

Bulanık Mantık Tabanlı Anlık Hava Sıcaklığı ve Toprak Nemi Verileri Kullanarak Sulama Yapacak Pompa Motorunun Devir Hızı ve Çalışma Süresinin Tahmini

Baybars SERENER - 214217022

Mekatronik Mühendisliği

Teknoloji Fakültesi / Afyonkocatepe Üniversitesi

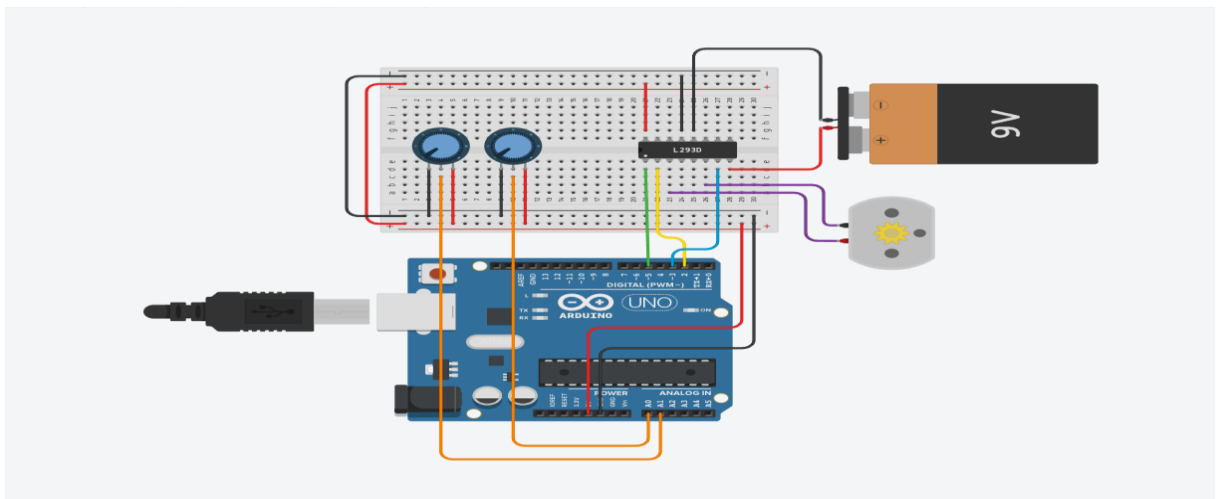
baybars.serener@usr.aku.edu.tr

1.Giriş

Bu çalışma, Ardino ve Matlab kullanılarak gerçekleştirilen bulanık mantık kontrolü ve ölçümler sayesinde, anlık hava sıcaklığı ve toprak nemi verilerini kullanarak tarımsal sulama için pompa motorunun devir hızını ve çalışma süresini tahmin etmeyi amaçlamaktadır.

2.Modelin Donanımı

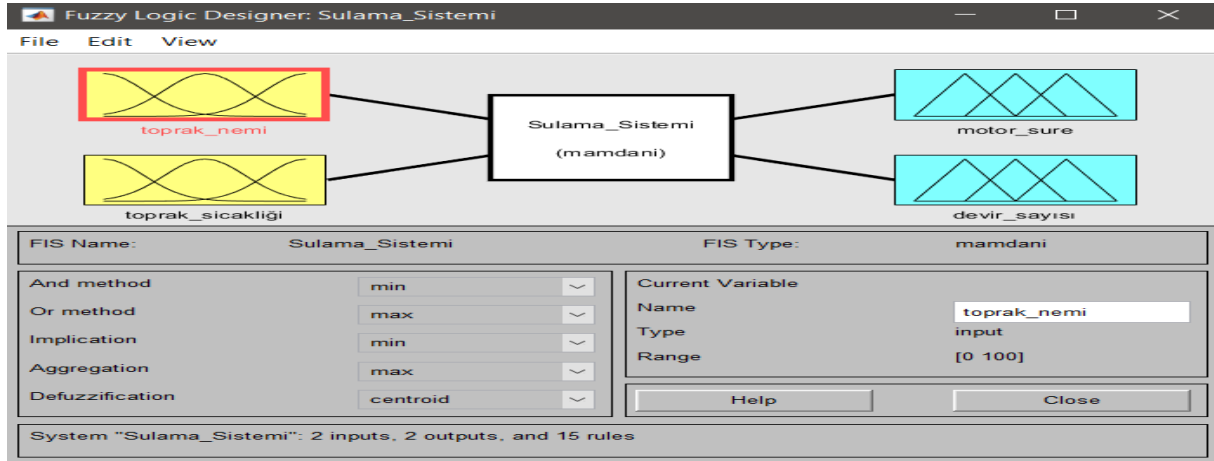
Modelin kontrolü için Arduino kullanılmış olup, modelin bağlantıları ve kullanılan donanımlar Şekil 1'de gösterilmiştir. Modelde tek bir motor kullanılmıştır. Arduino'ya bağlı potansiyometrelerden alınan toprak sıcaklığı ve toprak nem değerleri seri haberleşme ile MATLAB'a gönderilmiş ve ardından motorun devir hızı ve çalışma süresi için bulanık mantık çıkışlarını Arduino'ya aktarılmıştır. Bu süreç, toprak sıcaklığı ve nem değerlerinin bulanık mantık kontrol algoritmaları kullanılarak işlenmesini ve ardından motorun çalışma parametrelerinin belirlenmesini içerir. Bu şekilde, Arduino üzerinde gerçek zamanlı bir kontrol sağlanmış olur.



