



# **BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

## **VERİYE DAYALI BULANIK SİSTEM MODELLEME VE TANIMLAMA**

**Abdoulaye Abdramane MAKHAİLA**

**Yüksek Lisans**

**Tez Danışmanı**

**Doç. Dr. Cihan KARAKUZU**

**BİLECİK, 2017**

**Ref.No.: 10160183**



# **BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**

**Fen Bilimleri Enstitüsü**

**Bilgisayar Mühendisliği Bölümü**

## **VERİYE DAYALI BULANIK SİSTEM MODELLEME VE TANIMLAMA**

**Abdoulaye Abdramane MAKHAİLA**

**Yüksek Lisans**

**Tez Danışmanı**

**Doç. Dr. Cihan KARAKUZU**

**BİLECİK, 2017**



**BİLECİK SEYH EDEBALI UNIVERSITY**

**Graduate School of Sciences**

**Department of Computer Engineering**

**DATA-BASED FUZZY SYSTEM MODELING AND  
IDENTIFICATION**

**Abdoulaye Abdramane MKHAÏLA**

**Master's Thesis**

**Thesis Advisor**

**Assoc. Prof. Dr. Cihan KARAKUZU**

**BİLECİK, 2017**



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
YÜKSEK LİSANS JÜRİ  
ONAY FORMU

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun 12/07/2017 tarih ve 35 sayılı kararıyla oluşturulan jüri tarafından 28/07/2017 tarihinde tez savunma sınavı yapılan Abdoulaye Abdramane MAKHAILA 'ın "Veriye Dayalı Bulanık Sistem Modelleme ve Tanımlama" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak oy birliği ile kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Cihan KARAKUZU

ÜYE : Yrd. Doç. Dr. Sıtkı ÖZTÜRK

ÜYE : Doç. Dr. Uğur YÜZGEÇ

ONAY

Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulunun ..... tarih ve ..... sayılı kararı.

## TEŐEKKÖRLER

Bu tez alıőmamda bana bilgi ve tecrübeleri ile yardımcı olan yol gösteren, rehberlik eden ve adeta bana mükemmel bir kaynak olan deęerli hocam Do. Dr. Cihan KARAKUZU'ya teőekkürlerimi bir bor bilirim. alıőmam boyunca desteklerini benden esirgemeyen aileme sonsuz minnettarım. Ayrıca alıőmam boyunca bana yardımcı olan ve katlanan arkadaşlarıma ok teőekkür ederim.

## ÖZET

Bulanık mantık insan düşünüş tarzından esinlenerek bulunmuş bulanık küme teorisine dayanan genel bir hesaplama sistemidir. Bu sistem mantıksal ifadeler ve dilsel değişkenler arasındaki bağlantılara dayanmaktadır. Mühendislik açısından en önemli üstünlüklerinden biri ilgilenilen sistemin matematiksel modeline ihtiyaç duymamasıdır. Temel sorun kendisinden beklenen görevi yapacak şekilde parametrelerinin en uygun değerlerinin belirlenmesidir. Bu çalışmada, yaygın olarak kullanılan Takagi-Sugeno (TS) tipi bulanık sistemin parametrelerinin eldeki giriş-çıkış verilerine dayalı olarak belirlenmesi üzerine çalışılmıştır. Bu çalışmada verilen yöntem ile önce, bulanık sistemin giriş-çıkış değişkenleri belirlenerek girişe ait değişkenler ilgili uzayda yaklaşık olarak homojen dağıtılmış üyelik fonksiyonları ile bulandırılmaktadır. Böylece bulanık sistemin şart/öncül parametreleri belirlenir. Daha sonra, en küçük karesel kestirim (LSE, Least Square Estimation) yöntemi ile eldeki giriş-çıkış örnek verilerine dayalı olarak bulanık sistemin sonuç/kural parametreleri belirlenir. Bu çalışmada, anılan bu yöntem beş farklı denektaş dinamik sistemin TS tipi birer bulanık sistem ile modellenmesi üzerinde çalışılmıştır. Elde edilen sonuçlar tasarımcının elinde giriş-çıkış örneklerinin bulunması durumunda yöntemin etkin bir şekilde kullanılabileceğini göstermektedir.

**Anahtar Kelimeler-** Bulanık mantık, Sistem modelleme, Dinamik sistem, TS bulanık sistem, En küçük karesel hata

## ABSTRACT

Fuzzy logic is a general computation system based on the fuzzy set theory, which is inspired by human thinking. This system is based on the relations between logical expressions and linguistic variables. One of the most important advantages in terms of engineering is that it does not need a mathematical model of the system of interest. The main problem is to determine the most suitable values of its parameters so as to perform the task expected from it. In this study, it has investigated that the parameters of commonly used TS type fuzzy system are determined based on input / output data at hand. First, the input-output variables of the fuzzy system are determined by the method given in this study, and the variables of the input are fuzzificated by the homogeneous distributed membership functions in the related input space. Thus, the premise/antecedent parameters of the fuzzy system are determined. Then, the consequent/rule parameters of the fuzzy system are determined based on the input-output sample data with the least square estimation (LSE) method. In this study, this method is discussed on the modeling of five different dynamical systems with fuzzy systems of TS type. The results show that the method can be used effectively if the designer has input-output samples.

**Keywords-** Fuzzy logic, System modeling, Dynamic system, TS fuzzy system, Least square estimation

## İÇİNDEKİLER

### JÜRİ ONAY SAYFASI

### TEŞEKKÜR

ÖZET.....	i
ABSTRACT .....	ii
SIMGELER VE KISATMALAR DİZİNİ.....	V
ŞEKİLLER DİZİNİ .....	VI
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. BULANIK MANTIK VE BULANIK MODELLER.....</b>	<b>4</b>
<b>2.1. Bulanık Mantık.....</b>	<b>4</b>
2.1.1 Literatür .....	5
2.1.2. Bulanık mantık tarihi gelişimi.....	5
2.1.3. Bulanık mantık uygulama alanları .....	6
2.1.4. Bulanık sistemlerinin gelişimi.....	7
2.1.5. Klasik ve bulanık kümeler kavramı .....	7
2.1.6. Bulanık sonuç çıkarma .....	9
<b>2.2. Bulanık Modeller .....</b>	<b>10</b>
2.2.1. Mamdani bulanık modeli .....	10
2.2.2 Sugeno modeli .....	11
2.2.3. Tsukamoto modeli .....	12
<b>3. BULANIK SİSTEM ÖRNEKLERİ .....</b>	<b>13</b>
<b>3.1 Örnek-1.....</b>	<b>13</b>
<b>3.2. Örnek-2 .....</b>	<b>15</b>
<b>3.3. Örnek-3 .....</b>	<b>17</b>
<b>3.4. Örnek-4 .....</b>	<b>20</b>
<b>3.5. Örnek-5 .....</b>	<b>23</b>
<b>3.6. Örnek-6 .....</b>	<b>26</b>

<b>4. VERİYE DAYALI BULANIK SİSTEM MODELLEME.....</b>	<b>29</b>
<b>4.1. Veriye Dayalı Modelleme Örnekleri.....</b>	<b>31</b>
4.1.1. Dinamik sistem modelleme örnek 1.....	31
4.1.2. Dinamik sistem modelleme örnek 2.....	35
4.1.3 Dinamik sistem modelleme örnek 3.....	40
4.1.4. Dinamik sistem modelleme örnek 4.....	43
4.1.5. Dinamik sistem modelleme örnek 5.....	47
<b>5. SONUÇ ve ÖNERİLER.....</b>	<b>51</b>
<b>KAYNAKLAR.....</b>	<b>52</b>
<b>ÖZGEÇMİŞ</b>	

