



**Karadeniz Teknik Üniversitesi**  
**Mühendislik Fakültesi**  
**Bilgisayar Mühendisliği Bölümü**



# **BULANIK MANTIK**

## **DÖNEM PROJESİ**

**2019-2020 Bahar Dönemi**

<b>Öğretim</b>	1. Öğretim <input checked="" type="checkbox"/>	2. Öğretim <input type="checkbox"/>
<b>Numara</b>	330104	
<b>Ad ve Soyad</b>	Onur ERDAŞ	
<b>Dersin Sorumlusu</b>	Prof. Dr. İsmail Hakkı Çavdar	

# ÖNSÖZ

Bu proje ile Mamdani Bulanık Sonuçlandırma Algoritmasının çalışmasını öğrendim. Durulaştırma yöntemi ile bulunan kesin sayı çözümünün doğruluğunu öğrendim. Bu algoritma ile bulanık girişlere sahip uzayların kesin sonuç çıkarabileceğini öğrendim. Runge-Kutta Algoritması ile denklem sisteminin çözülmesini öğrendim. Ayrıca bulanık mantık denetleyicisiyle bir sistemin tasarımını ve simülasyonunu nasıl yapabileceğimi öğrendim. Bulanık mantık denetleyicisi oluşturmayı öğrendim ve hata değerlerine göre sistemin verdiği tepkileri gözlemledim. Bulanık mantık denetleyicilerinin çalışmasını öğrendim ve bulanık mantık denetleyicileri için sonuçların nasıl iyileştirilebileceğini öğrendim.

Bu projeyi verdiğinden ve değerli kaynaklarını bize sunduğundan dolayı sayın Prof. Dr. İsmail Hakkı Altaş'a teşekkür ederim.

## ETİK DAVRANIŞ VE MESLEKİ DÜRÜSTLÜK

Adil ve doğru değerlendirme eğitimin temelinde bulunması gereken temel etkenlerdir. Bir öğrenci olarak, kendinize ya da başka bir öğrenciye haksız avantaj yaratacak herhangi bir hareket, akademik sahtekârlık olarak değerlendirilir. Başkasının çalışmasını kopyalamak ve kendisininmiş gibi göstermek de akademik sahtekârlıktır. Buna aşırma denir.

Başka birinin sizin çalışmanızı kopyalamasına ve bunu kendi çalışması gibi göstermesine izin vermek de aynı zamanda sahtekârlıktır. Her türlü haksız kazanç ve avantaj sağlamak hem yanlış hem de etik dışıdır. Yardımlaşma sınırını aşıp başka bir öğrencinin ödevini, projesini veya laboratuvar raporunu onun adına hazırlamak da yanlış ve etik dışı bir davranıştır. Aynı şekilde kendi ödevini, projesini veya laboratuvar raporunu başkasına yaptırmak ve yazdırmak da tam bir aldatmaca ve ahlaksızlıktır.

Başka bir öğrencinin cevaplarını, ödevini veya raporunu onun bilgisi dışında kullanmak ve avantaj sağlamak da o öğrencinin emeğini çalmaktır ve buna hırsızlık denir. Sınavlarda kopya çekerek sınıf ortalamasının üzerine çıkmak da aynı şeydir.


Unutmayalım ki öğrenciyken doğruyu öğrenemeyenler mezun olduktan sonra da bu yanlışlara devam eder ve toplumda hep şikâyet ettiğimiz karakterlerden biri olur.

Biz öğrencilerimizin ve mezunlarımızın sorumluluk bilincine sahip, mesleki olduğu kadar insani etik davranışlara da sahip olmasını bekliyor ve sizden bu bu formu imzalayarak bize söz vermenizi istiyoruz.

Hadi gelin hep birlikte etik dışı davranışlarla mücadele edelim.

Prof. Dr. İsmail H. ALTAŞ

Yukarıda yazılı olanların etik kurallar çerçevesinde doğru olduğuna inanıyorum. Bu kurumdaki eğitimim süresince ve sonrasında etik bilinciyle davranacağıma söz veriyorum.

Öğrenci No	Öğrenci Adı ve Soyadı	Tarih	İmza
330104	ONUR ERDAŞ	03/06/2020	

## ÖZET

Projenin ilk kısmı özetle bulanık giriş olarak verilen 3 uzay için kesin bir karar vermektir. Bulanık girişlere sahip 3 uzayın gerekli işlemlerden geçirilerek çıkışına karar verilmesi gerekmektedir. Öğrenci sayısı, güncel uygulanabilirlik değeri ve teorik altyapısı değeri ile öğrencinin devam ettirilmesi durumu araştırılmaktadır. Öncelikle giriş uzayları tanımlandı ve bir kural tablosu oluşturuldu. Sonrasında oluşturulan bulanık kural sayısı kadar üyelik derecesi elde edildi. Elde edilen bu üyelik derecelerinin minimumu alındı. Minimumları alınan matris ile kural matrisi çarpıldı ve yeni bir matris oluşturuldu. Oluşturulan bu yeni matrisin elemanlarının toplamı, minimum alınan matrisin elemanlarının toplamına bölündü. Bu şekilde durulaştırma işlemi gerçekleştirdi ve kesin bir çıkış değeri elde edildi.

Projenin ikinci kısmında verilen problem bir sistemin herhangi bir denetleyici olmadan zamana göre davranışını elde etmektir. Durum denklemlerinin sayısal çözümünün Runge-Kutta algoritması ile bulunduktan sonra bu çözümün zamana göre çizdirilmesi istenmektedir. Çizdirilen bu grafiğin incelenmesi ve sabit bir değere oturup oturmadığının araştırılması istenmektedir. Sisteme giriş olarak verilen matris çarpımları 4 farklı aşamada yapıldı ve bu aşamaların sonuçları toplandı ve y değeri hesaplandı. Hesaplanan y değeri t zaman değerine göre çizdirildi ve herhangi bir sabit değere oturup oturmadığı gözlemlendi.

Projenin üçüncü kısmında, ikinci kısımdaki sistemin  $x_1$  çıkış değerini kontrol edecek bulanık mantık denetleyicisinin tasarlanması ve simülasyonunun yapılması istenmektedir. Problem sistem tarafından Runge-Kutta Algoritması ile hesaplanan y değerini , Mamdani Bulanık Sonuçlandırma Algoritmasını kullanarak hata hesaplamasını yapıp denetleyici ile kontrolü sağlanarak referans değerinde sabitlemektir. Hesaplanan y değerinin referans değerine ulaşması ile hata değeri 0'a indirilmektedir. Öncelikle giriş uzayları tanımlandı ve üyelik dereceleri hesaplanıp Mamdani Bulanık Sonuçlandırma Algoritması ile durulaştırılması yapıldı. Çıkan kesin sonuç değeri sınırlandırılıp bir önceki sonuç değeriyle toplandı. Toplanan değer tekrar sınırlandırılıp  $U_0$  değeriyle çarpıldı ve Runge-Kutta Algoritmasıyla denklemin çözümü yapıldı. Denklemin çözümü olan  $x_1$  değeri y değerine eşitlendi. Hata değeri, referans değerinden y değerini çıkararak hesaplandı ve hata değişimi ise hata değerinden bir önceki hata değerini çıkararak hesaplandı. Bu hata değeri ve hata değişimi değeri sisteme tekrardan verilerek yeniden hesaplamalar yapıldı. Belirli bir süre içerisinde işlemler tekrar etmektedir. Bu şekilde bulanık mantık denetleyicisi tasarlanmış ve simülasyonu tamamlanmıştır. Grafik sonuçlarından hata değerini sıfıra çok yakın bir değere kadar indirilerek salınım yapıldığı gözlemlendi. Kural tablosunun genişletilmesi ile sonuçların iyileştirildiği tespit edildi.