Şekil 1 _ Devre Modeli

3. Sulama için Bulanık Mantık Kontrol Sistemi Tasarımı ve Uygulaması

Oluşturulan Bulanık Mantık Kontrolü, iki giriş ve iki çıkıştan oluşmaktadır. Birinci giriş toprak sıcaklığı değerini, ikincisi ise toprak nemi değerini temsil etmektedir. Bu iki değer potansiyometrelerden elde edilmektedir. Çıkış olarak ise, sulama motorunun devir sayısı ve motorun çalışma süresi bulunmaktadır. Bulanık mantık kontrolü, Şekil 2'de gösterildiği şekilde tasarlanmıştır, girişleri ve çıkışları için üçgen üyelik fonksiyonları kullanılmıştır. Bu fonksiyonların değişkenleri aşağıda detaylı bir şekilde gösterilmiştir. Durulaştırıcı işlemlerde ise AND (min) bileşimsel çıkarım kuralı ve Mamdani ağırlık merkezi yöntemi kullanılmıştır.

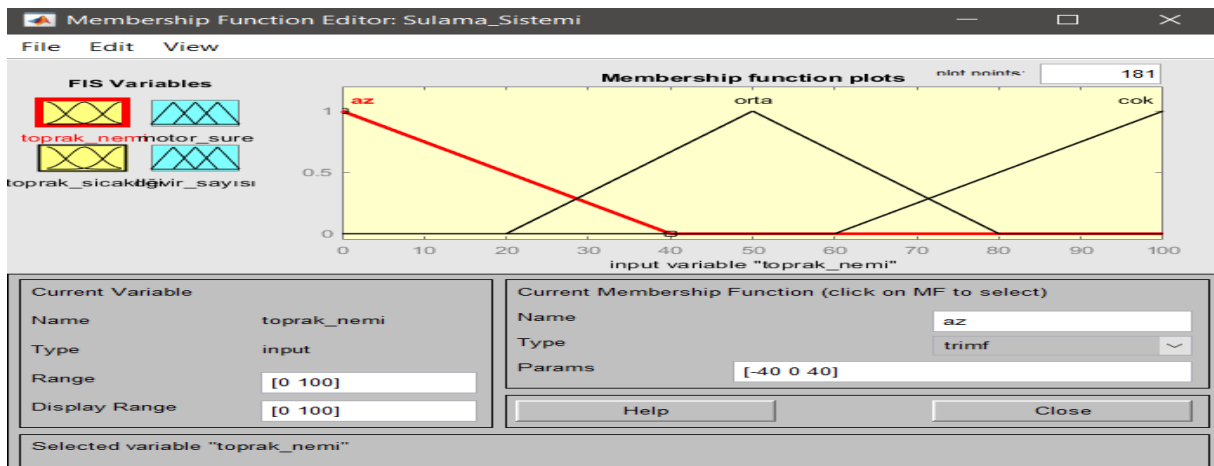


3.1) Giriş değişkeni olan toprak nemi (0 100) aralığındadır. Değerleri;

Az = [-40 0 40] (trimf)

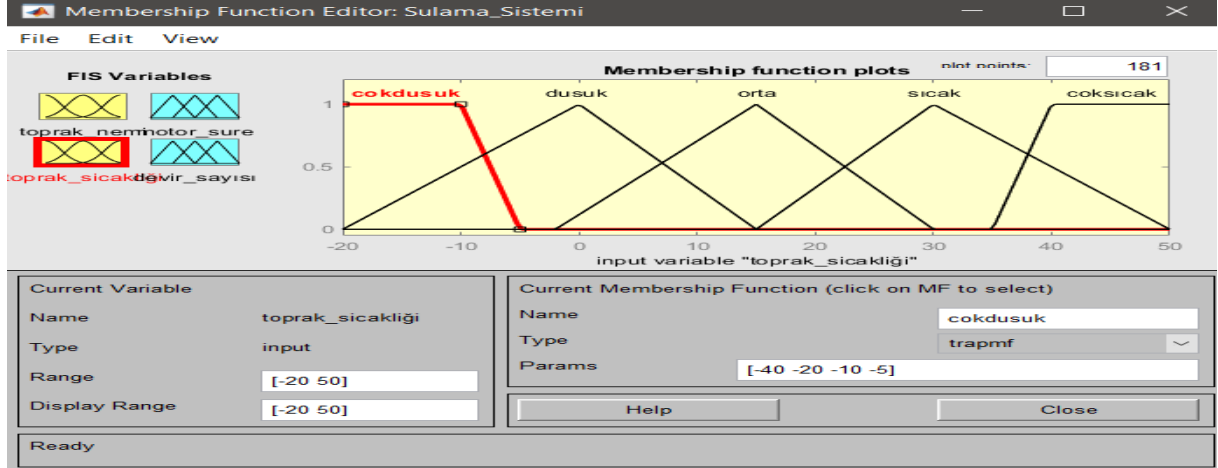
Orta = [20 50 80] (trimf)