## SIMGELER VE KISATMALAR DİZİNİ

### Simgeler

A,B: Bulanık Kümeler

K: Bir kural tabanındaki kural sayısı

R : Bulanık ilişki veya kural

a, b: TS modelindeki sonuç parametre vektörleri

q: Bulanık model (statik) çıktı sayısı

p: Bulanık model (statik) giriş sayısı

x: Giriş vektörü

y: çıkış vektörü

$u(k), y(k)$ : Dinamik sistemin k ayrık anında giriş ve çıkışı

$\beta$  : Bir kuralın yerine getirilmesi derecesi

$\gamma$  : Normalleştirilmiş tatmin derecesi

$\mu, \mu(\cdot)$  : üyelik derecesi, üyelik fonksiyon

X, Y :  $x$  ve  $y$  değişkenlerinin evrenleri

### Kısaltmalar

LSE: En küçük kareler ihtimali

MISO: Çok girişli tek çıkışlı

ÖDS: Örnek dinamik sistem

SISO: Tek girişli tek çıkışlı

TS: Tagaki Sugeno

TSK: Tagaki Sugeno ve Kang modeli

ÜF: Üyelik fonksiyon

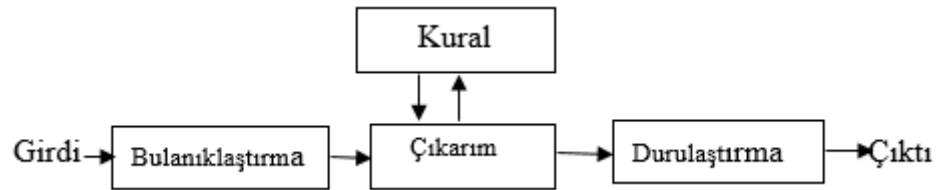
## ŞEKİLLER DİZİNİ

<b>Şekil 1.1.</b> Bulanık mantık sisteminin çalışma mantığı (Sarı, Murat ve Kırabalı)... <b>Hata!</b> <b>Yer işareti tanımlanmamış.</b>	
<b>Şekil 2.1.</b> Bulanıklaştırma-durulaştırma birimli bulanık sistem (Aydın).....	4
<b>Şekil 2.2.</b> Klasik küme aitliğine örnek bir grafik (Allahverdi, 2001).....	8
<b>Şekil 2.3.</b> Bulanık bir kümeye aitliğe ait örnek bir grafik (Allahverdi, 2001).....	9
<b>Şekil 2.4.</b> Bulanık sonuç çıkarım sisteminin blok şeması (Yılmaz).....	10
<b>Şekil 3.1.</b> Örnek-1 için TS bulanık sistemi öbek gösterimi.....	13
<b>Şekil 3.2.</b> $y(k)$ girişi için belirlenmiş üçgen üyelik fonksiyonları.....	13
<b>Şekil 3.3.</b> Bulanık sistemin çıkışı (a) ve kullanılan birinci giriş dizisi (b).....	15
<b>Şekil 3.4.</b> Bulanık sistemin çıkışı (a) ve kullanılan ikinci giriş dizisi (b).....	15
<b>Şekil 3.5.</b> Modelin giriş değişkenlerinin üyelik fonksiyonları.....	17
<b>Şekil 3.6.</b> Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.....	17
<b>Şekil 3.7.</b> Modelin giriş değişkenlerinin üyelik fonksiyonları.....	20
<b>Şekil 3.8.</b> Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.....	20
<b>Şekil 3.9.</b> Modelinin giriş değişkenlerinin üyelik fonksiyonları.....	21
<b>Şekil 3.10.</b> Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.....	23
<b>Şekil 3.11.</b> Örnek-5 için giriş (a, b) ve çıkış (c) değişkenlerinin üyelik fonksiyonları..	25
<b>Şekil 3.13.</b> Modelinin giriş değişkenlerinin üyelik fonksiyonları.....	28
<b>Şekil 3.14.</b> Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.....	28
<b>Şekil 4.1.</b> ÖDS 1 için eğitim (a) ve test (b) giriş $u(k)$ ve çıkış $y(k)$ dizileri.....	35
<b>Şekil 4.2.</b> ÖDS 1'in modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.....	35
<b>Şekil 4.3.</b> Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.....	35
<b>Şekil 4.4.</b> ÖDS 2 için eğitim (a) ve test (b) giriş $u(k)$ ve çıkış $y(k)$ dizileri.....	39
<b>Şekil 4.5.</b> ÖDS 2'nin modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.....	39
<b>Şekil 4.6.</b> Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.....	39
<b>Şekil 4.7.</b> ÖDS 3 için eğitim (a) ve test (b) giriş $u(k)$ ve çıkış $y(k)$ dizileri.....	42
<b>Şekil 4.8.</b> ÖDS 3'in modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.....	42
<b>Şekil 4.9.</b> Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.....	43

<b>Şekil 4.10.</b> ÖDS 4 için eğitim (a) ve test (b) giriş $u(k)$ ve çıkış $y(k)$ dizileri.....	46
<b>Şekil 4.11.</b> ÖDS 4'ün modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.....	46
<b>Şekil 4.12.</b> Eğitim (a) ve test (b) verisi için elde edilen modelleme başarıımı.....	47
<b>Şekil 4.13.</b> ÖDS 5 için eğitim (a) ve test (b) giriş $u(k)$ ve çıkış $y(k)$ dizileri.....	50
<b>Şekil 4.14.</b> Eğitim (a) ve test (b) verisi için elde edilen modelleme başarıımı.....	50

## 1. GİRİŞ

İnsanoğlu günlük hayatta bazı problemlerle karşılaşmakta ve bunlara çözüm bulmak amacıyla daha eski bilgi ve tecrübelerinden yardım almaktadır. Problemlere bazen basit bir şekilde çözüm bulunurken bazılarında çözüm bulmakta zorlanılabilir, işte tam bu noktada geleneksel anlayıştan modern anlayışa doğru bir geçiş başlatılmıştır. 1930’larda Max Black bu anlamda lider kavramlar geliştirmiş olsa da günümüzde Zadeh modern anlamda belirsizlik kavramının önemli noktasını tanımlamıştır. Yine günümüz teknolojisinde yaygın kullanılan akıllı ve uzman sistemlerle otomasyonda belirsizlik durumunda karar verme sürecinde bulanık modeller temel alınır. Bulanık kümeler baktığımızda, aynı anda sözel ve sayısal bilgilerle insan aklını en yakın şekilde modellemesi onun en büyük özelliğidir (Sarı, Murat ve Kırabalı). Birçok ülkede teknoloji alanında örneğin robotlar, trafik ışıkları, elektrik süpürgesi, soğutucular, iş yönetimi, uzaktan algılama, çamaşır makinesi gibi çok geniş bir alanda kullanılan bu kontrol sistemleri bulanık mantık ve bulanık mantığın modelleriyle tasarlanarak günümüzde önemli bir yer tutarlar. Bulanık kümeye dayalı modelleme çoklu değerler ile değerlendirildiğinden gerçeğe yakın olduğundan problemin çözümüne kolaylık getirir. Oysa klasik kümede sadece iki değer vardır. Bundan dolayı problemleri klasik kümede çözmek zordur. Bunun gibi problemlere çözüm getirmek için bulanık mantık yaklaşımı uygulamacılar ve tasarımcılar tarafından kullanılmaya başlanmıştır. Şekil 1.1’de bulanık bir sistemin genel öbek yapısı verilmiştir.



**Şekil 1.1.** Bulanık mantık sistemin genel öbek yapısı (Sarı, Murat ve Kırabalı).

Bulanık modellemede problemin belirlenmesi ve parametrelerin belirlenerek üyelik fonksiyonlarının (ÜF) oluşturulması modelin ilk aşamasıdır. Sonrasında bu parametreler problemin çözümü için belirlenen kurallar dizisi veya kural tabanı ile oluşturulur. Daha sonra çıkarım yöntemi seçilir ve son olarak durulama yöntemi

belirlenir. Şekil 1.1’de gösterilen bulanık sistem grafiğinde girdiler ÜFler ile bulanıklaştırılır ve seçilen çıkarım ve kural tabanı yardımıyla çıkarım yapılarak bulanık sonuçlar durularak keskin çıkış haline getirilir.

Gerçek sistemlerin matematiksel modelleri, birçok mühendislik ve bilim dalının temel ilgi alanlarından biri olarak görülmüştür. Sistem modelleri, sistem davranış analizi ve benzetiminde, yeni sistem tasarımında, sistemlerin denetlenmesinde vb. alanlarda kullanılır. Bu sistem modellerinin, gerçeğine mümkün olduğunca uygun olarak geliştirilmesi önemli bir görevdir. Eğer model yeterince doğru değil ise özellikle yeni bir sistemin tasarım aşamasında ciddi sorunlar ortaya çıkar. Gerçeği yeterince yansıtan model elde etmek ile modelin karmaşıklığı arasında doğrusal bir orantı vardır. Eğer model çok basit ise üzerinde çalışılan sistemi doğru bir şekilde temsil edemez. Bu da yukarıda zikredilen kullanım amaçlarına hizmet edemeyeceği sonucunu doğurur. Diğer taraftan, modelin çok karmaşık olması da uygulamada kullanılabilirliğini kısıtlar.

Çoğu alışlagelmiş sistem modelleme yaklaşımlarının ortak sakıncası, üzerinde çalışılan sistem hakkında deneyime sahip işletici ve/veya mühendislerin bilgi birikimlerini etkin bir şekilde kullanımını sağlayamamalarıdır (Babuska, 1998). İnsanın düşünüş tarzıyla karmaşık görevleri belirsizlik altında yönetebilme kabiliyeti, bu düşünüş tarzından esinlenerek değişik modelleme kipleri arayışını tetiklemiştir.

Bu tetikleyiş biyolojik sistemler ve insan zekâsından esinlenerek bulunan teknikleri çalıştıran akıllı yöntemleri doğurmuştur. Bulanık modellemede bu yöntemlerden biridir. Dilsel değişkenler -ya da daha anlaşılır olarak bulanık kümeler-kullanarak sistem modelleme kavramı ilk kez Zadeh (Zadeh, 1973) tarafından tanımlanmış olup günümüze kadar ciddi bir gelişme göstermiştir.

Bulanık modellemenin yapay sinir ağları gibi bilinen diğer akıllı modelleme yöntemlerine kıyasla üstün yanları mevcuttur. Kısaca bu üstünlükler:

- Bulanık modeller, bilginin işlenmesini genel fonksiyon yaklaşımalarının matematiksel özellikleri ile birleştirebilirler.
- Bulanık modeller, kural tabanlı sistemler olarak üzerinde çalışılan sistem hakkında uzmanlık bilgisini model içine katmaya uygun esnek matematiksel yapıya sahiptir.

- Bulanık modeller, geleneksel regresyon yöntemleri ile birleştirilip veri-sürümlü öğrenme algoritmalarının kullanılmasını sağlayabilirler.

Bu çalışmada, bulanık modellemenin yukarıda verilen üstünlüklerini veri-sürümlü eğrisel dinamik sistem modelleme problemi üzerinde sergilemek amaçlanmıştır. Bu amaç için, öğrenme algoritmalarını koşturmaya uygun esnek matematiksel yapıya sahip olması sebebiyle sıklıkla kullanılan Takagi-Sugeno (TS) bulanık modeli kullanılmıştır (Tagaki ve Sugeno, 1985). TS bulanık model ile örnek eğrisel dinamik sistemlerin geleneksel LSE öğrenme yöntemiyle modellenmesi irdelenmiştir.