## İçindekiler Tablosu

<b>1. Mamdani Bulanık Karar Verme Algoritmasının Uygulanması.....</b>	<b>6</b>
1.1.Problemin Tanımı .....	6
1.2.Kodun Kısaca Açıklanması ve Mamdani Bulanık Karar Verme Algoritması Çalışması .....	6
1.3.Soruların Cevaplandırılması.....	8
1.4.Elde Edilen Sonuçlar .....	9
<b>2. Sistem Simülasyonu.....</b>	<b>11</b>
2.1.Problemin Tanımı.....	11
2.2.Kodun Açıklanması.....	11
2.3.Elde Edilen Grafiklerin İncelenmesi.....	14
2.4.Elde Edilen Sonuçlar .....	15
<b>3. Bulanık Mantık Denetleyici .....</b>	<b>16</b>
3.1.Problemin Tanımı.....	16
3.2.9 Kurallı Bilgi Tabanı İçeren Kodun Açıklanması.....	16
3.3.Sistemin Çalışmasının Anlatımı .....	20
3.4.25 Kurallı Bilgi Tabanı İçeren Kodun Açıklanması .....	21
3.5.Grafiklerin Yorumlanması .....	22
3.6.Elde Edilen Sonuçlar .....	23
<b>4. Kaynakça.....</b>	<b>24</b>

# 1. Mamdani Bulanık Karar Verme Algoritmasının Uygulanması

## 1.1. Problemin Tanımı

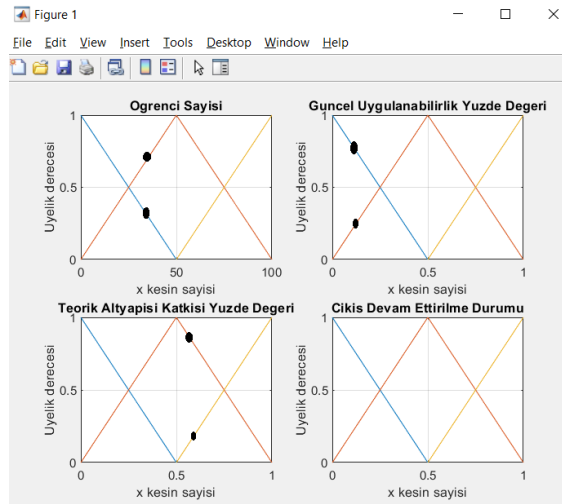
Problem özetle bulanık giriş olarak verilen 3 değer için kesin bir karar vermektir. Bulanık girişlere sahip 3 uzayın gerekli işlemlerden geçirilerek çıkışına karar verilmesi gerekmektedir. Öğrenci sayısı, güncel uygulanabilirlik değeri ve teorik altyapısı değeri ile öğrencinin devam ettirilmesi durumu araştırılmaktadır.

## 1.2. Kodun Kısaca Açıklanması ve Mamdani Bulanık Karar Verme Algoritması Çalışması

Triangle.m scripti içerisindeki fonksiyon bir üçgen üyelik fonksiyonudur. Verilen  $x_1, x_2, x_T$  değerlerine uygun üçgeni çizdirir. Triangle\_value.m scripti içerisindeki fonksiyon ise verilen  $x_1, x_2, x_T$  ve üyelik derecesi istenen  $x$  değerleri için üçgen fonksiyonundaki  $x$  değerinin karşılık geldiği üyelik derecesini döndürür.

Ana script içerisinde gerekli tanımlamalar yapılmıştır. Örneğin DXD değeri düşük öğrenci sayısının alt değerini temsil etmektedir. Bir sonraki adımda giriş üyelik fonksiyonları ve çıkış üyelik fonksiyonu çizdirilmiştir. Kural tablosu matrisi oluşturulmuştur. Bir sonraki adımda ise gerekli üyelik dereceleri alt script olan “Triangle\_value.m” scriptinden elde edilmiştir. Bu üyelik dereceleri ile kural işleme yapılmıştır. 27 tane bulanık kurala sahip olan bu sistemin 27 tane karşılaştırma yapması gerekmektedir. Bu nedenle 9 satır 3 sütun kullanarak bu üyelik dereceleri arasında minimum alma işlemi yapılmaktadır. Diğer bir işlem ise oluşturulan kural matrisi ile minimumları alınmış üyelik derecelerinin matrisinin çarpım işlemidir. En son adımda elde edilen 2.matrisin indis değerlerinin toplamının 1.matrisin indis değerlerinin toplamına bölünmesi ile  $z$  kesin çıkış değeri elde edilmektedir.

Mamdani bulanık karar verme algoritmasına değinecek olunursa aşağıdaki şekilde (Figure 1)  $x=40$ ,  $y=0.2$  ve  $z=0.6$  değerleri için çıkıştaki değer elde edilecektir. Örneğin  $x=40$  değeri için düşük öğrenci sayısı üyelik derecesi yaklaşık 0.3, orta öğrenci sayısı üyelik derecesi yaklaşık 0.7 ve yüksek öğrenci üyelik derecesi 0 olmaktadır. Mamdani bulanık modellemesi yapılırken tüm üçgen üyelik fonksiyonlarındaki üyelik dereceleri karşılaştırılıp bu üyelik derecelerinin minimum alınmaktadır.



MU = [

min(min(MUDX,MUDY),MUDZ) min(min(MUDX,MUDY),MUOZ) min(min(MUDX,MUDY),MUYZ)  
min(min(MUDX,MUOY),MUDZ) min(min(MUDX,MUOY),MUOZ) min(min(MUDX,MUOY),MUYZ)  
min(min(MUDX,MUYZ),MUDZ) min(min(MUDX,MUYZ),MUOZ) min(min(MUDX,MUYZ),MUYZ)  
min(min(MUOX,MUDY),MUDZ) min(min(MUOX,MUDY),MUOZ) min(min(MUOX,MUDY),MUYZ)  
min(min(MUOX,MUOY),MUDZ) min(min(MUOX,MUOY),MUOZ) min(min(MUOX,MUOY),MUYZ)  
min(min(MUOX,MUYZ),MUDZ) min(min(MUOX,MUYZ),MUOZ) min(min(MUOX,MUYZ),MUYZ)  
min(min(MUYX,MUDY),MUDZ) min(min(MUYX,MUDY),MUOZ) min(min(MUYX,MUDY),MUYZ)  
min(min(MUYX,MUOY),MUDZ) min(min(MUYX,MUOY),MUOZ) min(min(MUYX,MUOY),MUYZ)  
min(min(MUYX,MUYZ),MUDZ) min(min(MUYX,MUYZ),MUOZ) min(min(MUYX,MUYZ),MUYZ)  
];

Yukarıdaki kod ile minimum alma işlemleri gerçekleştirilmektedir.

MUK=[MU(1,1)\*K(1,1) MU(1,2)\*K(1,2) MU(1,3)\*K(1,3)

MU(2,1)\*K(2,1) MU(2,2)\*K(2,2) MU(2,3)\*K(2,3)  
MU(3,1)\*K(3,1) MU(3,2)\*K(3,2) MU(3,3)\*K(3,3)  
MU(4,1)\*K(4,1) MU(4,2)\*K(4,2) MU(4,3)\*K(4,3)  
MU(5,1)\*K(5,1) MU(5,2)\*K(5,2) MU(5,3)\*K(5,3)  
MU(6,1)\*K(6,1) MU(6,2)\*K(6,2) MU(6,3)\*K(6,3)  
MU(7,1)\*K(7,1) MU(7,2)\*K(7,2) MU(7,3)\*K(7,3)  
MU(8,1)\*K(8,1) MU(8,2)\*K(8,2) MU(8,3)\*K(8,3)  
MU(9,1)\*K(9,1) MU(9,2)\*K(9,2) MU(9,3)\*K(9,3)];

Yukarıdaki çarpım aşağıdaki formüle dayanmaktadır. K matrisi z(i) değerlerini vermektedir. MU matrisi ise minimumu alınmış üyelik derecelerini vermektedir.

$$\frac{\mu_{MP}(z)z_1 + \mu_{MP}(z)z_2 + \mu_{HP}(z)z_3 + \mu_{HP}(z)z_4}{\mu_{MP}(z) + \mu_{MP}(z) + \mu_{HP}(z) + \mu_{HP}(z)}$$

Mamdani bulanık modellemesinde durulaştırma ile son işlem yukarıdaki formülün aşağıdaki kod yardımı ile hesaplanması ile öğrencinin devam ettirilmesi kesin değeri (f) bulunmaktadır.

```

sizeMU=size(MU);
Toplam1=0;Toplam2=0;
i=1; j=1;
while(i<=sizeMU(1))
    while(j<=sizeMU(2))
        Toplam1=Toplam1+MUK(i,j);
        Toplam2=Toplam2+MU(i,j);
        j=j+1;
    end
    j=1;
    i=i+1;
end
f=Toplam1/Toplam2;
fprintf('w kesin cikis degeri = %f ',f);

```

Ayrıca minimumu alınan üyelik derecelerinin maksimumu alınarak alanların merkezi yöntemi ile bu alanlar birleştirilirse Mamdani bulanık modellemesi elde edilmiş olacaktır. Bu tanıma örnek olarak aşağıdaki şekil verilmiştir.