Çok = [60 100 140] (trimf)



3.2) Giriş değişkeni olan toprak sıcaklığı (-20 50) aralığındadır. Değerleri;

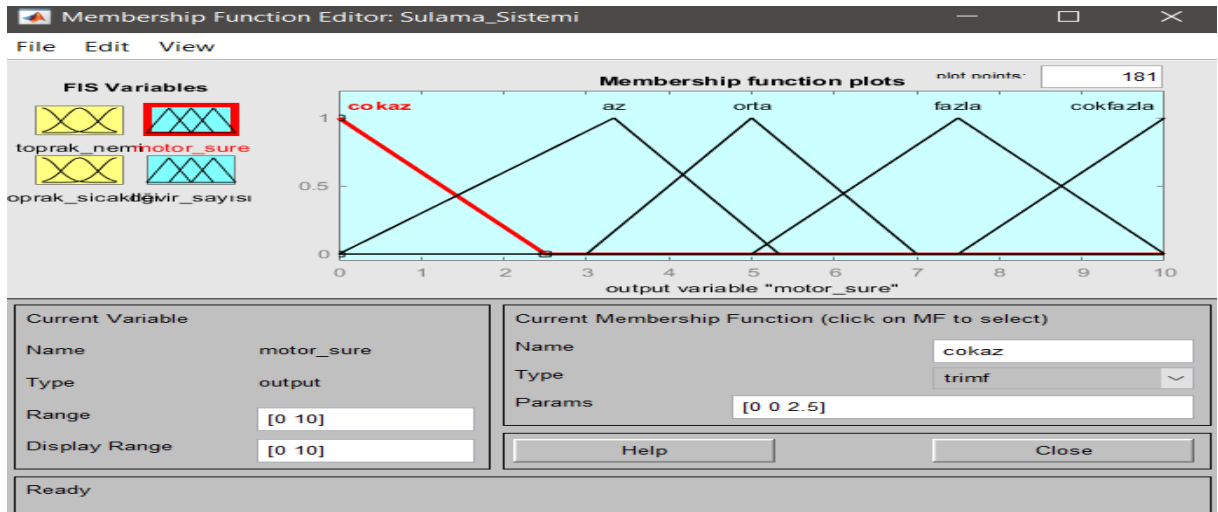
Çok düşük = [-40 -20 -10 -5] (trapmf)
Düşük = [-20 0 15] (trimf)
Orta = [-2 15 30] (trimf)
Sıcak = [-40 0 40] (trimf)
Çok sıcak = [15 30 50] (trapmf)



Şekil 4_ Toprak sıcaklığı giriş değişkeninin bulanık mantık üyelik fonksiyonları

3.3) Çıkış değişkeni olan motor süresi (0 10) aralığındadır. Değerleri;

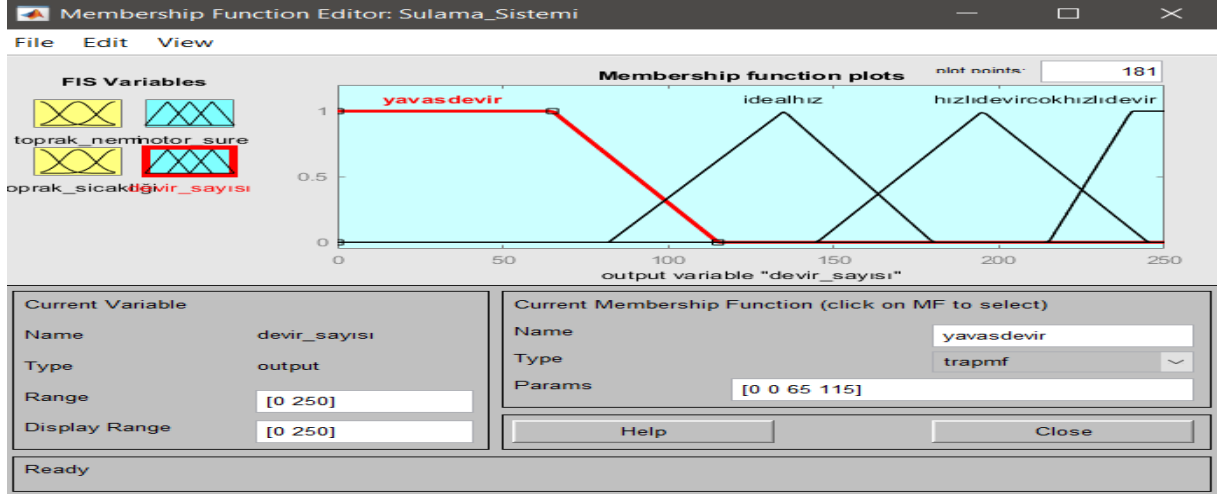
Çok az = [0 0 2.5] (trimf)
Az = [0 3.333 5.333] (trimf)
Orta = [3 5 7] (trimf)
Fazla = [5 7.5 10] (trimf)
Çok fazla = [7.5 10 16.67] (trimf)