## 2. BULANIK MANTIK VE BULANIK MODELLER

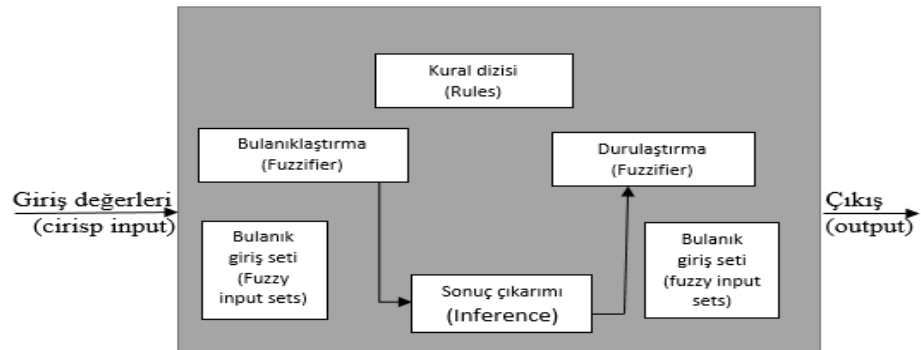
Mantık eskiden günümüze bazı gelişmeler göstermiştir ve zamanımızın gelişmelerine uyum sağlamaya çalışmıştır. Bu gelişmelerle bulanık mantık ihtiyacı ortaya çıkmıştır. Bulanık mantıkta ihtiyaçlara göre bazı modeller zamanla ortaya çıkmıştır, bu modellere de bulanık modeller denir.

### 2.1. Bulanık Mantık

Bulanık sistem bulanık mantığı esas alan bir modeller serisidir. Bu sistemlerde elemanlar ve kümeleri olan yaklaşma derecesi önemlidir. Bulanık mantığı klasik mantıkla kıyaslamak gerekirse örneğin pazartesi günü hafta sonu kümesine ait değildir ve bu 0 durumuna karşılık gelir. Aynı şekilde Pazar günü hafta sonu kümesine ait olduğundan bu 1 durumuna karşılık gelir.

Bulanık mantıkta bir eleman birden fazla kümeye ait olabilir. Buna göre pazartesi günü %45 oranı ile hafta sonu kümesine, %55 oranı ile hafta içi kümesine dâhil olurken, Pazar günü %95 hafta sonu kümesinde, bir sonraki günden ötürü %5 oranı ile hafta içi kümesine dahil olmaktadır (Allaahverdi, 2011). Bu oranlar önceden belirlenen kurallar vasıtasıyla sisteme tanımlanır. Bir bulanık sistem iki temel öğeden oluşur. Bunlar bulanık kümeler ve kurallardır.

Bulanık kümelerle belirlenen kurallarla sistem girdileri, istenen çıktılara dönüştürülmektedir. Bu girdiler Şekil 2.1'den görüleceği üzere sırayla, bulanıklaştırma, bulanık çıkarım ve durulaştırma aşamalarından geçer.



Şekil 2.1. Bulanıklaştırma-durulaştırma birimli bulanık sistem (Aydın).

Kurallar dizisinden bakılacak olursa, örneğin bir sürücüsüz aracın giriş değerlerine göre karar mekanizmasını tanımlayacak kurallar olmalıdır. Giriş değerlerini verilen derecelerle üyelik fonksiyonları hazırlanır. Üyelik fonksiyonları bir elemanın bir alt kümeye olan yakınlığını  $[0, 1]$  arasında ölçeklendirmeye yarar (Köşe, 2017). Bir bulanık sistem dışarıdan aldığı değişkenlerle karar verme mekanizmasını sağlayan üyelik fonksiyonlarını hazırlayarak ve kural süzgecinden geçirerek en uygun çıktıyı almayı sağlamaktadır. Örneğin bu değişkenleri araç hızı için düşünürsek bunlar düşük, normal, yüksek olabilmektedir. Hazırlanan üyelik fonksiyonlarında ise bu değişkenler, elemanların kümelere olan yakınlık uzaklık dereceleri belirler.

### 2.1.1 Literatür

Bulanık küme kavramı Prof. Dr. Lofti Asker Zadeh tarafından “Fuzzy Set” adı altında 1965 yılından önce getirilmiştir ve bu yeni alanla ilgili çok sayıda araştırma ve uygulama yapılmıştır (Elmas, 2011). Bulanık küme farklı üyelik derecelerine sahip bir sınıf olarak karşımıza çıkar. Günlük hayatımızda kullanılan sınıflandırmaların hemen hemen çoğu bulanıktır. Örnek olarak: {‘Büyük kazanç’}, {‘Yüksek basınç’}, {‘\_ılımlı sıcaklık’} ...60° derece üzerinde insan dayanabilir mi?

### 2.1.2. Bulanık mantık tarihi gelişimi

Bulanık mantık tarihçesi ilk olarak 1920’ler ile 1930’lara dayanır. Doğru ve yanlış ilkesine dayanan sistemler kümesini üçüncü veya daha fazla ihtimal ekleyerek sistemleri daha hassas ifade etmemizi sağlayan sistem olarak bulanık sistem ortaya çıkmıştır. Batılı bilimciler çok değerliliği kabullenmesine rağmen Lukasiewicz Godel ve Black ilk olarak çok değerli bulanık mantık küme sistemleri geliştirdiler. 1930’lerden Jan Lukasiewicz ilk üç değerli sistemi geliştirdi. Yine 1930’larda Max Black sürekli değerlere sahip sistemi kümeler üzerinde uyguladı.

1965’lerde Azerbaycanlı Lutfi Asker Zade ilk bulanık sistem ilkelerini ortaya koyan kişidir. Lutfi Asker Zadenin bu ilkelerine batı dünyası şüphe ile bakmıştır. Çünkü batı kültürünün temelinde Aristo mantığı vardır. Batılılara göre her şey olasılıklar teorisiyle açıklanabilir (Zadeh, 1965).

1970 yıllarda doğu dünyası baştan Japonya olmak üzere bulanık mantığı birçok teknolojik malzemelerinde kullanmıştır. İlk defa 1975 yılında bulanık sistemler buhar makinesinde denenmiştir, ayrıca bu sistemler elektronik cihazlarda da kullanılmıştır.

Çamaşır ve bulaşık makineleri, asansörler, fotoğraf makineleri, yapay zekâ çalışmaları vb. gibi pek çok konuda bu sistemin ilkeleri kullanılmıştır. Örneğin, 1987 yılında Sendai metrosunda bulanık mantık yoluyla istenen yerde durması üç kat iyileştirilmiş ve kullandığı enerji %10 düşürülmüştür (Ertunç, 2012). 1988 yılında yine bulanık mantığa dayalı bir uzman sistem Kara Pazar adlı Tokyo borsasındaki krizi 18 gün önceden belirlemiştir. Bulanık mantığın başarılı olmasıyla beraber dünyanın ilgisi artmış ve 1989 yılında Tomson, IBM, Toshiba SGS, Omron, Hitaci, NCR, Matsuhita gibi 51 firma LİFE (labrotory for interchange fuzzy engineering) laboratuvarları kurulmuştur (Çobanoğlu, 2000).

### **2.1.3. Bulanık mantık uygulama alanları**

Bulanık mantık sistemleri su arıtma ve çimento sanayisinde ilk olarak kullanılmıştır, sonrasında vinç denetimi, nükleer reaktör; asansör, buhar türbini alanlarında uygulanmıştır. 1988 yılında Japonya’da yaygın olarak bulanık sistemler kullanılmıştır Japonya’daki Sendai metrosu için geliştirilen sistem başarılı olmuş akabinde çok rağbet görmüştür daha sonra fotoğraf makinelerinde ev aletlerinde, borsa denetim sistemlerinde uygulanmıştır (Bozkurt, 2012).

Örneğin günümüz kameralarının çekim sırasında sarsıntıların ortadan kaldırılması, Nissan firmasının ABC fren sistemleriyle aracın tekerinin kilitlemeden durması, Toshiba, Hitachi, Fujitsu, Mitsibishi, asansör denetim sistemlerin, çamaşır makinelerindeki çamaşırların cinsine, kirliliğini ve ağırlığına göre program seçme, ses komutlarıyla bir helikopterin hareket ettirilmesi gibi birçok alanda bulanık mantık kullanılmıştır.

Ayrıca bulanık mantık biyoloji ve tıp biliminde, yönetim ve karar desteklerinde, ekonomi ve finansta araştırma çalışmalarında, kalite kontrolünde, mühendislikte psikolojide, sinyal işlemede, süreç planlamada, veri analizinde, veri tabanı uygulamalarında lineer ve nanlineer kontrol sistemlerinde, kuantum fiziğinde, çevre biliminde, kalıp tanımı ve sınıflandırmada yaygın bir şekilde kullanılır. Bulanık mantığın bu alanlarda kullanım sayesinde örneğin kameralar görüntüye odaklanıyor, televizyon alıcıları ayarlanıyor, robot kolları ayarlanıyor, buzdolaplarının buzlanması engelleniyor, metrolardaki süreçler kontrol ediliyor, trafik lambaları programlanıyor (Karakuzu, 2010),

otomobillerin fren sistemleri süspansiyonları, motorları kontrol ediliyor, asansörler denetlenebiliyor.

