The step of defining the membership function (Here, the formula for the triangular membership function is coded within the calculateMembership function.)

```
function membership = calculateMembership(X,trianle)
    x1 = trianle(1);
    xT = trianle(2);
    x2 = trianle(3);

    a=(X-x1)/(xT-x1);
    b=(x2-X)/(x2-xT);
    c=0;
    membership =max(min(a,b),c);
    disp(membership);
    return
end
```

Rule Table,

```
% rules
%          LI    SuI    GI    HI
% FT    [  0     0.3    0.6   1
% CT       0     0.3    0.6   1
% WT       0     0.3    0.6   1
% HT       0     0.3    0.6   1  ]
rules = [
    0 0.3 0.6 1
    0 0.3 0.6 1
    0 0.3 0.6 1
    0 0.3 0.6 1
    ];
```

The outputs resulting from the fuzzification of the inputs,

```
% Sıcaklık
temperature = 25;
FT = [-20,-20,0];
CT = [-20,0,20];
WT = [0,20,40];
HT = [20,40,40];

% Güneş Işınımı
irradation = 75;
LI = [50,50,100];
SuI = [50,100,150];
GI = [100,150,200];
HI = [150,200,200];

disp("Sıcaklık kümeleri üyelik derecesi");
calculateMembership(temperature,FT);
calculateMembership(temperature,CT);
calculateMembership(temperature,WT);
calculateMembership(temperature,HT);
disp("Işınım kümeleri üyelik derecesi");
calculateMembership(irradation,LI);
calculateMembership(irradation,SuI);
calculateMembership(irradation,GI);
calculateMembership(irradation,HI);
```

```
Command Window
>> rules_and_fuzzier
Sıcaklık kümeleri üyelik derecesi
     0

     0

    0.7500

    0.2500

Işınım kümeleri üyelik derecesi
    0.5000

    0.5000

     0

     0
```

Rule processing,

```
% kural işleme,
%           MuLI MuSuI MuGI MuHI
% MuFT    [
% MuCT       min()
% MuWT
% MuHT          ]
MU = [ min(MuFT, MuLI) min(MuFT, MuSuI) min(MuGI, MuGI) min(MuFT, MuHI)
    min(MuCT, MuLI) min(MuCT, MuSuI) min(MuCT, MuGI) min(MuCT, MuHI)
    min(MuWT, MuLI) min(MuWT, MuSuI) min(MuWT, MuGI) min(MuWT, MuHI)
    min(MuHT, MuLI) min(MuHT, MuSuI) min(MuHT, MuGI) min(MuHT, MuHI) ]

% önerme sonuçları
MUK = [MU(1,1)*K(1,1) MU(1,2)*K(1,2) MU(1,3)*K(1,3) MU(1,4)*K(1,4)
    MU(2,1)*K(2,1) MU(2,2)*K(2,2) MU(2,3)*K(2,3) MU(2,4)*K(2,4)
    MU(3,1)*K(3,1) MU(3,2)*K(3,2) MU(3,3)*K(3,3) MU(3,4)*K(3,4)
    MU(4,1)*K(4,1) MU(4,2)*K(4,2) MU(4,3)*K(4,3) MU(4,4)*K(4,4) ]
```

```
MU =

         0         0         0         0
         0         0         0         0
    0.5000    0.5000         0         0
    0.2500    0.2500         0         0


MUK =

         0         0         0         0
         0         0         0         0
         0    0.1500         0         0
         0    0.0750         0         0
```

Defuzzification step,

```
% Durulaştırma, alanların toplamı yöntemi
TOP1 = 0;
TOP2 = 0;
for n=1:3
    for m=1:3
        TOP1 = TOP1 + MUK(n,m);
        TOP2 = TOP2 + MU(n,m);
    end
end
z = TOP1/TOP2
```

```
Sonuç:

z =

    0.1500
```

As can be seen, we obtained the same results manually and using MATLAB with the Mamdani Fuzzy Modeling. When the solar irradiance level is 75 mW/cm² and the temperature is 25 °C, the power output will be observed as 0.15 pu.