Şekil 5_ Motor süresi çıkış değişkeninin bulanık mantık üyelik fonksiyonları

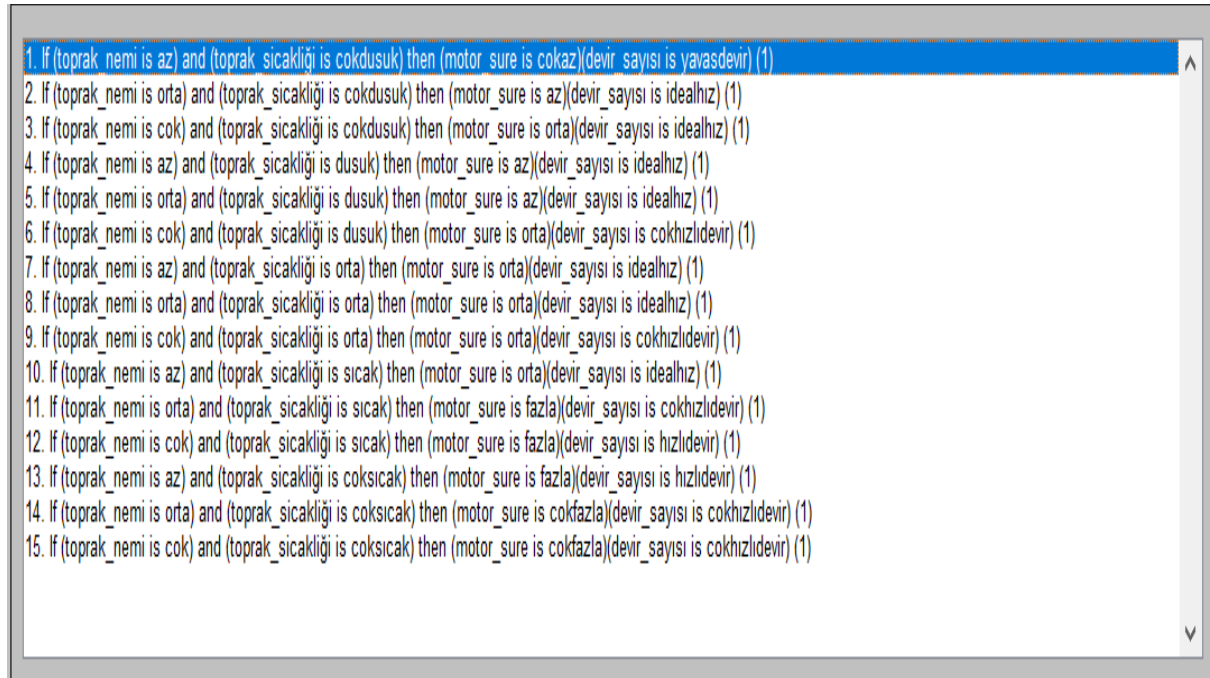
3.4) Çıkış değişkeni olan motor devri (0 250) aralığındadır. Değerleri;

Yavaş devir = [0 0 65 115] (trapmf)
Orta devir = [82 135 180] (trimf)
Hızlı devir = [145 195 245] (trimf)
Çok hızlı devir = [215 240 250 280] (trapmf)