#### **2.1.4. Bulanık sistemlerinin gelişimi**

Bulanık sistemler günümüzde yüzlerce hatta daha fazla ticari ve diğer alanlarda uygulanmıştır ve dünyada kullanımı önemli oranda artmıştır. İlk uygulamalara baktığımızda 1974 yılında buhar makinesinin denetiminde uygulanmıştır ve ardından önemli uygulamaların da başlamasına zemin hazırlamıştır. 1980'lerde çimento fırınlarında denetim için yine bulanık mantık uygulanmıştır. 1987 IFSA kongresinde ilk bulanık sistem denetleyicileri gösterilmiştir. Daha sonra bu denetleyiciler daha da olgunlaşarak Omron firması tarafından 700'den fazla uygulamada kullanılmıştır (Elmas, 2011). Yine aynı şirketin Japonya'da bulanık sonuç-boardla yapılan depolama, tekrar etme ve sonuçları için (RISC) bilgisayara dayalı çalışmaları olmuştur. 1990'larda çamaşır makinasının denetiminde en etkili yıkama programın kullanılması bulanık sistemle sağlanmıştır. Günümüzde birçok elektronik pazarında çoğu ürün bu bulanık sisteme dayanır ayrıca elektrik akımı fiziksel ve kimyasal süreçlerde sıvı gaz akımı denetiminde bulanık mantık uygulanmaktadır. Bulanık sistemlerde kuralların belirlenmesi kolaydır ve bundan dolayı bulanık mantık denetleyiciler az kuralla yüksek başarımlar sağlarlar. Eğer bir sistemin davranışı kurallar ile tanımlanabiliyorsa veya karışık matematiksel işlem gerekmiyorsa uygulanacak sistem için bulanık mantık kullanılabilir.

#### **2.1.5. Klasik ve bulanık kümeler kavramı**

Klasik mantıkta olay var-yok veya 0-1 veya doğru-yanlış gibi sadece iki ihtimal ya da önermeden bahsedilir. Bulanık mantıkta ise klasik mantıktan farklı olarak olaylar daha çok ihtimale dayanır. Bulanık kümede bir olay var-az var- çok var gibi esnek sonuçlar vardır. Klasik kümede bir olayı 0-1 ile gösterirken bulanık mantıkta 0-1 ve arasındaki sonsuz değerle veya sonuçla gösterebiliriz (Şen, 2016).

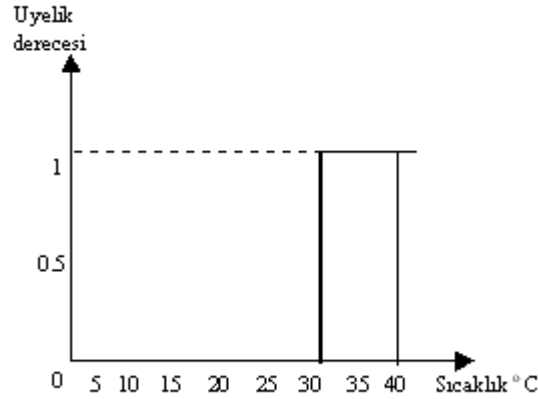
- **Klasik kümeler**

Klasik mantık kesin değerlere sahip önermelerle ilgilenir. Yani 'doğru' ya da 'yanlış' tam birisine sahiptir. Bilgisayarın çalışma mantığında sıfır ya da bir değerlerine bağlı olarak bir işlem kesin ifadelerle ya da değerlerle açıklanır. Klasik kümede iki değerden başka üçüncü bir değer durumu yoktur. Bu nedenle klasik küme elemanlarının üyelikleri  $\{0,1\}$  değerlerini alırlar. Klasik küme de bir değer ya da varlık kümenin elemanıdır ya da

değildir. Matematiksel olarak anlatmak gerekirse bir elemanın kümeye aitliği kümenin elemanı olduğunda '1', kümenin elemanı olmadığıda '0' değerini alır. Örneğin bir odanın sıcaklığını 25 derece kabul ettiğimizde klasik kümeye göre bunun üzerindeki sıcaklık değerlerini sıcak kabul ederiz ve bu derecelerin sıcak kümesindeki üyelik dereceleri '1' olarak kabul edilir. 25 derecenin altındaki değerleri soğuk kabul ederiz ve bu derecelerin sıcak kümesindeki üyelik dereceleri '0' olarak kabul edilir (Elmas, 2011).

$$\mu_A(x) = \begin{cases} 1 & \text{eger } x \in A \\ 0 & \text{eger } x \notin A \end{cases} \quad (2.1)$$

Klasik kümede üçüncü bir durum imkânsızdır fakat çoğu zaman bu tür durumlar paradoks olarak tanımlanır. Klasik kümede var yok mantığı ya da '0' '1' mantığı vardır ve bu değerlerin arasındaki değerler kümeye dâhil değildir. Sınırları kesindir üyelik dereceleri '0' ya da '1'dir. Şekil 2.2'de örnek bir klasik küme üyelik grafiği verilmiştir.



Şekil 2.2. Klasik küme aitliğine örnek bir grafik (Allahverdi, 2001).

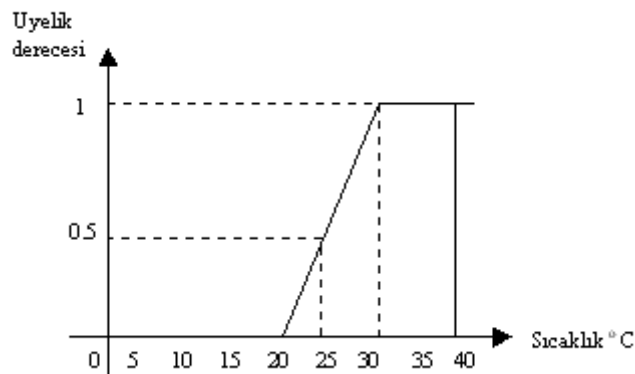
- **Bulanık küme**

Bulanık küme mantığında klasik kümenin aksine ara aitlik değerleri de alınır. Yani '0' '1' değerleri arasındaki değerler de kümeye aitliği ifade eder. Bulanıklık bir durumun belirsizliğini ifade eder, yani durumun olup olmadığını değil hangi dereceye kadar olduğunu belirtir. Bulanıklık deterministik olmasını rağmen olasılık tahminselidir. Bulanık kümelerde kesin sınırları olmayan kümelerdir yani kümeye ait olandan olmayana geçiş kademeli olur. Örneğin 'su sıcak' veya 'sıcaklık çok yüksek' gibi modellemede

dilsel tanımlamalara esneklik kazandıran bu geçiş üyelik fonksiyonları olarak tanımlanır (Beykal, Beyan, 2004)

$$\mu_A(x): x \rightarrow [0,1] \text{ yani } 0 \leq \mu_A(x) \leq 1 \quad (1.2)$$

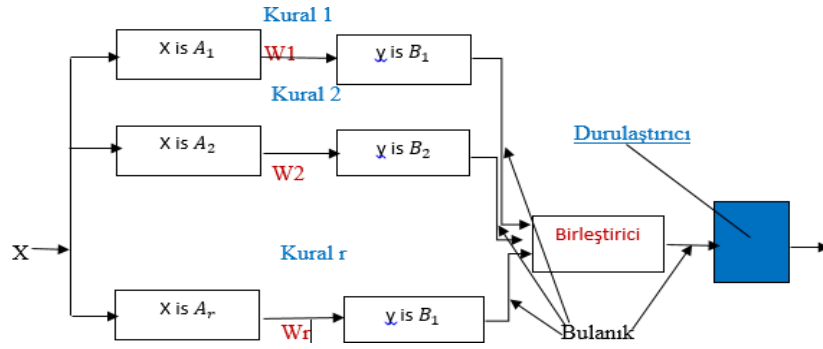
Yukarıda bulanık mantığın matematiksel modeli gösterilmiştir. Burada  $\mu_A(x) = 0$  olması  $x$ 'in  $A$  ya ait olmadığını,  $\mu_A(x) = 1$  olması  $x$ 'in  $A$ 'nın üyesi olduğunu gösterir.  $\mu_A(x) = 0,5$  değeri bulanık  $A$  kümesinin geçiş noktasıdır. Böylece klasik kümede her hangi bir nesne kümeye aittir ya da ait değildir kavramı bulanık kümelerde kısmen vardır. Şekil 2.3'de bulanık bir kümeye ait üyelik fonksiyonu grafiği verilmiştir. Şekilden de görüleceği üzere ait olandan olmayan bir geçiş vardır.



Şekil 2.3. Bulanık bir kümeye aitliğe ait örnek bir grafik (Allahverdi, 2001).

### 2.1.6. Bulanık sonuç çıkarma

Klasik mantıktan verilen değerlerden bir sonuç çıkarmaya bulanık çıkarım denir. Örneğin kuşlar, şahin bir kuştur önermelerini karşılık, şahin uçar çıkarımını yapabiliriz. Bulanık sistemlerde karara ve sonuca varma (if then ) kuralı ile mümkündür çünkü bulanık sistemlerde girişler ikiden fazla değişkenden oluşur. Bulanık sistemlerde elde edilen sonuçlar kurallarla yorumlanır ve yeni bir sonuca varılır (Yılmaz). Girişteki üyelik dereceleri sonuçların ne oranda geçerli olacağını belirler bu belirlemede içerme (implication) mantığı vardır. Kurallardan elde edile sonuçlar birleştirilerek birleşik (aggregation) sonuç kümesi elde edilir buna bulanık sonuç çıkarma (fuzzy inference ) denir. Şekil 2.4'de bu işleyiş görsel olarak gösterilmiştir.



Şekil 2.4. Bulanık sonuç çıkarım sisteminin blok şeması (Yılmaz).