MATLAB codes that graphically show the input and output sets,

```
% solar radiation space and set
% X = [50,60,70,80,90,100,110,120,130,140,150,160,170,180,190,200];
% A = [50,50,100];
% B = [50,100,150];
% C = [100,150,200];
% D = [150,200,200];

% Temperature space and sets
% X = [-20,-10,0,10,20,30,40];
% A = [-20,-20,0];
% B = [-20,0,20];
% C = [0,20,40];
% D = [20,40,40];

% Power space and sets
X = [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1];
A = [0,0,0.3];
B = [0,0.3,0.6];
C = [0.3,0.6,0.9];
D = [0.6,1,1];

draw_triangles(X, A, B, C,D);

function draw_triangles(X, A, B, C, D)
    % X: Space
    % U : Set Lists
    U = [A;B;C;D];

    figure;
    hold on;
    plot(X, zeros(size(X)), 'k--'); % X axis
    colors = {'r', 'g', 'b', 'y'};

    for i = 1:length(U)
```

```
        x = zeros(1,3);
        y = zeros(1,3); % Y vector to 0
        for j = 1:3
            x(j) = U(i,j);
        end
        if i == 1 % Right boundary triangle
            y = [1,1,0];
        elseif i == length(U)
            y = [0,1,1]; %  Left boundary triangle
        else
            y = [0,1,0]; % Middle Triangles
        end

        plot(x, y, colors{i}); % triangle i
    end
    % xlabel('Solar Radiation);
    % xlabel('Temperature');
    xlabel('Power');
    ylabel(' Membership Degree');
    % title(' Solar Radiation Fuzzy Sets');
    % title(' Temperature Fuzzy Sets');
    title(' Power Fuzzy Sets');

    axis([min(X)-1 max(X)+1 -1 5]);
    grid on;
    hold off;
end
```

MATLAB codes that contains Mamdani Fuzzy Models Steps

```
% Temperature
temperature = 25;
FT = [-20,-20,0];
CT = [-20,0,20];
WT = [0,20,40];
HT = [20,40,40];

% Solar Radiation
irradation = 75;
LI = [50,50,100];
SuI = [50,100,150];
GI = [100,150,200];
HI = [150,200,200];

disp("Temperature Sets Membership Degree");
MuFT = calculateMembership(temperature,FT);
MuCT = calculateMembership(temperature,CT);
MuWT = calculateMembership(temperature,WT);
MuHT = calculateMembership(temperature,HT);
disp("Solar Radiation Sets Membership Degree");
MuLI = calculateMembership(irradation,LI);
MuSuI = calculateMembership(irradation,SuI);
MuGI = calculateMembership(irradation,GI);
MuHI = calculateMembership(irradation,HI);

% Rules
%         LI    SuI    GI    HI
% FT    [  0     0.3    0.6   1
% CT       0     0.3    0.6   1
```

```
% WT      0    0.3   0.6  1
% HT      0    0.3   0.6  1  ]
K = [
    0 0.3 0.6 1
    0 0.3 0.6 1
    0 0.3 0.6 1
    0 0.3 0.6 1
    ];

% Rules Processing,
%          MuLI MuSuI MuGI MuHI
% MuFT   [
% MuCT      min()
% MuWT
% MuHT        ]
MU = [ min(MuFT, MuLI) min(MuFT, MuSuI) min(MuGI, MuGI) min(MuFT, MuHI)
    min(MuCT, MuLI) min(MuCT, MuSuI) min(MuCT, MuGI) min(MuCT, MuHI)
    min(MuWT, MuLI) min(MuWT, MuSuI) min(MuWT, MuGI) min(MuWT, MuHI)
    min(MuHT, MuLI) min(MuHT, MuSuI) min(MuHT, MuGI) min(MuHT, MuHI) ]

% Results
MUK = [MU(1,1)*K(1,1) MU(1,2)*K(1,2) MU(1,3)*K(1,3) MU(1,4)*K(1,4)
    MU(2,1)*K(2,1) MU(2,2)*K(2,2) MU(2,3)*K(2,3) MU(2,4)*K(2,4)
    MU(3,1)*K(3,1) MU(3,2)*K(3,2) MU(3,3)*K(3,3) MU(3,4)*K(3,4)
    MU(4,1)*K(4,1) MU(4,2)*K(4,2) MU(4,3)*K(4,3) MU(4,4)*K(4,4) ]

% Defuzzification (the method of total areas)
TOP1 = 0;
TOP2 = 0;
for n=1:3
    for m=1:3
        TOP1 = TOP1 + MUK(n,m);
        TOP2 = TOP2 + MU(n,m);
    end
end
disp("Result: ")
z = TOP1/TOP2

function membership = calculateMembership(X,trianle)
    x1 = trianle(1);
    xT = trianle(2);
    x2 = trianle(3);

    a=(X-x1)/(xT-x1);
    b=(x2-X)/(x2-xT);
    c=0;
    membership =max(min(a,b),c);
    disp(membership);
    return
end
```