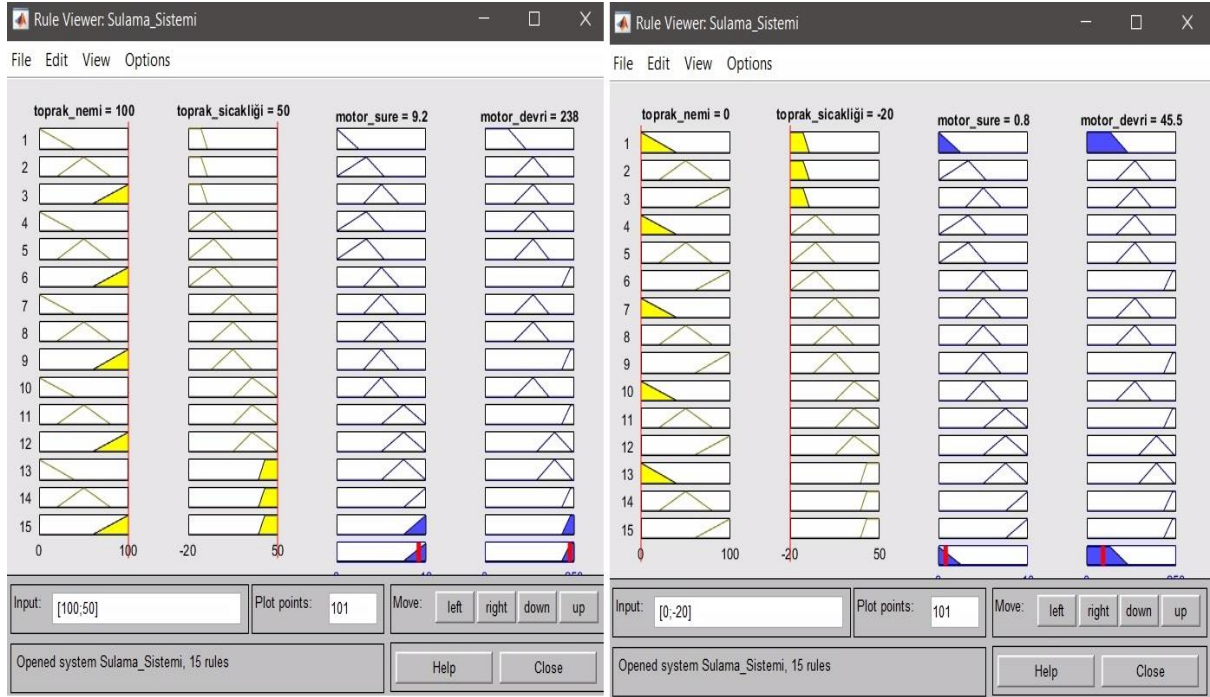


Şekil 6_ Devir sayısı çıkış değişkeninin bulanık mantık üyelik fonksiyonları

3.5) Optimum değerleri almak için oluşturulan kullar;