Bulanık sonuç çıkarım sistemleriyle ilgili birkaç yöntem vardır. Bunlar Mamdani modeli, Tagaki-Sugeno-Kang (TSK) modeli ve Tsukamoto modelleridir. Bulanık çıkarım üyelik fonksiyonlarının, if-then kurallarının, üyelik değerlerinin, bulanık mantık operatörlerinin kullanıldığı bir işlemdir (Yıldırım, 2012). Birçok çıkarım metodu vardır bunlardan Mamdanın minimum çıkarım metodunun matematiksel modeli aşağıdaki gibidir.

$$\mu_{B'} = \max_{1=1} M [\min(\mu_{A'_1}(x), \mu_{A'_2}(x), \dots, \mu_{A'_n}(x), \mu_{B'}(x))] \quad (2.3)$$

## 2.2. Bulanık Modeller

Bulanık mantığın ortaya çıkmasından sonra ihtiyaca göre bazı bulanık modeller ortaya çıkmıştır. Bunlardan bazıları Mamadani, Sugeno, Tsukamoto modelleridir. Bu modeller arasındaki fark sonuç kısmında yatmaktadır ve birbirine benzememektedir.

### 2.2.1. Mamdani bulanık modeli

Her hangi bir andaki değerlere göre x ve y girişlerini tanımlandığı bir bulanık kümede bu girişlerin üyelik dereceleri belirlenir. Bu x ve y değerlerinin üyelik dereceleri min operatöründen geçirilir, sonuçta en küçük üyelik derecesine göre kuralın işaret ettiği çıkış bulanık kümesi belirlenir. Bu işlem bütün kurallar için ayrı ayrı işletilir ve kural sayısı kadar çıkış bulanık kümeleri elde edilir. Bu çıkış, bulanık kümeleri max operatöründen geçirilir. Sonuçta yine bir bulanık küme elde edilir daha sonra sistem çıkışında keskin değere ulaşmak için sonuç çıkış bulanık kümesi durulandırma işleminden geçirilmelidir (Karakuzu, 2014).

### Mamdani bulanık sistem tipik kuralları

Çıkış üyelik fonksiyonu için bazen tek-ton üyelik fonksiyonu olarak adlandırılan tek bir değer kullanılabilir. Aşağıda, dört kuralı, iki girişi ve bir çıkışı olan “Mamdani bulanık modeli” kural örneği verilmektedir () .

X küçük ve Y küçük ise o halde Z büyük negatif.

X küçük ve Y büyük ise o halde Z küçük negatif.

X büyük ve Y küçük ise o halde Z küçük pozitif.

X büyük ve Y büyük ise o halde Z büyük pozitif.

Durulandırma: çıkış bulanık kümesinden keskin değerler elde etme yoludur.

Durulandırmada en sık kullanılan stratejiler şunlardır:

- Ağırlık merkezi (centroid of area) yöntemi
- Alan açıortayı (bisector of area) yöntemi
- Maksimum ortalama (mean of maximum ) yöntemi
- Maksimumun en küçüğü (smallest of maximum) yöntemi
- Maksimumun en büyüğü (largest of maximum) yöntemi

#### 2.2.2 Sugeno modeli

Sugeno, Mamdani’den farklı olarak ayrıca durulandırma işlemi gerektirmeyen sistematik bir bulanık çıkarım yöntemidir. Sugeno bulanık çıkarım modeli ilk defa 1985’te kullanılmıştır (İnaç, 2014). Bu modelde çıkış ÜFleri yerine her bir kural için birer polinom tanımlanır. Aşağıda bu modele ilişkin örnek bir kural verilmiştir.

Eğer x, A ve y de B ise o halde  $z = f(x,y)$

Mamdani ve Sugeno arasındaki esas fark, Sugeno yönteminde üyelik fonksiyonların çıktısı ya lineer ya da sabit bir değer almaktır. Çıktının lineer olması durumuna, “Birinci-derece Sugeno bulanık çıkarımı”, çıktının sabit olması durumuna, “Sıfır-derece Sugeno bulanık çıkarımı” denilir. Sugeno modelinin en sık kullanan, her bulanık kuralın çıktısının sabit bir değer aldığı “Sıfır derece Sugeno bulanık çıkarımı”dır. Mamdani ve Sugeno arasındaki temel fark Sugeno çıkış üyelik fonksiyonlarının doğrusal veya sabit olmasıdır.

Her kuralın çıkış değeri  $z_i$ , kuralın ateşleme gücü  $w_i$  ile ağırlıklıdır. Sistemin keskin çıkışı, tüm kural çıktılarının ağırlıklı ortalamasıdır, Eşitlik (2.4)'deki gibi hesaplanır.

$$\text{Son çıkış} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (2.4)$$

Aşağıda, dört kurallı, iki girdisi ve bir çıktısı olan “birinci dereceden Sugeno bulanık model” örneği gösterilmektedir.

*X küçük ve Y küçük ise o alde*  $z = -x + y + 1$

*X küçük ve Y büyük ise o halde*  $z = -y + 3$ .

*X büyük ve Y küçük ise o halde*  $z = -x + 3$ .

*X is büyük ve Y büyük ise o hslde*  $z = x + y + 2$ .

Yukarıdaki örnek için çıktı yüzeyi dört düzlemin birleşiminden oluşur ve her bir düzlem, bir bulanık kuralın çıktı denklemi ile belirlenmektedir.

### 2.2.3. Tsukamoto modeli

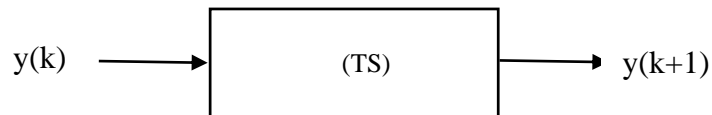
Tsukamoto modelinde tüm kuralların sonuçları ÜFlü bulanık küme ile tanımlanır. Kuralların çıkışları yine kuralların aktiflik seviyeleri ile tanımlanır. Sistemin keskin çıkışı her bir kuralın çıkışının ağırlıklı ortalaması alınarak bulunur (Karakuzu, 2014).

### 3. BULANIK SİSTEM ÖRNEKLERİ

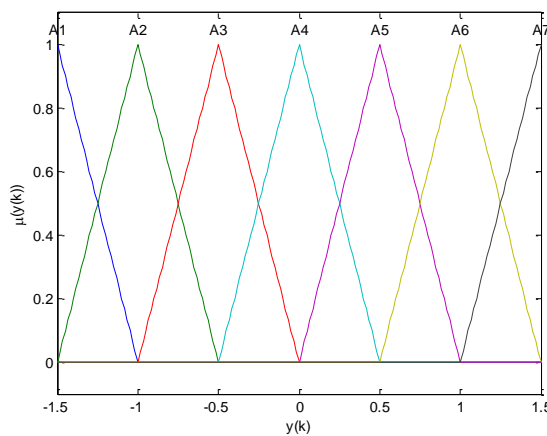
Bulanık sistem günümüzde birçok alanda kullanılmaktadır. Örneğin biyoloji ve tıpta, mühendislikte, araştırmalarda, karar destek yönetiminde ve daha birçok alanda kullanılmaktadır. Aşağıda birçok bulanık sistem örneği mevcuttur. Bu örnekler belirli kurallar dâhilinde bize çeşitli sonuçlar vermektedir. Örneklerimizde Sugeno ve Mamdani yöntemleri kullanılmıştır.

#### 3.1 Örnek-1

Örnek sistemimiz 7 adet bulanık kural içermekte olup, kural parametre vektörü  $\theta = [a_1 b_1 a_2 b_2 a_3 b_3 \dots a_7 b_7]_{K(p+1)=7 \times (1+1)=14}^T$  şeklindedir. Ağırlıklı en küçük kareler kestirimi ile TSK bulanık sistemin kural/sonuç parametreleri  $a^T = [0.8992 \ 1.0184 \ 1.1399 \ 0.5092 \ 1.1689 \ 0.9973 \ 0.9924]$ ,  $b^T = [-0.1076 \ 0.0206 \ 0.1813 \ -0.0214 \ -0.1253 \ 0.0045 \ 0.0101]$  değerleri (Babuska) kullanılarak bu sistem benzetimi yapılmıştır. Sistem Şekil 3.1'den de görüleceği üzere SISO bir sistemdir. Sistem bir sonraki ayrık adımdaki değeri önceki adımdaki değeri kullanarak bir adım kestirimci gibi çalışmaktadır. Şekil 3.2'de sistemin giriş ÜFleri verilmiştir. Bu şeklin altında da kullanılan kurallar listelenmiştir. Bu yapı itibarıyla aşağıda verilen Matlab koduyla farklı  $u(k)$  giriş dizisi kullanılarak elde edilen sonuçlar Şekil 3.3 ve 3.4'de ayrı ayrı verilmiştir.



Şekil 3.1. Örnek-1 için TS bulanık sistemi öbek gösterimi.



Şekil 3.2.  $y(k)$  girişi için belirlenmiş üçgen üyelik fonksiyonları.