### 1.3. Soruların Cevaplandırılması

- Bu bulanık karar verme sisteminde en fazla kaç bulanık kural üretilebilir?

Bu bulanık karar verme sisteminde en fazla 27 farklı bulanık kural üretilebilir. Aşağıda yazılan İF-ELSE yapısı ile inceleyebiliriz.

Aşağıdaki tabloda bulanık kurallar yazılırken kullanılan sembollerin temsili değerleri verilmiştir.

Sembol	Temsili Değeri
<b>DX</b>	Düşük Öğrenci Sayısı
<b>OX</b>	Orta Öğrenci Sayısı
<b>YX</b>	Yüksek Öğrenci Sayısı
<b>DY</b>	Düşük Güncel Uygulanabilirlik Yüzde Değeri
<b>OY</b>	Orta Güncel Uygulanabilirlik Yüzde Değeri
<b>YY</b>	Yüksek Uygulanabilirlik Yüzde Değeri
<b>YZ</b>	Düşük Teorik Altyapı Yüzde Değeri
<b>OZ</b>	Orta Teorik Altyapı Yüzde Değeri
<b>YZ</b>	Yüksek Teorik Altyapı Yüzde Değeri
<b>HW</b>	Hayır Devam Ettirme Değeri
<b>OW</b>	Kararsız Devam Ettirme Değeri
<b>YW</b>	Evet Devam Ettirme Değeri



Aşağıda verilen İF-ELSE yapısı ile oluşturulabilecek bulanık kurallar verilmiştir.

*If x is DX and y is DY and z is DZ then w is HW*  
*Else if x is DX and y is DY and z is OZ then w is HW*  
*Else if x is DX and y is DY and z is YZ then w is KW*  
*Else if x is DX and y is OY and z is DZ then w is HW*  
*Else if x is DX and y is OY and z is OZ then w is KW*  
*Else if x is DX and y is OY and z is YZ then w is KW*  
*Else if x is DX and y is YY and z is DZ then w is KW*  
*Else if x is DX and y is YY and z is OZ then w is KW*  
*Else if x is DX and y is YY and z is YZ then w is EW*  
*Else if x is OX and y is DY and z is DZ then w is HW*  
*Else if x is OX and y is DY and z is OZ then w is KW*  
*Else if x is OX and y is DY and z is YZ then w is KW*  
*Else if x is OX and y is OY and z is DZ then w is KW*  
*Else if x is OX and y is OY and z is OZ then w is KW*  
*Else if x is OX and y is OY and z is YZ then w is EW*  
*Else if x is OX and y is YY and z is DZ then w is KW*  
*Else if x is OX and y is YY and z is OZ then w is EW*  
*Else if x is OX and y is YY and z is YZ then w is EW*  
*Else if x is YX and y is DY and z is DZ then w is KW*  
*Else if x is YX and y is DY and z is OZ then w is KW*  
*Else if x is YX and y is DY and z is YZ then w is EW*  
*Else if x is YX and y is OY and z is DZ then w is KW*  
*Else if x is YX and y is OY and z is OZ then w is EW*  
*Else if x is YX and y is OY and z is YZ then w is EW*  
*Else if x is YX and y is YY and z is DZ then w is EW*  
*Else if x is YX and y is YY and z is OZ then w is EW*  
*Else if x is YX and y is YY and z is YZ then w is EW*

- Bu sistem için kendi düşüncenize göre bir kural tablosu oluşturunuz.

KURAL TABLOSU	DZ	OZ	YZ
DX/DY	HW	HW	KW
DX/OY	HW	KW	KW
DX/YY	KW	KW	EW
OX/DY	HW	KW	KW
OX/OY	KW	KW	EW
OX/YY	KW	EW	EW
YX/DY	KW	KW	EW
YX/OY	KW	EW	EW
YX/YY	EW	EW	EW

Yukarıdaki tabloda görüldüğü üzere bir kural tablosu oluşturulmuştur. Kural tablosu oluşturulurken düşük değerler 0, orta değerler 0.5 ve yüksek değerler 1 alınmıştır. Buna göre toplamı 0 ve 0.5 olanlar için hayır devam ettirilmez denilmiştir. Yani düşük öğrenci sayısı yüzde değeri, düşük

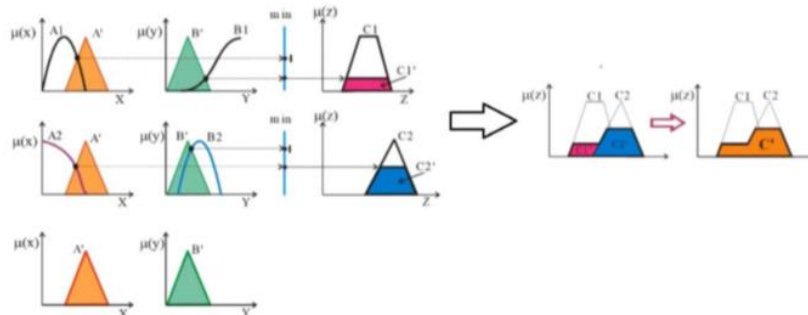
güncel uygulanabilirlik yüzde değeri ve düşük teorik altyapı değeri 0 olduğundan hayır devam ettirilmez denilmiştir. Benzer şekilde düşük öğrenci sayısı yüzde değeri, düşük güncel uygulanabilirlik yüzde değeri ve orta teorik altyapı değeri 0.5 olduğundan hayır devam ettirilmez denilmiştir. Toplam değeri 1 ve 1.5 olan değerler için kararsız devam ettirilme denilmiştir. Buna örnek olarak orta öğrenci sayısı yüzde değeri, orta güncel uygulanabilirlik yüzde değeri ve düşük teorik altyapı değeri 1 olduğundan kararsız devam ettirilme denilmiştir. Benzer şekilde düşük öğrenci sayısı yüzde değeri, orta güncel uygulanabilirlik yüzde değeri ve yüksek teorik altyapı yüzde değeri 1.5 olduğundan kararsız devam ettirilme denilmiştir. Toplam değeri 2,2.5 ve 3 olan değerler için evet devam ettirilebilir denilmiştir. Buna örnek olarak orta öğrenci sayısı yüzde değeri, orta güncel uygulanabilirlik yüzde değeri ve yüksek teorik altyapı değeri 2 olduğundan evet devam ettirilebilir denilmiştir. Benzer şekilde yüksek öğrenci sayısı yüzde değeri, orta güncel uygulanabilirlik yüzde değeri ve yüksek teorik altyapı yüzde değeri 2.5 olduğundan evet devam ettirilebilir denilmiştir. Örnekte verildiği üzere kural tablosu oluşturulmuştur.

- **Oluşturduğunuz kural tablosunu kullanarak,  $x=40$ ,  $y=\%20$  ve  $z=\%60$  olması durumunda  $w$  kesin çıkış değeri ne olur, Mamdani Bulanık Sonuçlandırma algoritmasını kullanarak kesin sonucu bulunuz.**

Program çalıştırıldığında 0.5 sonucunu vermektedir. Programa göre 0.5 sonucunun alınması kararsız devam ettirme sonucunu vermektedir.

- **Sizce bu işlem sizin gibi karar verebiliyor mu?**

Evet karar verebiliyor. Bunun sebebi 3. Maddede programa verilen  $x=40$ ,  $y=0.2$  ve  $z=0.6$  değerleri için programın döndürdüğü sonuç 0.5 olmaktadır. Programa verilen öğrenci sayısının yüzde değeri 0.4, güncel uygulanabilirlik yüzde değeri 0.2 ve teorik altyapı yüzde değeri 0.6 olmaktadır. Bu değerlerin toplamı 1.2'dir. Oluşturulan kural tablosuna göre 1 ve 1.5 değerlerine kararsız devam ettirme durumu denmiştir. Programın döndürdüğü sonuç ile oluşturulan kural tablosu aynı sonuçları vermektedir.



Aşağıdaki tabloda ise program çalıştırıldığında elde edilen bazı sonuçlar gözlenmektedir.