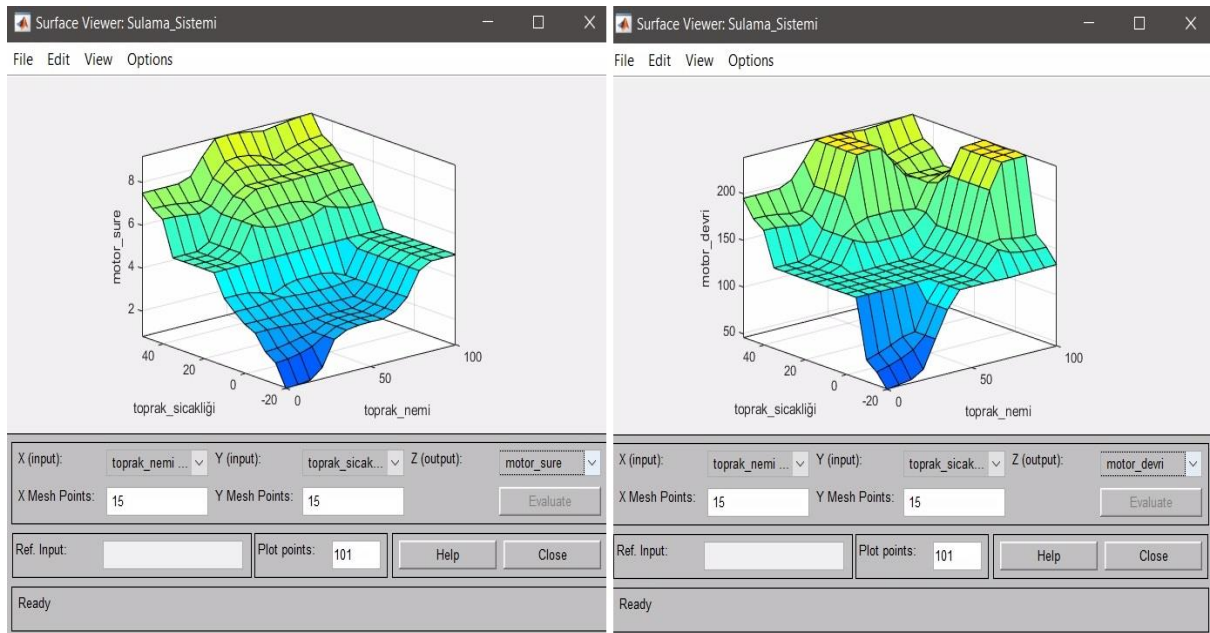
Şekil 7_ Bulanık mantık kural listesi



Şekil 8_ Bulanık mantık kural tablosu

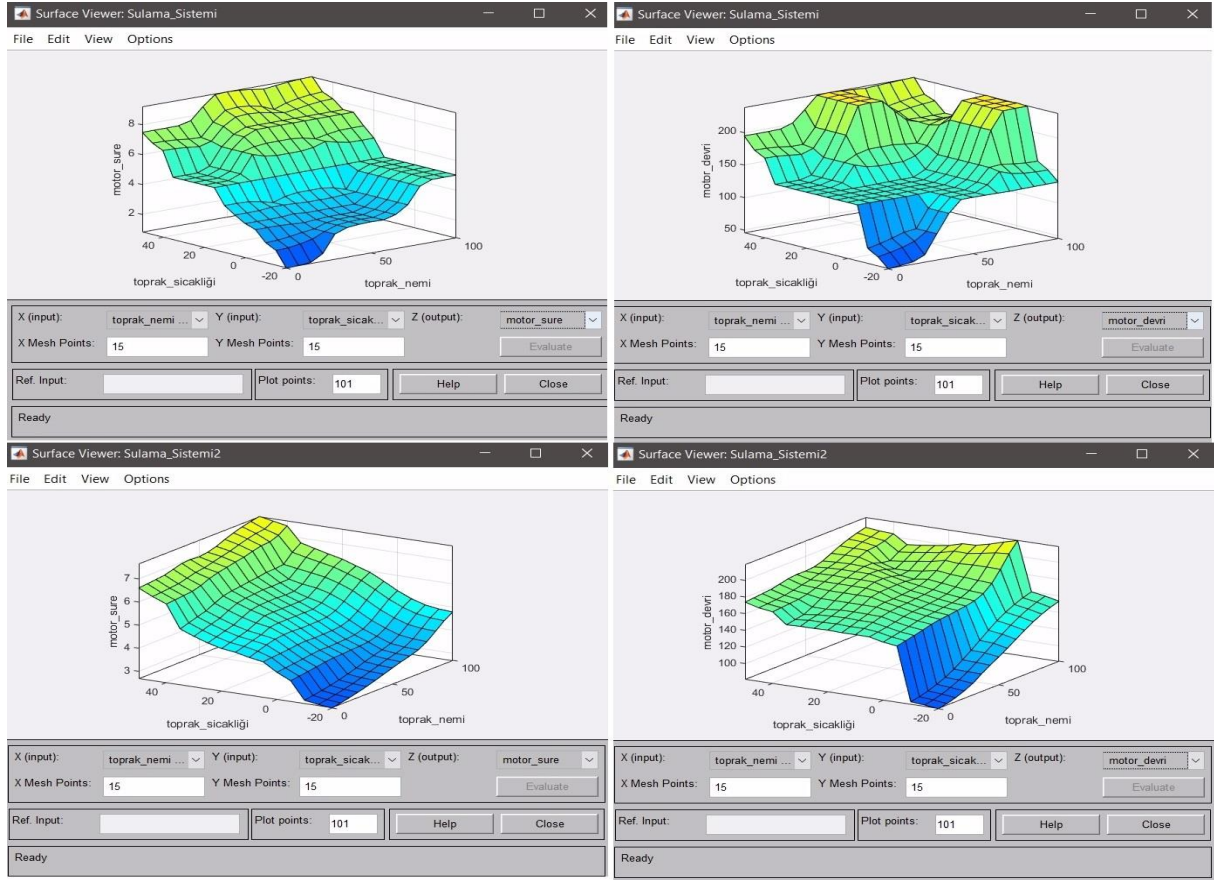
3.6) Optimum kurallar ile oluşturulan yüzeyler;

Yüzey, bulanık mantık sistemlerinde giriş değişkenlerinin birleşimini ifade eder ve sistemin nasıl tepki vereceğini belirlemek için kullanılır. Bulanık mantıkta, her bir yüzey için bir çıkış değişkeni değeri belirlenir. Yüzeyler, genellikle bulanık mantık sistemlerinin kurallarını tanımlamak ve sistemin nasıl tepki vereceğini belirlemek için kullanılır.



Şekil 9_ Motor süresi ve motor devrinin üç boyutlu yüzey grafikleri

4. Üçgen MF/ Gauss MF Karşılaştırması



Şekil 10_ Üçgen MF (üst) - Gauss MF (alt) Karşılaştırması

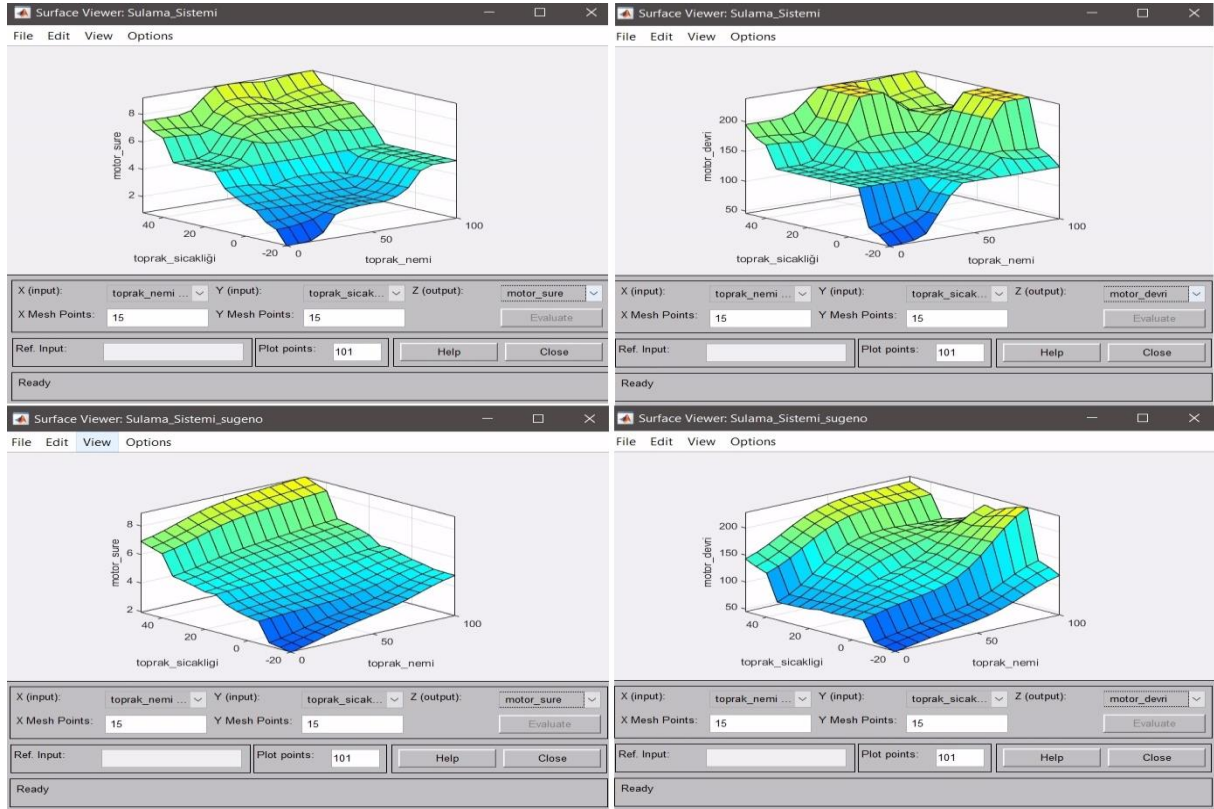
Gaussmf kullanımıyla;