$R_i : IF y(k) \text{ is } A_i \text{ then } y(k+1) = a_i y(k) + b_i u(k)$

$R_1 : IF y(k) \text{ is } A_1 \text{ then } y(k+1) = a_1 y(k) + b_1 u(k)$

$R_2 : IF y(k) \text{ is } A_2 \text{ then } y(k+1) = a_2 y(k) + b_2 u(k)$

$R_3 : IF y(k) \text{ is } A_3 \text{ then } y(k+1) = a_3 y(k) + b_3 u(k)$

$R_4 : IF y(k) \text{ is } A_4 \text{ then } y(k+1) = a_4 y(k) + b_4 u(k)$

$R_5 : IF y(k) \text{ is } A_5 \text{ then } y(k+1) = a_5 y(k) + b_5 u(k)$

$R_6 : IF y(k) \text{ is } A_6 \text{ then } y(k+1) = a_6 y(k) + b_6 u(k)$

$R_7 : IF y(k) \text{ is } A_7 \text{ then } y(k+1) = a_7 y(k) + b_7 u(k)$

### Matlab kodu

uktrain = 0.333;

yotrain=0.5; uotrain = 0.333;

for k=0:(300-1)

if (k==0)

yotrain(k+1)= yotrain+uotrain\* exp(-3\*abs(yotrain));

else

yotrain(k+1)=yotrain(k)+uktrain(k)\*exp(-3\*abs(yotrain(k)));

end

end

subplot(2,1,1); plot(yotrain);

subplot(2,1,2); plot(uktrain);

yotest=0.5833; uotest = 0.22;

for k=0:(300-1)

if (k==0)

yotest(k+1)= yotest+uotest\* exp(-3\*abs(yotest));

else

yotest(k+1)=yotest(k)+uktest(k)\*exp(-3\*abs(yotest(k)));

end

end

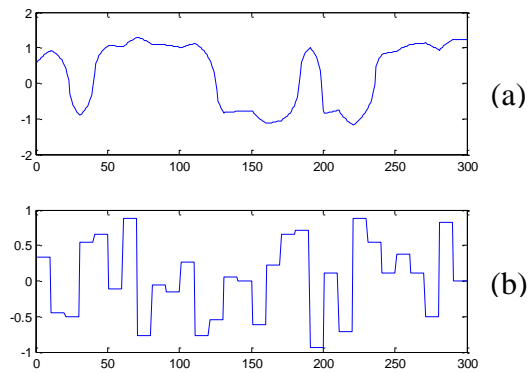
figure (2);

subplot(2,1,1); plot(yotest);

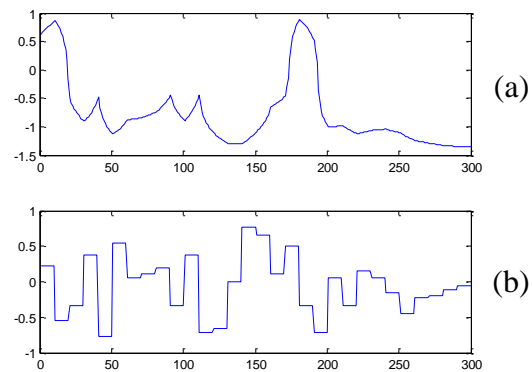
subplot(2,1,2); plot(uktest);

X=[0.5 yotrain(1:299)]';

y=yotrain';



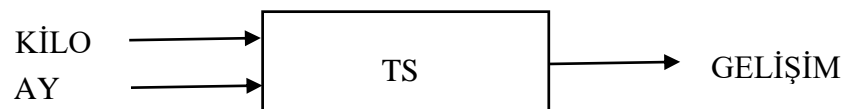
Şekil 3.3. Bulanık sistemin çıkışı (a) ve kullanılan birinci giriş dizisi (b).



Şekil 3.4. Bulanık sistemin çıkışı (a) ve kullanılan ikinci giriş dizisi (b).

### 3.2. Örnek-2

İki giriş tek çıkışlı Sugeno bulanık sistem modeli aşağıda verilen ÜFler ve kurallar ile tanımlanmıştır. Sistemin GELİŞİM çıkışının yüzeyini *min T-norm* operatörü kullanarak *ağırlıklı ortalama* yöntemi ile belirlenmiştir. Şekil 3.5'te giriş ÜFleri, Şekil 3.6'da elde edilen giriş çıkış yüzeyi verilmiştir.



#### Üyelik fonksiyonu tanımlamaları

- KÜÇÜK = Z\_MF (KİLO;2,5)
- ORTA = Gauss-MF (KİLO;5,1)
- BÜYÜK = S-MF(kilo;5,8)
- Emekleme = yamuk (AY;1,1,3,8)
- Emekleme = yamuk (AY;4,9,11,11)

#### Kural polinom tanımlamaları

$$P1 = 0.1 * KİLO + 0.1 * AY + 0.2$$

$$P2 = -0.5 * KİLO - 0.1 * AY + 0.4$$

$$P3 = -0.3 * KİLO + 0.2 * AY + 0.6$$

$$P4 = 0.01 * KİLO^2 - 0.4 * AY + 0.8$$

### ***Kurallar***

(KİLO küçük) ve (AY Emekleme) ise o halde (GELİŞİM P1)  
 (KİLO küçük) ve (AY Yürüme) ise o halde (GELİŞİM P2)  
 (KİLO orta) ve (AY Emekleme) ise o halde (GELİŞİM P2)  
 (KİLO orta) ve (AY Yürüme) ise o halde (GELİŞİM P3)  
 (KİLO Büyük) ve (AY Emekleme) ise o halde (GELİŞİM P3)  
 (KİLO Büyük) ve (AY Yürüme) ise o halde (GELİŞİM P4)

### **Matlab kodu**

```
kilo = 0:0.5:10;
month = 0:0.5:12;

mu_kiloSMALL = zmf(kilo, [2 5]);
mu_kiloMEDIUM = gaussmf(kilo, [1 5]);
mu_kiloBIG = smf(kilo, [5 8]);
subplot(2,1,1);
plot(kilo, mu_kiloSMALL, kilo, mu_kiloMEDIUM, kilo, mu_kiloBIG);
grid
axis([0 10 -0.1 1.1])

mu_monthCRAWLING = trapmf(month, [1 1 3 8]);
mu_monthWALK = trapmf(month, [4 9 11 11]);
subplot(2,1,2);
plot(month, mu_monthCRAWLING, month, mu_monthWALK);
grid
axis([0 12 -0.1 1.1]);

for i = 1:length(kilo)
    for j = 1:length(month)
        kilo_x = kilo(i);
        month_y = month(j);
        KİLO(i,j) = kilo_x;
        MONTH(i,j) = month_y;

        p1 = 0.1 * kilo_x + 0.1 * month_y + 0.2;
        p5 = 2 * (kilo_x)^2 + 4 * (month_y)^2 + 3 * kilo_x + 5 * month_y + 0.9;
        p2 = 0.5 * kilo_x - 0.1 * month_y + 0.4;
        p3 = 0.3 * kilo_x + 0.2 * month_y + 0.6;
        p4 = 0.01 * (kilo_x)^2 - 0.4 * month_y + 0.8;

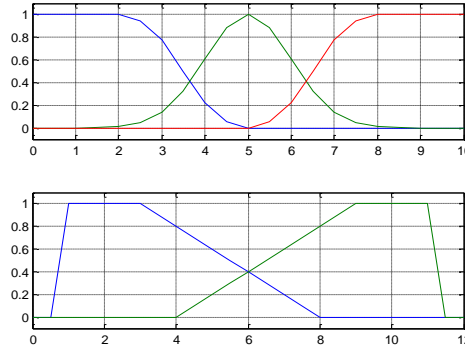
        w1 = min(mu_kiloSMALL(i), mu_monthCRAWLING(j));
        w2 = min(mu_kiloSMALL(i), mu_monthWALK(j));
        w3 = min(mu_kiloMEDIUM(i), mu_monthCRAWLING(j));
        w4 = min(mu_kiloMEDIUM(i), mu_monthWALK(j));
        w5 = min(mu_kiloBIG(i), mu_monthCRAWLING(j));
        w6 = min(mu_kiloBIG(i), mu_monthWALK(j));

        GROWTH(i,j) = (w1 * p1 + w2 * p2 + w3 * p2 + w4 * p3 + w5 * p3 +
w6 * p4) / (w1 + w2 + w3 + w4 + w5 + w6);
```

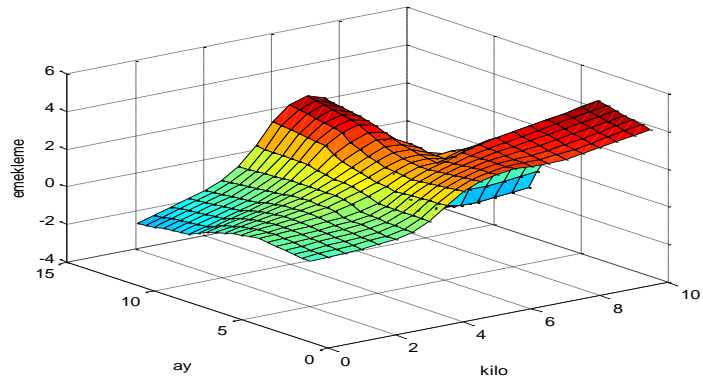
```

end
end
figure(2);
surf(KILO, MONTH, GROWTH);
xlabel('kilo'), ylabel('ay'), zlabel('emekleme');
title('bulanık sistemin yüzey');

```



Şekil 3.5. Bulanık modelin giriş değişkenlerinin üyelik fonksiyonları.



Şekil 3.6. Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.

### 3.3. Örnek-3

İki girişli (X, Y) tek çıkışlı (Z) Mamdani bulanık sisteminin kullandığı kurallar ve ÜFler aşağıda tanımlandığı gibidir. Sistemi gerçeklediği çıkış yüzeyini max (S-normu) cebirsel-çarpım (T-normu) ve ağırlık merkezi durulama kullanarak bulunmuştur. Şekil 3.7'de giriş ÜFleri, Şekil 3.8'de de elde edilen giriş çıkış yüzeyi verilmiştir.