Öğrenci Sayısı (x)	Güncel Uygulanabilirlik (y)	Teorik Altyapı (z)	Devam ettirme (w)
50	0.5	0.5	0.5
60	0.4	0.8	0.772727
70	0.7	0.8	0.941176
20	0.2	0.3	0.235294
10	0.6	0.2	0.409091
50	0.4	0.1	0.428571
50	0.6	0.3	0.571429

#### 1.4. Elde Edilen Sonuçlar

Öncelikle bulanık mantık teoremini ve bulanık mantık problemlerine nasıl yaklaşılacağını öğrendim. Ayrıca Mamdani bulanık algoritması ile bulanık olarak verilen girişlerin bulanık modellemesinin nasıl yapıldığını öğrendim. Verilen problem ile Mamdani bulanık karar verme algoritmasını kullanarak bulanık olarak verilen girişleri nasıl kesin bir çıkış değeri olarak elde edileceğini öğrendim. Sisteme giriş olarak verilen uzman kural tablosu kullanılarak, örnek giriş değerlerine göre çıkış değerini elde etmeyi öğrendim.

## 2. Sistem Simülasyonu

### 2.1. Problemin Tanımı

Verilen problem bir sistemin herhangi bir denetleyici olmadan zamana göre davranışını elde etmektir. Durum denklemlerinin sayısal çözümünün Runge-Kutta algoritması ile bulunduktan sonra bu çözümün zamana göre çizdirilmesi istenmektedir. Çizdirilen bu grafiğin incelenmesi ve sabit bir değere oturup oturmadığının araştırılması istenmektedir.

### 2.2. Kodun Açıklanması

Runge-Kutta Algoritması ile denklem çözümü yapılmak istenmektedir. Runge-Kutta yöntemi, diferansiyel denklemlerin sayısal çözümü için kullanılan yöntemlerden birisidir. Diğer sayısal yöntemlere göre, oldukça iyi sonuçlar veren bir yöntemdir. Problem aşağıdaki şekildeki gibidir. Bu algoritma ile yapılan işlem her x değeri için denklemin çözümünü elde etmektir.

```
A=[-3 -3;1 -9];
B=[1 0 ; 0 1];
C=[1 0];
U0=[100 0];
dt=0.01;
t0=0;
x0=[0 0];
k=1;
sizeA=size(A);
```

Yukarıdaki kod bölümünde gerekli tanımlamalar yapılmaktadır. sizeA değişkeni A matrisinin satır ve sütun boylarını içermektedir.

```
while(t0<tend)
    [x] = myrunge(A,B,U0,x0,dt);
    y(k) = 0;
    for i=1:sizeA(1)
        y(k) = y(k) + C(i) * x(i);
    end
    x0=x;
    t(k)= t0;
    t0 = t0 + dt;
    k = k + 1;
end
plot(t,y);
xlabel('t');
ylabel('y');
title('y-t grafigi');
grid
```

Yukarıdaki kod bölümünde ise belirli bir zaman aralığında myrunge.m scripti alt fonksiyonu çağırılarak denklem çözümü yapılmaktadır. Ayrıca y değeri C matrisi ile x değerlerinin çarpımına eşit olduğundan for döngüsü içerisinde bu işlem gerçekleştirilmektedir. Her iterasyonda x0 dizisinin değerleri yeni x dizisinin değerleri olarak güncellenmektedir. Zaman olarak belirtilen t dizinin değerleri dt kadar artmakta olup t0 ile doldurulmaktadır. Son olarak t0 değeri dt kadar ve k değeri 1 artırılarak işlemlere devam edilmektedir. İterasyon t0 değerinin tend değerine eşit olmasıyla son bulmaktadır. Son işlem olarak y değerleri ile t değerlerinin grafiği çizdirilmektedir. Burada x değerlerinin çözümünün elde edilmesi için myrunge.m scripti içeriği incelenmelidir.

```
function [x] = myrunge(A,B,U,x0,dt)

sizeA=size(A);
i=1; j=1;
while(i<=sizeA(1))
    a(i)=0;
    while(j<=sizeA(2))
        x1(j)=x0(j);
        a(i) = a(i) + A(i,j)*x1(j) + B(i,j)*U(j);
        j = j + 1;
    end
    a(i) = a(i) * dt;
    j = 1;
    i = i + 1;
end
```

Yukarıdaki kod myrunge.m scriptinin ilk kod bölümüdür. Bu script alt program olarak kullanıldığından function deyimi kullanılmaktadır ve bu alt program geriye bir dizi döndürmektedir. sizeA ile A dizisinin satır ve sütun değerleri alınmıştır. 4 adımlı Runge-Kutta algoritmasının ilk adımı yukarıdaki while döngüleri ile temsil edilmektedir. Yapılan işlem A matrisinin i. satırı ile x değerlerinin sütunun çarpılma işlemi ile B matrisinin i. satırı ile U matrisinin sütun değerlerinin çarpımının toplanmasıdır. Bu toplam dt ile çarpıldığında a(i) değerine eşit olacaktır. Kısaca yapılan işlem A matrisi ile x değerlerinin çarpımının B matrisi U matrisinin çarpımının toplanması işlemidir. Bu şekilde x değerlerinin ilk adımdaki çözümü bulunmaktadır.

```

i=1; j=1;
while(i<=sizeA(1))
    b(i)=0;
    while(j<=sizeA(2))
        x2(j)=x0(j)+a(j)/2;
        b(i)= b(i) + A(i,j)*x2(j) + B(i,j)*U(j);
        j = j + 1;
    end
    b(i) = b(i) * dt;
    j = 1;
    i = i + 1;
end

```

Yukarıdaki kod bölümünde Runge-Kutta algoritmasının ikinci adımı gerçekleştirilmektedir. İterasyon ve işlemler ilk adıma benzerdir ancak kullanılan x değerlerinin güncellenmesi gerekmektedir. Kullanılacak x değerleri programa verilen x değeri ile ilk adımda elde edilen a değerinin yarısının toplamı ile elde edilmektedir. Bu şekilde Runge-Kutta algoritmasının ikinci adımı da gerçekleştirilmiştir.

```

i=1; j=1;
while(i<=sizeA(1))
    c(i)=0;
    while(j<=sizeA(2))
        x3(j)=x0(j)+b(j)/2;
        c(i) = c(i) + A(i,j)*x3(j) + B(i,j)*U(j);
        j = j + 1;
    end
    c(i) = c(i) * dt;
    j = 1;
    i = i + 1;
end

```

Yukarıdaki kod bölümünde Runge-Kutta algoritmasının üçüncü adımı gerçekleştirilmektedir. İterasyon ve işlemler ikinci adıma benzerdir ve aynı şekilde kullanılan x değerlerinin güncellenmesi gerekmektedir. Kullanılacak x değerleri programa verilen x değeri ile ikinci adımda elde edilen b değerinin yarısının toplamı ile elde edilmektedir. Bu şekilde Runge-Kutta algoritmasının üçüncü adımı da gerçekleştirilmiştir.

```

i=1; j=1;
while(i<=sizeA(1))
    d(i) = 0;
    while(j<=sizeA(2))
        x4(j)=x0(j)+c(j);
        d(i) = d(i) + A(i,j)*x4(j) + B(i,j)*U(j);
        j = j + 1;
    end
    d(i) = d(i) * dt;
    j = 1;
    i = i + 1;
end

```

Yukarıdaki kod bölümünde Runge-Kutta algoritmasının dördüncü adımı gerçekleştirilmektedir. İterasyon ve işlemler üçüncü adıma benzerdir ve aynı şekilde kullanılan x değerlerinin güncellenmesi gerekmektedir. Kullanılacak x değerleri programa verilen x değeri ile üçüncü adımda elde edilen c değerinin toplamı ile elde edilmektedir. Bu şekilde Runge-Kutta algoritmasının dördüncü adımı da gerçekleştirilmiştir.

```

for i=1:sizeA(1)
    x(i) = x0(i) + (a(i) + 2 * b(i) + 2 * c(i) + d(i)) / 6;
end
end

```

Yukarıdaki kod bölümünde ise hesaplanan  $x_0$ ,  $a$ ,  $b * 2$ ,  $c * 2$  ve  $d$  değerlerinin toplamının 6ya bölünmesi ile verilen denklemin çözüm işlemi gerçekleştirilmiştir.