1. Çıkış değerleri daha düzgün oldu. Trimf ve trapmf kullanıldığında, çıkış değerleri daha keskindi, gaussmf kullanımı daha yumuşak bir çıkış grafik sağladı.
2. Gaussmf ile, giriş değerlerinin çıkışa olan etkisi daha yumuşak bir geçiş sağladı. Trimf ve trapmf kullanıldığında, geçişler daha belirgindi.
3. Gaussmf, daha yüksek bir hassasiyet seviyesi sundu. Özellikle, giriş değerleri arasında keskin geçişlerin olmadığı durumlarda gaussmf kullanımı, daha doğru sonuçlar elde etmemizi sağladı.

Fakat gaussmf'nin daha karmaşık bir matematiksel yapıya sahip olması nedeniyle, hesaplama maliyeti artabilir. Trimf ve trapmf daha basit hesaplamalarla çalışırken, gaussmf'nin daha fazla hesaplama gücü gerektirir.

5. Mandani ile Sugeno Karşılaştırması

Bulanık mantıkta, Mandani ve Sugeno iki farklı çıkarım yöntemidir. Mandani çıkarımı, her kuralın sonucunda bir bulanık çıktı elde edilir ve bu çıktılar birleştirilerek sonuç bulanık çıktı oluşturulur. Sugeno çıkarımı ise kural sonuçlarının belirli bir formül ile ağırlıklandırılmasıyla kesin bir çıktı elde edilmesini sağlar.



Şekil 10_ Mamdani (üst) - Sugeno (alt) Karşılaştırması

Eğer bu kodu Sugeno ile oluştursaydık, çıktıların kesin değerler alacağı bir model elde edilecekti. Özellikle, çıktılar doğrudan bir matematiksel formül kullanılarak ifade edilirdi, bu durumda, sistemin davranışı daha belirgin ve genellikle daha hızlı bir şekilde yorumlanabilir olurdu. Mandani mantığına göre ise çıktılar bulanık kümeler olarak ifade edilir ve bu kümelerin birleştirilmesi sonucunda genellikle daha esnek, ancak daha yorumlanması karmaşık olan bir model elde edilir.

6. Bulanık Mantık Kodları

6.1) Ardıuno Kodu

```
#define ADC0 A0 // ADC0, analog 0 pinini temsil ediyor.
#define ADC1 A1 // ADC1, analog 1 pinini temsil ediyor.

int MOTOR_PIN_1 = 2; // L239D sürücüsü 1. pin (yön)
int MOTOR_PIN_2 = 3; // L239D sürücüsü 2. pin (yön)
int MOTOR_SPEED_PIN = 5; // L239D sürücüsü enable pin
float devir;
float sure;

void setup() {

    Serial.begin(9600);

    pinMode(MOTOR_PIN_1, OUTPUT);
    pinMode(MOTOR_PIN_2, OUTPUT);
    pinMode(MOTOR_SPEED_PIN, OUTPUT);
}
void loop() {

    Serial.print(analogRead(ADC0));
    Serial.print(analogRead(ADC1));
    delay(100);

    if (Serial.available() > 0) {

        Serial.readBytes((char*)&devir, sizeof(devir));

        Serial.readBytes((char*)&sure, sizeof(sure));

        analogWrite(MOTOR_SPEED_PIN, map(devir, 0, 100, 0, 255));
        digitalWrite(MOTOR_PIN_1, HIGH);
        digitalWrite(MOTOR_PIN_2, LOW);
        analogWrite(MOTOR_SPEED_PIN, devir);

        unsigned long baslangicZamani = millis();
        while (millis() - baslangicZamani < sure * 1000) {

        }

        digitalWrite(MOTOR_PIN_1, LOW);
        digitalWrite(MOTOR_PIN_2, LOW);

        delay(2000);
    }
}
```