### ***Kurallar***

X küçük ve Y küçük ise o halde Z büyük negatif

X küçük ve Y orta ise o halde Z orta negatif

X küçük ve Y büyük ise o halde Z küçük negatif

X büyük ve Y küçük ise o halde Z büyük pozitif

X büyük ve Y orta ise o halde Z orta pozitif

X büyük ve Y büyük ise o halde Z büyük pozitif

### ***Kural tablosu***

X	Küçük	Orta	Büyük
Y			
NEG	POZ	SFR	NEG
POZ	SFR	POZ	NEG

### ***Giriş değişkenleri ÜF tanımları***

$\mu_{\text{KÜÇÜK}}(x) = \text{yamuk}(x, -5, -5, -1, 1)$ .

$\mu_{\text{BÜYÜK}}(x) = \text{yamuk}(x, -1, 1, 5, 5)$ .

$\mu_{\text{KÜÇÜK}}(y) = \text{yamuk}(y, -5, -5, -3, -1)$ .

$\mu_{\text{ORTA}}(y) = \text{yamuk}(y, -3, -1, 1, 3)$ .

$\mu_{\text{BÜYÜK}}(y) = \text{yamuk}(y, 1, 3, 5, 5)$ .

### ***Çıkış değişkeni ÜF tanımları***

$\mu_{\text{BÜYÜK\_NEGATİVE}}(z) = \text{yamuk}(z, -5, -5, -3.5, -2.5)$ .

$\mu_{\text{ORTA\_ORTA}}(z) = \text{yamuk}(z, -5, -4, -2, -1)$ .

$\mu_{\text{KÜÇÜK\_NEGATİVE}}(z) = \text{yamuk}(z, -3.5, -2.5, -0.5, 0.5)$ .

$\mu_{\text{KÜÇÜK\_POZİTİVE}}(z) = \text{yamuk}(z, -0.5, 0.5, 2.5, 3.5)$ .

$\mu_{\text{ORTA\_POZİTİVE}}(z) = \text{yamuk}(z, 1, 2, 4, 5)$ .

$\mu_{\text{BÜYÜK\_POZİTİVE}}(z) = \text{yamuk}(z, 2.5, 3.5, 5, 5)$ .

Yukarıda tanımlanan bulanık sistem aşağıda verilen Matlab kodu kullanılarak gerçekleştirilmiştir. Sistemin girdilerine göre giriş-çıkış yüzeyi bulunmuştur.

### **Matlab kodu**

```

x= 0:1:10;
y=-3:0.5:3;
z=-5:0.5:5;

mu_xSMALL= trapmf (x, [0 0 4 6]);
mu_xMEDIEUM =trimf(x, [3 5 7]);
mu_xBIG = trapmf(x, [4 6 10 10]);
subplot(3,1,1);
plot (x, mu_xSMALL,x, mu_xMEDIEUM,x, mu_xBIG);
grid
axis([0 10 -1 1.1]);

mu_yneg = trimf (y, [-3 -3 3]);
mu_ypoz =trimf(y, [-3 3 3]);
subplot(3,1,2);
plot (y, mu_yneg,y, mu_ypoz);
grid
axis([-3 3 -1 1.1]);

mu_zNEG= trapmf (z, [-5 -3 -2 0]);
mu_zSFR =trapmf(z, [-2 -1 1 2]);
mu_zPOZ = trapmf(z, [0 2 3 5]);
subplot(3,1,3);
plot (z, mu_zNEG,z, mu_zSFR,z, mu_zPOZ);
grid
axis([-5 5 -1 1.1]);

for i=1:length(x)
    for j=1:length(y)
        x_value=x(i);
        X(i,j)=x_value;
        y_value=y(j);
        Y(i,j)=y_value;

        w1= min(mu_xSMALL(i),mu_zNEG(j));
        w2= min(mu_xSMALL(i),mu_zNEG(j));
        w3= min(mu_xMEDIEUM(i),mu_zSFR(j));
        w4= min(mu_xMEDIEUM(i),mu_zSFR(j));
        w5= min(mu_xBIG(i),mu_zPOZ(j));
        w6= min(mu_xBIG(i),mu_zPOZ(j));

        sonuc_mu_zNEG1 = min(w3*ones(size(mu_zNEG)),mu_zNEG);
        sonuc_mu_zNEG2 = min(w6*ones(size(mu_zNEG)),mu_zNEG);
        sonuc_mu_zPOZ1 = min(w1*ones(size(mu_zPOZ)),mu_zPOZ);
        sonuc_mu_zPOZ2 = min(w5,mu_zPOZ);
        sonuc_mu_zSFR1 = min(w2,mu_zSFR);
        sonuc_mu_zSFR2 = min(w4,mu_zSFR);

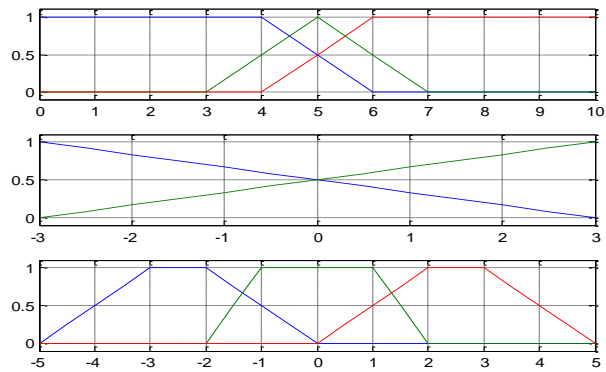
        sonuc_zNEG = max(sonuc_mu_zNEG1,sonuc_mu_zNEG2);
        sonuc_zPOZ = max(sonuc_mu_zPOZ1,sonuc_mu_zPOZ2);
        sonuc_zSFR = max(sonuc_mu_zSFR1,sonuc_mu_zSFR2);
    
```

```

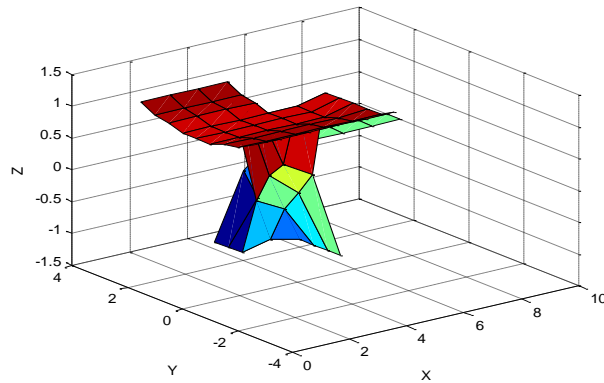
Result_MF=max(sonuc_zNEG,max(sonuc_zPOZ,sonuc_zSFR));

Z(i,j) =sum(Result_MF.*z)/sum(Result_MF);
end
end
figure(2);
surf(X, Y, Z);
xlabel('X'), ylabel('Y'), zlabel('Z');
title('sıfır derece mamadani bulanık sistem');

```



Şekil 3.7. Modelin giriş değişkenlerinin üyelik fonksiyonları.

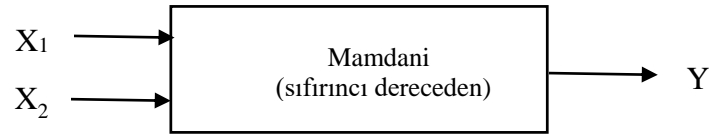


Şekil 3.8. Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.

### 3.4. Örnek-4

Aşağıda tanımlanan tek-ton çıkış ÜFli Mamdani veya eşdeğeri olan sıfırıncı dereceden Sugeno bulanık sistemi T-norm cebirsel çarpım, S-norm max'a göre çalışmaktadır. Çıkış ağırlıklı ortalama ile bulunmaktadır. Sistemin girişleri Şekil 3.9'da verilen üçgen ve yamuk ÜFler ile bulandırılmıştır. Sistemin işlettiği kurallar aşağıda verilmiştir. Kural tablosundaki çıkış değişkeninin tek-ton ÜFleri C1= 0, C2= 2, C3= 4,

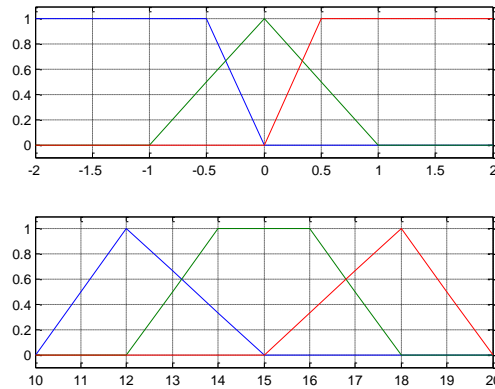
$C4 = 6$ ,  $C5 = 8$ ,  $C6 = 10$  olarak tanımlanmıştır. Sistemin tanımladığı yüzeyi ağırlıklı ortalama yöntemiyle elde edilmiştir.



Tanımlanan sistemin gerçekleyeceği yüzeyi bulmak için aşağıda verilen Matlab kodu yazılmış olup bu program ile Şekil 3.10'da verilen yüzey elde edilmiştir.

Kural tablosu

	X2	N	S	P
X1				
AZ		C1	C2	C4
BRZ		C2	C3	C5
VAR		C3	C4	C6



Şekil 3.9. Modelinin giriş değişkenlerinin üyelik fonksiyonları.