#### 4 Adımlı R-K algoritması

$$y(n) = y(n-1) + (1/6)*(k_1 + 2*k_2 + 2*k_3 + k_4)$$

$$k_1 = dy/dt (y(n-1)) * dt/2$$

$$k_2 = dy/dt (y(n-1) + k_1) * dt/2$$

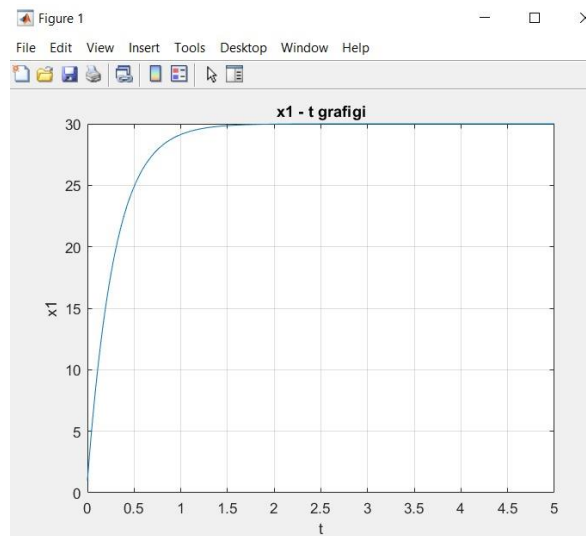
$$k_3 = dy/dt (y(n-1) + k_2) * dt$$

$$k_4 = dy/dt (y(n-1) + k_4)$$

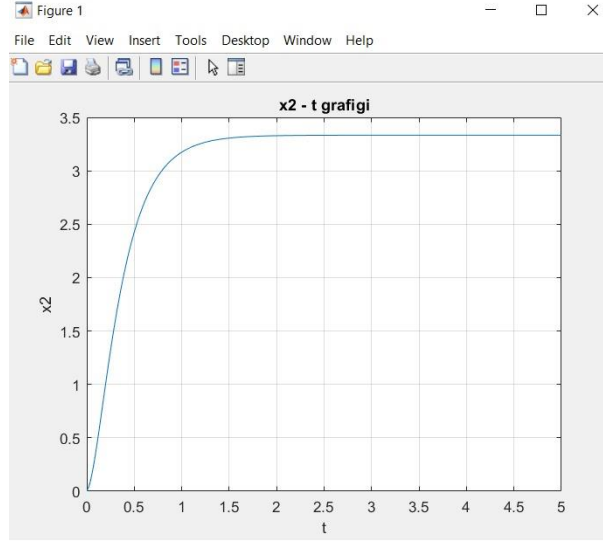
4 adımlı Runge-Kutta Algoritmasının kodlanması yukarıdaki şekilden yararlanılarak yapılmıştır.

### 2.3. Elde Edilen Grafiklerin İncelenmesi

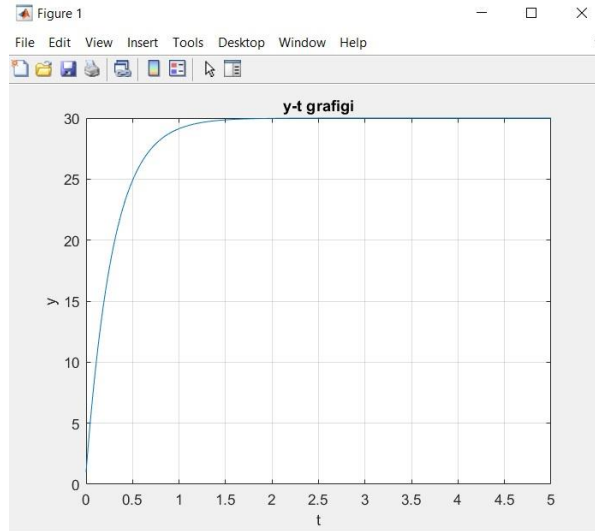
Aşağıdaki şekil  $x_1-t$  grafiğini temsil etmektedir. Grafik incelenirse 4 adımlı Runge-Kutta Algoritması ile denklemler çözüldükten sonra  $x_1$  değerleri zamana bağlı görüntülenmektedir. Burada grafik sabit 30 değerine oturmuştur.



Aşağıdaki şekil ise  $x_2$ - $t$  grafiğini temsil etmektedir. Grafik incelenirse 4 adımlı Runge-Kutta Algoritması ile denklemler çözüldükten sonra  $x_2$  değerleri zamana bağlı görüntülenmektedir. Burada grafik sabit  $x_2$  değeri yaklaşık 3.3 değerine oturmuştur.



Aşağıdaki şekil  $y$ - $t$  grafiğini temsil etmektedir. Program incelenirse  $y$  değeri  $x_1$  ve  $x$  değerinin  $C$  matrisi ile çarpımına eşittir.  $C$  matrisi  $[1 \ 0]$  olduğundan  $y$  değerleri  $x_1$  değerlerine eşit olacaktır. Bu şekilde  $y$ - $t$  grafiği ile  $x_1$ - $t$  grafiği aynı grafiği temsil etmektedirler.



## 2.4. Elde Edilen Sonuçlar

Bir sistemin herhangi bir denetleyici olmadan zamana göre davranışının nasıl elde edileceğini öğrendim. Durum denklemlerinin sayısal çözümünün Runge-Kutta algoritması ile bulunduktan sonra bu çözümün zamana göre çizdirilmesini öğrendim. Çizdirilen bu grafiğin incelenmesi ve sabit bir değere oturup oturmadığını gözlemledim.

### 3. Bulanık Mantık Denetleyici

#### 3.1. Problemin Tanımı

2. kısımdaki sistemin y çıkış değerini kontrol edecek bulanık mantık denetleyicisinin tasarlanması ve simülasyonunun yapılması. Problem sistem tarafından Runge-Kutta Algoritması ile hesaplanan y değerini , Mamdani Bulanık Sonuçlandırma Algoritmasını kullanarak hata hesaplamasını yapıp denetleyici ile kontrolü sağlanarak referans değerinde sabitlemektir. Hesaplanan y değerinin referans değerine ulaşması ile hata değeri 0'a indirilmektedir.

#### 3.2. 9 Kurallı Bilgi Tabanı İçeren Kodun Açıklanması

2. kısımda kullanılan sistemin çıkışı bir bulanık mantık denetleyici ile tasarlanıp, simülasyonu yapılacaktır. Giriş olarak veriler aşağıdaki kod bölümündeki şekilde tanımlanmıştır. Referans girişi 25 olarak belirlenmiştir.

```
A=[-3 -3 ; 1 -9];  
B=[1 0 ; 0 1];  
u=[100 ; 0];  
dt=0.01;  
t0=0;  
tend=2;  
k=1;  
U0=[u(1) ; u(2)];  
sizeA=size(A);  
r0=25;
```

Aşağıdaki kod bölümünde sisteme giriş olarak verilecek x değerleri 0 olarak tanımlanmışlardır. Maksimum E değeri, referans değeri olarak tanımlanmıştır. Minimum E değeri ise referans değerinin negatifi olarak tanımlanmıştır. DE maksimum ve minimum değerleri, E maksimum ve minimum değerlerinin 10 da 1 i olarak tanımlanmışlardır. Son olarak DU maksimum değeri 1 ve DU minimum değeri -1 olarak alınmıştır.

```
for i=1:sizeA(1)  
    x0(i)=0;  
end  
  
EMAX = r0;  
EMIN = -EMAX;  
DEMAX = EMAX/10;  
DEMIN = -DEMAX;  
DUMAX = 1;  
DUMIN = -1;
```



Aşağıdaki kod bölümünde e, de ve du değerleri için düşük, orta ve yüksek sınırları tanımlanmıştır. Bu tanımlamalar ile uygun üçgen üyelik fonksiyonları çizdirilebilir.

```
NeD = EMIN;      NeO = EMIN;      NeY = 0;
ZeD = EMIN;      ZeO = 0;          ZeY = EMAX;
PeD = 0;          PeO = EMAX;      PeY = EMAX;

NdeD = DEMIN;    NdeO = DEMIN;    NdeY = 0;
ZdeD = DEMIN;    ZdeO = 0;          ZdeY = DEMAX;
PdeD = 0;          PdeO = DEMAX;    PdeY = DEMAX;

NduD = DUMIN;    NduO = DUMIN;    NduY = 0;
ZduD = DUMIN;    ZduO = 0;          ZduY = DUMAX;
PduD = 0;          PduO = DUMAX;    PduY = DUMAX;
```