6.2) Matlab Bulanık Mantık Kodu

```
clear;clc;

b = newfis ('Sulama_Sistemi');
b = addvar(b, 'input', 'toprak_nemi', [0 100]);
b = addmf(b, 'input', 1, 'az', 'trimf', [-40 0 40]);
b = addmf(b, 'input', 1, 'orta', 'trimf', [20 50 80]);
b = addmf(b, 'input', 1, 'cok', 'trimf', [60 100 140]);

b = addvar(b, 'input', 'toprak_sicakligi', [-20 50]);
b = addmf(b, 'input', 2, 'cokdusuk', 'trapmf', [-40 -20 -10 -5]);
b = addmf(b, 'input', 2, 'dusuk', 'trimf', [-20 0 15]);
b = addmf(b, 'input', 2, 'orta', 'trimf', [-2 15 30]);
b = addmf(b, 'input', 2, 'sicak', 'trimf', [15 30 50]);
b = addmf(b, 'input', 2, 'coksicak', 'trapmf', [35 40 50 50]);

b = addvar(b, 'output', 'motor_sure', [0 10]);
b = addmf(b, 'output', 1, 'cokaz', 'trimf', [0 0 2.5]);
b = addmf(b, 'output', 1, 'az', 'trimf', [0 3.333 5.333]);
b = addmf(b, 'output', 1, 'orta', 'trimf', [3 5 7]);
b = addmf(b, 'output', 1, 'fazla', 'trimf', [5 7.5 10]);
b = addmf(b, 'output', 1, 'cokfazla', 'trimf', [7.5 10 16.67]);

b = addvar(b, 'output', 'motor_devri', [0 250]);
b = addmf(b, 'output', 2, 'yavasdevir', 'trapmf', [0 0 65 115]);
b = addmf(b, 'output', 2, 'ortadevir', 'trimf', [82 135 180]);
b = addmf(b, 'output', 2, 'hizlidevir', 'trimf', [145 195 245]);
b = addmf(b, 'output', 2, 'cokhizlidevir', 'trapmf', [215 240 250 280]);

rulkey = [
    1 1 1 1 1 1;
    2 1 2 2 1 1;
    3 1 3 2 1 1;
    1 2 2 2 1 1;
    2 2 2 2 1 1;
    3 2 3 4 1 1;
    1 3 3 2 1 1;
    2 3 3 2 1 1;
    3 3 3 4 1 1;
    1 4 3 2 1 1;
    2 4 4 4 1 1;
    3 4 4 3 1 1;
    1 5 4 3 1 1;
    2 5 5 4 1 1;
    3 5 5 4 1 1
];

b = addrule(b, rulkey); writefis(b, 'Sulama_Sistemi.fis');
```

6.3) Ardiuno ile Haberleşme Matlab Kodu

```
clc;
clear;
close all;
delete(instrfindall);
g = serial('COM3', 'BaudRate', 9600);
fopen(g);
b = readfis('D:\Allahin Adami Program File\MathLab\bin\sulama_sistemi.fis'); %
Oluşturulan bulanık mantık kontrol sistemi çağırıldı.
x = 1;
while x
    sicaklik=fscanf(g,'%s');
    s=strsplit(sicaklik, '.');
    sicaklik=str2double(s(2));
    sicaklik=sicaklik*5/1024; %gelen değer 0-1024 aralığından 0-5 aralığına
    çekildi.
    nem=str2double(s(1));
    nem=nem*5/1024;%gelen değer 0-1024 aralığından 0-5 aralığına çekildi.
    sicaklik(x)=14*sicaklik-20;%pot'tan gelen değer (-20,50) arasına uyarlanı.
    nem(x)=20*nem;%sensörden gelen değer (0,100) arasına uyarlandı.
    y = evalfis(b, [nem(x) sicaklik(x)]); % belirtilen girişlere (nem(x) ve
    sicaklik(x)) dayalı olarak bulanık mantık modelinin çıkışları hesaplandı.
    sure(x)=y(2);
    devir(x)=y(1);
    fprintf('Sıcaklık=%f Nem=%f Süre= %f Devir=%f\n',sicaklik(x),nem(x),y(1),
    y(2));
    fwrite(g, [sure(x)], 'float');
    fwrite(g, [devir(x)], 'float');
    x=x+1;
end
```

7. Sonu

Bu alıřmada, Arduino ve MATLAB kullanılarak gerekleřtirilen bulanık mantık kontrol ve lmler aracılıėıyla anlık hava sıcaklıėı ve toprak nemi verilerini kullanarak tarımsal sulama iin pompa motorunun devir hızını ve alıřma sresini tahmin etmeyi amaladık.

Sonu olarak, bu alıřma bulanık mantık kontroll bir sulama sistemi tasarlamak iin bir model sunmaktadır. Bu model, tarımsal sulama sistemlerinde suyun daha verimli kullanılmasına ve bitkilerin ihtiya duyduėu su miktarının daha doėru bir řekilde belirlenmesine yardımcı olabilir.