Aşağıdaki kod bölümünde üçgen üyelik fonksiyonundan üyelik derecesi almak için gönderilen E ve DE değerleri tanımlanmıştır. Kullanılan ee değeri maksimum hata değerinin üstündeki bir değeri EMAX'a ve minimum hata değerinin altındaki bir değeri EMIN'e eşitlemek için kullanılmaktadır. Benzer şekilde dee değişkeni de maksimum hata değeri değişiminin üstündeki değerleri DEMAX'a ve minimum hata değeri değişiminin altındaki değerleri DEMIN'e eşitlemek için kullanılmaktadır. Z dizisi, Mamdani Bulanık Sonuçlandırma Algoritması ile hesaplanan kesin çıkışları içermektedir ve akış şemasında U(k) dizisini temsil etmektedir. İlk elemanı 0 'a eşitlenmiştir. Z0 ise akış şemasında belirtilen U(k-1) değerini temsil etmektedir.

```
ee = EMAX;
dee = 0;
E = EMAX;
DE = 0;
e0 = EMAX;
Z(1) = 0;
Z0 = 0;
```

Aşağıdaki kod bölümünde ödev dosyasında 3.kısımda 1.çizelgedeki bilgi tabanı, kural tablosu olarak tanımlanmıştır. Ayrıca SizeDU değişkenine DU matrisinin satır ve sütun boyutları atanmıştır.

```
DU=[ZduO   PduO   PduO
     NduO   ZduO   PduO
     NduO   NduO   ZduO
     ];

SizeDU=size(DU);
```

Çizelge 1. Simetrik 9 kurallı bilgi tabanı.

e(k)	de(k)		
	N <sub>de</sub>	Z <sub>de</sub>	P <sub>de</sub>
P <sub>e</sub>	Z <sub>du</sub>	P <sub>du</sub>	P <sub>du</sub>
Z <sub>e</sub>	N <sub>du</sub>	Z <sub>du</sub>	P <sub>du</sub>
N <sub>e</sub>	N <sub>du</sub>	N <sub>du</sub>	Z <sub>du</sub>

Aşağıdaki kod bölümünde while döngüsü t0 değeri dt kadar artarak tend değerine ulaşana kadar devam edecektir. İlk olarak E değeri EMAX ve EMIN arasında sınırlandırılmıştır. Aşağıda limiter.m scripti içerisindeki kod verilecektir. Değer sınırlandırıldıktan sonra e ve de değerlerinin üyelik dereceleri üçgen üyelik fonksiyonu ile elde edilmiştir. Alt script olan Triangle\_value alt programı ilk kısımda da kullanılmış olup verilen değerlere göre üyelik derecesini döndürmektedir. Mamdani üyelik derecesi adında e değerinin negatifi, sıfırı ve pozitifini için üyelik dereceleri elde edilmiştir. Bu üyelik dereceleri Mamdani Bulanık Sonuçlandırma Algoritması ile kesin değer bulmada kullanılacaklardır. Benzer şekilde DE değeri DEMAX ve DEMIN değerleri arasında sınırlandırılmıştır. Uygun üyelik dereceleri Triangle\_value alt programı ile elde edilmiştir. Mamdani Bulanık Sonuçlandırma Algoritmasının çalışması 1.kısımda anlatılmıştır buna göre uygun üyelik derecelerinin minimumları ile MU ile temsil edilen matriste minimum üyelik dereceleri tutulmaktadır. MUDU matrisi, kural matrisi ve minimum üyelik derecesi matrisinin çarpımı ile elde edilmektedir. MUDU matrisinin elemanlarının toplamı, MU matrisinin elemanlarının toplamına bölünerek durulaştırma işlemi gerçekleştirilmektedir. Durulaştırma ile bulanık olarak verilen e değerleri DU kural tablosu kullanılarak kesin bir sonuç olarak elde edilir.

```
while(t0<tend)

E=limiter(EMIN,EMAX,ee);
MUeN = Triangle_value(NeD, NeO, NeY, E);
MUeZ = Triangle_value(ZeD, ZeO, ZeY, E);
MUeP = Triangle_value(PeD, PeO, PeY, E);

DE=limiter(DEMIN,DEMAX,dee);
MUDeN = Triangle_value(NdeD, NdeO, NdeY, DE);
MUDeZ = Triangle_value(ZdeD, ZdeO, ZdeY, DE);
MUDeP = Triangle_value(PdeD, PdeO, PdeY, DE);

MU=[
    min(MUeP,MUDeN)    min(MUeP,MUDeZ)    min(MUeP,MUDeP)
    min(MUeZ,MUDeN)    min(MUeZ,MUDeZ)    min(MUeZ,MUDeP)
    min(MUeN,MUDeN)    min(MUeN,MUDeZ)    min(MUeN,MUDeP)
];

MUDU=[MU(1,1)*DU(1,1)    MU(1,2)*DU(1,2)    MU(1,3)*DU(1,3)
       MU(2,1)*DU(2,1)    MU(2,2)*DU(2,2)    MU(2,3)*DU(2,3)
       MU(3,1)*DU(3,1)    MU(3,2)*DU(3,2)    MU(3,3)*DU(3,3)
];
```

Aşağıdaki kod bölümünde z kesin sonucu durulaştırma yöntemi ile yukarıda açıklandığı üzere elde edilmiştir. Z değeri limiter.m alt programına gönderilerek DMIN ile DMAX arasındaki bir değerle sınırlandırılmıştır. Z dizisinin k. elemanına bir önceki kesin değer ve elde edilen kesin değer toplamı atanmıştır. Akış diyagramı ile ilişkilendirmek gerekirse Z(k) değeri U1(k) değerini, Z değeri U(k-1) değerini, Z değeri DU(k) değerini ve z değeri du(k) değerini temsil etmektedir. Z(k) değeri akış diyagramında belirtildiği gibi 0 ile 1 arasında sınırlandırılmıştır. Z(k) değeri bir sonraki adımda kullanılmak üzere Z0 değişkenine atanmıştır. Ayrıca akış diyagramındaki u(k) değerini temsil eden U00 değişkeni u dizinin maksimum elemanı olan U0 ve Z(k) nın çarpımı ile elde edilir. Elde edilen U00, A matrisi, B matrisi, başlangıç x değerleri olan x0 vektörü ve dt değeri 2. kısımda kullanılan myrunge.m

alt programına gönderilmiştir. 2. kısımda da anlatıldığı üzere myrunge.m alt programı ile 4 adımlı Runge-Kutta Algoritması kullanılarak x değerleri hesaplanmaktadır.

UU(k) dizisi sisteme verilen UU0 değerlerinin dizisini tutmaktadır. Zaman eksenini için ihtiyaç duyulan t(k) değeri her iterasyonda t0 değerine dt eklenilmesiyle elde edilen bir dizidir. Referans dizisi olan r(k), her adımda aynı değer olan r0 ile doldurulmaktadır ve grafikte sabit düz bir çizgi olarak referans değerini göstermek amacıyla kullanılmaktadır. Çıkış değeri olan y değeri x1 değerine eşit olduğundan y(k) değerine x0(1) değeri atanmıştır. Akış diyagramında belirttiği üzere hata değeri olan e(k) değeri, referans değeri olan r(k) değerinden çıkış değeri olan y(k) değeri çıkarılarak elde edilmektedir. Hata değişimi olan de(k) değeri, hata değeri olan e(k) değerinden bir önceki hata değeri olan e(k-1) değerini çıkararak elde edilir. Sisteme verilecek olan ee değeri yeni hata değeri olan e(k) ile güncellenir ve dee değeri yeni hata değişimi değeri olan de(k) ile güncellenir. X değerleri for döngüsü içerisinde güncellenirler. K değeri her iterasyonda bir artırılarak dizilerin indislerini belirlemektedir. Bu şekilde güncellenen değerler sisteme tekrar verilerek denetim gerçekleştirilir. Sistemin kontrolü belirli bir zaman aralığında incelenmektedir. Bu sistem hata değeri sıfır olana dek sürdürülebilir.

```
sizeMU=size(MU);
z=sum(MUDU)/sum(MU);
Z=limiter(DUMIN, DUMAX, z);
Z(k) = Z0 +Z;
Z(k)=limiter(0, DUMAX, Z(k));
Z0 = Z(k);
U00 = Z(k) * U0;

[x]=myrunge(A,B,U00,x0,dt);
UU(k)=U00(1);
t(k)=t0+dt;
t0=t(k); r(k)=r0;
y(k)=x0(1); e(k)=r(k)-y(k);
de(k)=e(k)-e0; ee=e(k); dee=de(k); e0=e(k);
for n=1:sizeA(1)
    x0(n)=x(n);
    XX(k,n)=x(n);
end
k=k+1;
end
```

Aşağıdaki kod bölümünde grafiklerin çizdirilme işlemleri gerçekleştirilmektedir.

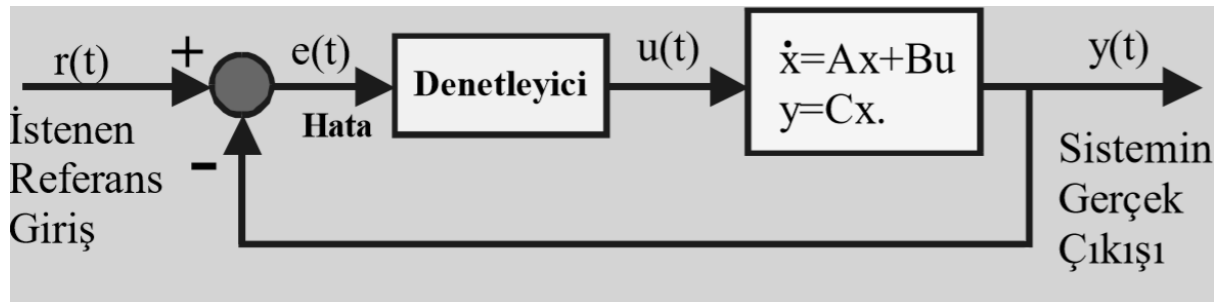
```
subplot(211);
plot(t,y,t,r);
xlabel('Zaman(t)'); ylabel('Cikis(y) - Referans(r)');
title(' (Cikis - Referans) - Zaman Grafigi');
grid
subplot(212)
plot(t,e)
xlabel('Zaman(t)'); ylabel('Hata(e)');
title(' Hata - Zaman Grafigi');
grid
```

Aşağıdaki kod bölümünde limiter.m alt programı görüntülenmektedir. Bu alt program alınan ee değerini EMIN ile EMAX değeri arasında sınırlandırmak için yazılmıştır. Eğer ee değeri EMAX değerinin üzerinde veya eşit bir değer ise geri döndürülecek E değerine EMAX değeri atanmıştır. Benzer şekilde ee değeri EMIN değerinin altında veya eşit bir değer ise geri döndürülecek E değerine EMIN değeri atanmıştır. Mümkün olan durum bu iki durumdan biri olmadığı takdirde E değerine ee değeri atanmıştır.

```
function E=limiter(EMIN,EMAX,ee)
    if ee>=EMAX
        E=EMAX;
    elseif ee<=EMIN
        E=EMIN;
    else
        E=ee;
    end
end
```

### 3.3. Sistemin Çalışmasının Anlatımı

3. Kısımda yapılan çalışmalar Bulanık Mantık Denetleyici ile verilen sistemin çıkış değerini referans değerinde sabitlemek ve hata değerini sıfıra indirmektir. Bu işlem için öncelikle oluşturulan kural tablosu ile bulanık olarak girilen değerleri Mamdani Bulanık Sonuçlandırma Algoritmasıyla durulaştırıp kesin bir değer elde etmek gerekir. Sonrasında elde edilen kesin değer sınırlandırılarak, bir önceki kesin değerle toplanır. Sonrasında bu toplanan değerler 0 ile 1 arasında sınırlandırılıp maksimum u değeriyle çarpılır. Elde edilen bu değer ve sisteme giriş olarak verilen değerler Runge-Kutta Algoritması ile hesaplanır. Hesaplama sonucunda çıkış değeri elde edilir ve referans değerinden çıkış değeri çıkarılarak hata değeri elde edilir. Hata değerinden bir önceki hata değeri çıkarılarak hata değişimi elde edilir. Hata değeri ile hata değişimi değeri sisteme verilir ve yazılan bu aşamalar tekrar uygulanır. Bu şekilde hataya göre denetim yapan bir sistem elde edilir. Bu sistem çıkış değerini referans değerine çıkartır ve hata değerini sıfıra indirir. Yapılan çalışma aşağıdaki sistemin tasarımı ve simülasyonudur.



### 3.4. 25 Kurallı Bilgi Tabanı İçeren Kodun Açıklanması

25 kurallı bilgi tabanı içeren kod ile 9 kurallı bilgi tabanı içeren kod tanımlamaları haricinde algoritma bakımından aynıdır. 25 kurallı bilgi tabanının kullanılması salınım miktarının daha az olması ve daha az zamanda çıkış değerinin referans değerine eşitlenmesidir. Aşağıdaki kod bölümlerinde 9 kurallı bilgi tabanı içeren koddan farklı olan kod bölümleri görüntülenecektir. Farklı olan kod bölümlerinden birisi üçgen üyelik fonksiyonunun başlangıç, tepe ve bitiş noktalarının tanımlanması bölümüdür. Aşağıda bu kod bölümü görüntülenmektedir. Üçgen üyelik fonksiyonu tanım aralığı hata değeri olan  $e$  için EMIN ile EMAX arasında olmaktadır. Bu uzay 5'e bölündüğünde sınır değerleri aşağıdaki şekilde olmaktadır. Bu bölge arasına 5 ayrı üçgen çizdirilmiştir.

```

NBeD = EMIN;      NBeO = EMIN;      NBeY = EMIN/2;
NSeD = EMIN;      NSeO = EMIN/2;      NSeY = 0;
ZEeD = EMIN/2;     ZEeO = 0;          ZEeY = EMAX/2;
PSeD = 0;          PSeO = EMAX/2;     PSeY = EMAX;
PBeD = EMAX/2;     PBeO = EMAX;       PBeY = EMAX;

NBdeD = DEMIN;     NBdeO = DEMIN;     NBdeY = DEMIN/2;
NSdeD = DEMIN;     NSdeO = DEMIN/2;   NSdeY = 0;
ZEdeD = DEMIN/2;   ZEdeO = 0;         ZEdeY = DEMAX/2;
PSdeD = 0;         PSdeO = DEMAX/2;   PSdeY = DEMAX;
PBdeD = DEMAX/2;   PBdeO = DEMAX;     PBdeY = DEMAX;

NBduD = DUMIN;     NBduO = DUMIN;     NBduY = DUMIN/2;
NSduD = DUMIN;     NSduO = DUMIN/2;   NSduY = 0;
ZEduD = DUMIN/2;   ZEduO = 0;         ZEduY = DUMAX/2;
PSduD = 0;         PSduO = DUMAX/2;   PSduY = DUMAX;
PBduD = DUMAX/2;   PBduO = DUMAX;     PBduY = DUMAX;

```

Aşağıdaki kod bölümünde ise kural tablosunun oluşturulması görüntülenmektedir. Bu kural tablosu Çizelge 2'den yararlanılarak oluşturulmuştur.

Çizelge 2. Simetrik 25 kurallı bilgi tabanı

e(k)	de(k)				
	NB <sub>de</sub>	NS <sub>de</sub>	ZE <sub>de</sub>	PS <sub>de</sub>	PB <sub>de</sub>
PB <sub>e</sub>	ZE <sub>du</sub>	PS <sub>du</sub>	PS <sub>du</sub>	PB <sub>du</sub>	PB <sub>du</sub>
PS <sub>e</sub>	NS <sub>du</sub>	ZE <sub>du</sub>	PS <sub>du</sub>	PS <sub>du</sub>	PB <sub>du</sub>
ZE <sub>e</sub>	NS <sub>du</sub>	NS <sub>du</sub>	ZE <sub>du</sub>	PS <sub>du</sub>	PS <sub>du</sub>
NS <sub>e</sub>	NB <sub>du</sub>	NS <sub>du</sub>	NS <sub>du</sub>	ZE <sub>du</sub>	PS <sub>du</sub>
NB <sub>e</sub>	NB <sub>du</sub>	NB <sub>du</sub>	NS <sub>du</sub>	NS <sub>du</sub>	ZE <sub>du</sub>

```

DU=[ ZEduO  PSduO  PSduO  PBduO  PBduO
    NSduO  ZEduO  PSduO  PSduO  PBduO
    NSduO  NSduO  ZEduO  PSduO  PSduO
    NBduO  NSduO  NSduO  ZEduO  PSduO
    NBduO  NBduO  NSduO  NSduO  ZEduO
];

```

Aşağıdaki kod bölümünde üçgen üyelik fonksiyonunda üyelik derecelerinin alınması görüntülenmektedir. Alınan bu üyelik derecelerinin minimumları alınarak MU matrisi oluşturulmuştur. MU matrisiyle DU kural matrisinin çarpımıyla MUDU matrisi oluşturulmuştur. MUDU matrisinin elemanlarının toplamı , MU matrisinin elemanlarının toplamına bölünerek z kesin çıkış değerini oluşturmuştur. Bu işlemin devamı 9 kurallı bilgi tabanı içeren kodun açıklanmasında detaylı bir şekilde anlatılmıştır.

```
E=limiter(EMIN,EMAX,ee);
MUeNB = Triangle_value(NBeD, NBeO, NBeY, E);
MUeNS = Triangle_value(NSeD, NSeO, NSeY, E);
MUeZE = Triangle_value(ZeD, ZeO, ZeY, E);
MUePS = Triangle_value(PSeD, PSeO, PSeY, E);
MUePB = Triangle_value(PBeD, PBeO, PBeY, E);

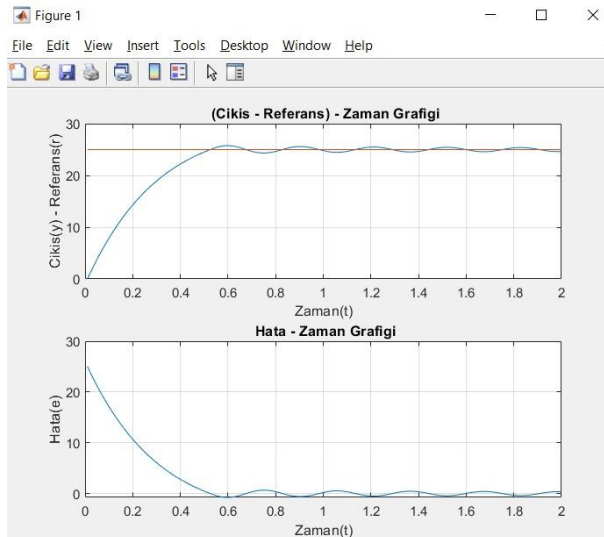
DE=limiter(DEMIN,DEMAX,dee);
MUDeNB = Triangle_value(NBdeD, NBdeO, NBdeY, DE);
MUDeNS = Triangle_value(NSdeD, NSdeO, NSdeY, DE);
MUDeZE = Triangle_value(ZEdeD, ZEdeO, ZEdeY, DE);
MUDePS = Triangle_value(PSdeD, PSdeO, PSdeY, DE);
MUDePB = Triangle_value(PBdeD, PBdeO, PBdeY, DE);

MU=[
    min(MUePB,MUDeNB)    min(MUePB,MUDeNS)    min(MUePB,MUDeZE)    min(MUePB,MUDePS)    min(MUePB,MUDePB)
    min(MUePS,MUDeNB)    min(MUePS,MUDeNS)    min(MUePS,MUDeZE)    min(MUePS,MUDePS)    min(MUePS,MUDePB)
    min(MUeZE,MUDeNB)    min(MUeZE,MUDeNS)    min(MUeZE,MUDeZE)    min(MUeZE,MUDePS)    min(MUeZE,MUDePB)
    min(MUeNS,MUDeNB)    min(MUeNS,MUDeNS)    min(MUeNS,MUDeZE)    min(MUeNS,MUDePS)    min(MUeNS,MUDePB)
    min(MUeNB,MUDeNB)    min(MUeNB,MUDeNS)    min(MUeNB,MUDeZE)    min(MUeNB,MUDePS)    min(MUeNB,MUDePB)
];

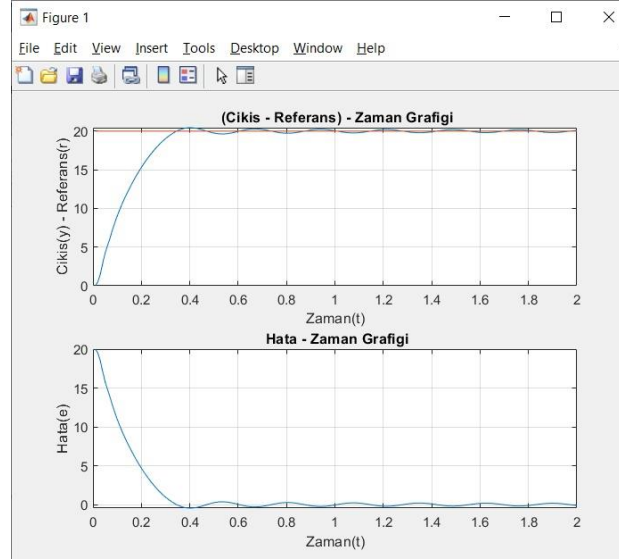
MUDU=[MU(1,1)*DU(1,1)    MU(1,2)*DU(1,2)    MU(1,3)*DU(1,3)    MU(1,4)*DU(1,4)    MU(1,5)*DU(1,5)
        MU(2,1)*DU(2,1)    MU(2,2)*DU(2,2)    MU(2,3)*DU(2,3)    MU(2,4)*DU(2,4)    MU(2,5)*DU(2,5)
        MU(3,1)*DU(3,1)    MU(3,2)*DU(3,2)    MU(3,3)*DU(3,3)    MU(3,4)*DU(3,4)    MU(3,5)*DU(3,5)
        MU(4,1)*DU(4,1)    MU(4,2)*DU(4,2)    MU(4,3)*DU(4,3)    MU(4,4)*DU(4,4)    MU(4,5)*DU(4,5)
        MU(5,1)*DU(5,1)    MU(5,2)*DU(5,2)    MU(5,3)*DU(5,3)    MU(5,4)*DU(5,4)    MU(5,5)*DU(5,5)
];
```

### 3.5. Grafiklerin Yorumlanması

Aşağıdaki grafik 9 kurallı bilgi tablosu kullanılarak elde edilmiştir. Çıktıdaki y değeri referans değerini takip etmektedir. Çıkış değeri referans değeri üzerinde ufak salınımlar yapmakta ancak değerini referans değeriyle eşitleyememiştir. Zamanın daha fazla arttırılması veya hata kontrolü ile hata sıfıra indirilebilir ve çıkış değeri referans değerine eşitlenebilir.



Aşağıdaki grafik 25 kurallı bilgi tablosu kullanılarak elde edilmiştir. Çıkıştaki  $y$  değeri referans değerini takip etmektedir. Çıkış değeri referans değeri üzerinde çok daha ufak salınımlar yapmaktadır ancak değerini referans değeriyle eşitleyememiştir. 9 kurallı bilgi tablosu kullanılarak elde edilen sonuca göre çok daha iyidir. Benzer şekilde zamanın daha fazla artırılması veya döngüde hata kontrolü ile hata sıfıra indirilebilir ve çıkış değeri referans değerine eşitlenebilir. Kural tabanı arttıkça sistemin daha iyi sonuç verdiği gözlemlenmiştir. Kural tabanı da artırılarak daha iyi sonuçlar alınabilir.



### 3.6. Elde Edilen Sonuçlar

Çıkış değerine göre hata hesaplaması ile bulanık mantık denetleyicisinin tasarlanması ve simülasyonunun yapılmasını öğrendim. Problem sistem tarafından Runge-Kutta Algoritması ile hesaplanan  $x_1$  değerini, Mamdani Bulanık Sonuçlandırma Algoritmasını kullanarak hata hesaplamasını yapıp denetleyici ile kontrolü sağlanarak referans değerinde sabitlemeyi öğrendim. Bulanık mantık denetleyicisinin tasarımında kullanılması gereken yöntem ve algoritmaları öğrendim. Denetimli sistemlerin nasıl çalıştığını ve altyapısında bulundurduğu faktörleri öğrendim.

#### 4. KAYNAKÇA

<http://courses.altas.org/BM/bm.html>