**Matlab kodu**

```
x1=-2:0.5:2;
x2=10:1:20;
mu_x1N= trapmf(x1, [-2 -2 -0.5 0]);
mu_x1S =trimf(x1, [-1 0 1]);
mu_x1P = trapmf(x1, [0 0.5 2 2]);
subplot(2,1,1);
plot(x1, mu_x1N,x1, mu_x1S,x1, mu_x1P);
grid
axis([-2 2 -1 1.1]);
```

```

mu_x2AZ= trimf (x2, [10 12 15]);
mu_x2BRZ =trapmf(x2, [12 14 16 18]);
mu_x2VAR= trimf(x2, [15 18 20])
subplot(2,1,2);
plot (x2,mu_x2AZ,x2,mu_x2BRZ,x2,mu_x2VAR );
grid
axis([10 20 -.1 1.1]);
for i= 1:length(x1)
    for j= 1:length(x2)

        x1_x= x1(i);
        x2_y= x2(j);

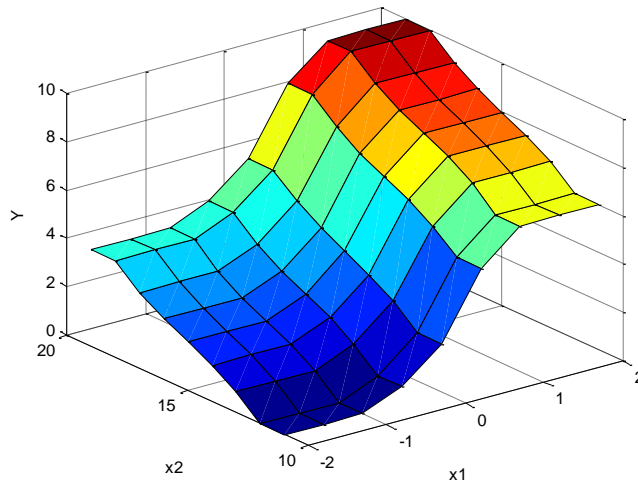
        X1(i,j) = x1_x;
        X2(i,j) = x2_y;
        c1=0;
        c2=2;
        c3=4;
        c4=6;
        c5=8;
        c6=10;

        w1= mu_x1N(i)*mu_x2AZ(j);
        w2= mu_x1S(i)*mu_x2AZ(j);
        w3= mu_x1P(i)*mu_x2AZ(j);
        w4= mu_x1N(i)*mu_x2BRZ(j);
        w5= mu_x1S(i)*mu_x2BRZ(j);
        w6= mu_x1P(i)*mu_x2BRZ(j);
        w7= mu_x1N(i)*mu_x2VAR(j);
        w8= mu_x1S(i)*mu_x2VAR(j);
        w9= mu_x1P(i)*mu_x2VAR(j);

        Y(i,j)= (w1*c1+ w2*c2 +w3*c4 + w4*c2 + w5*c3 + w6*c5 +w7*c3+ w8*c4+w9*c6
)/(w1+w2+w3+w4+w5+w6+w7+w8+w9);

    end
end
figure(2);
surf(X1, X2, Y);
xlabel('x1'), ylabel('x2'), zlabel('Y');
title('sıfır derece sugeno bulanık sistem');

```



Şekil 3.10. Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.

### 3.5. Örnek-5

İki girişli tek çıkışlı Mamdani bulanık sistemi aşağıdaki Mamdani bulanık modelinin bir örneği olarak tanımlanmıştır. 2 girişli 1 çıkışlı Mamdani bulanık modelinin bir örneği aşağıdaki gibi tanımlanabilir.



#### ***Kurallar***

X küçük ve Y küçük ise o halde Z büyük negatif

X küçük ve Y orta ise o halde Z orta negatif

X küçük ve Y büyük ise o halde Z küçük negatif

X büyük ve Y küçük ise o halde Z büyük pozitif

X büyük ve Y orta ise o halde Z orta pozitif

X büyük ve Y büyük ise o halde Z büyük pozitif

#### ***Giriş-Çıkış değişkeni ÜF tanımları***

$\mu_{\text{KÜÇÜK}}(x) = \text{yamuk}(x, -5, -5, -1, 1)$ .

$\mu_{\text{BÜYÜK}}(x) = \text{yamuk}(x, -1, 1, 5, 5)$ .

$\mu_{KÜÇÜK}(y) = \text{yamuk}(y, -5, -5, -3, -1)$ .  
 $\mu_{ORTA}(y) = \text{yamuk}(y, -3, -1, 1, 3)$ .  
 $\mu_{BÜYÜK}(y) = \text{yamuk}(y, 1, 3, 5, 5)$ .  
 $\mu_{BÜYÜK\_NEGATİVE}(z) = \text{yamuk}(z, -5, -5, -3.5, -2.5)$ .  
 $\mu_{ORTA\_ORTA}(z) = \text{yamuk}(z, -5, -4, -2, -1)$ .  
 $\mu_{KÜÇÜK\_NEGATİVE}(z) = \text{yamuk}(z, -3.5, -2.5, -0.5, 0.5)$ .  
 $\mu_{KÜÇÜK\_POZİTİVE}(z) = \text{yamuk}(z, -0.5, 0.5, 2.5, 3.5)$ .  
 $\mu_{ORTA\_POZİTİVE}(z) = \text{yamuk}(z, 1, 2, 4, 5)$ .  
 $\mu_{BÜYÜK\_POZİTİVE}(z) = \text{yamuk}(z, 2.5, 3.5, 5, 5)$ .

Bu örnek için *max S-norm ve cebirsel çarpım T-norm* operetörleri kullanılarak çıkış yüzeyi bulunmuştur.

### **Matlab kodu**

```

x=-5:.5:5;
y=-5:.5:5;
z=-5:.5:5;
mu_xSMALL = trapmf(x, [-5 -5 -1 1]);
mu_xLARGE = trapmf(x, [-1 1 5 5]);
subplot(3,1,1);
plot(x, mu_xSMALL,x, mu_xLARGE);
grid
axis([-5 5 -.1 1.1]);

mu_ySMALL = trapmf(y, [-5 -5 -3 -1]);
mu_yMEDIUM =trapmf(y, [-3 -1 1 3]);
mu_yLARGE = trapmf(y, [1 3 5 5]);
subplot(3,1,2);
plot(y, mu_ySMALL,x, mu_yMEDIUM,y, mu_yLARGE);
grid
axis([-5 5 -.1 1.1]);

mu_zNEGATIVE_LARGE = trapmf(z, [-5 -5 -3.5 -2.5]);
mu_zNEGATIVE_MEDIUM =trapmf(z, [-5 -4 -2 -1]);
mu_zNEGATIVE_SMALL = trapmf(z, [-3.5 -2.5 -0.5 0.5]);
mu_zPOZITIVE_SMALL = trapmf(z, [-0.5 0.5 2.5 3.5]);
mu_zPOZITIVE_MEDIUM =trapmf(z, [1 2 4 5]);
mu_zPOZITIVE_LARGE = trapmf(z, [2.5 3.5 5 5]);
subplot(3,1,3);
plot(z, mu_zNEGATIVE_LARGE,z, mu_zNEGATIVE_MEDIUM,z,
mu_zNEGATIVE_SMALL,z, mu_zPOZITIVE_SMALL,z,
mu_zPOZITIVE_MEDIUM,z,mu_zPOZITIVE_LARGE);
grid
axis([-5 5 -.1 1.1]);

for i=1:length(x)
    for j=1:length(y)

```

```

x_value=x(i);
X(i,j)=x_value;
y_value=y(j);
Y(i,j)=y_value;

w1= mu_xSMALL(i) *mu_ySMALL(j);
w2= mu_xSMALL(i) *mu_yMEDIUM(j);
w3= mu_xSMALL(i) *mu_yLARGE(j);
w4= mu_xLARGE(i) *mu_ySMALL(j);
w5= mu_xLARGE(i) *mu_yMEDIUM(j);
w6= mu_xLARGE(i) *mu_yLARGE(j);

sonuc_mu_zNEGATIVE_LARGE =
w1*ones(size(mu_zNEGATIVE_LARGE)).*mu_zNEGATIVE_LARGE;
sonuc_mu_zNEGATIVE_MEDIUM = w2*mu_zNEGATIVE_MEDIUM;
sonuc_mu_zNEGATIVE_SMALL = w3*mu_zNEGATIVE_SMALL;
sonuc_mu_zPOZITIV_SMALL = w4*mu_zPOZITIV_SMALL;
sonuc_mu_zPOZITIV_MEDIUM = w5*mu_zPOZITIV_MEDIUM;
sonuc_mu_zPOZITIV_LARGE = w6*mu_zPOZITIV_LARGE;

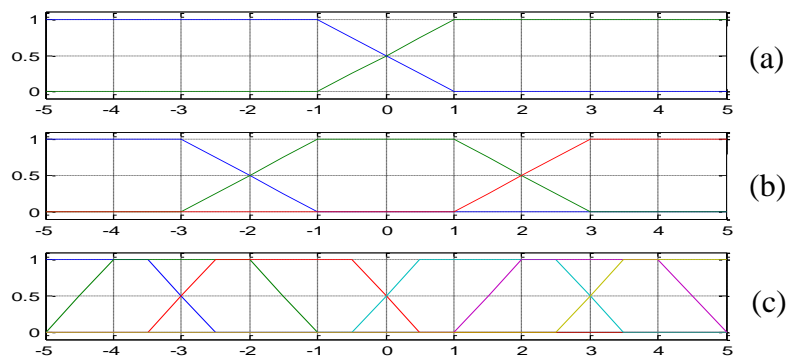
sonuc_zNEGATIVE = max
(sonuc_mu_zNEGATIVE_LARGE,max(sonuc_mu_zNEGATIVE_MEDIUM,
sonuc_mu_zNEGATIVE_SMALL));
sonuc_zPOZITIV = max (sonuc_mu_zPOZITIV_SMALL,max(
sonuc_mu_zPOZITIV_MEDIUM,sonuc_mu_zPOZITIV_LARGE));

Result_MF =max(sonuc_zNEGATIVE,max(sonuc_zPOZITIV));

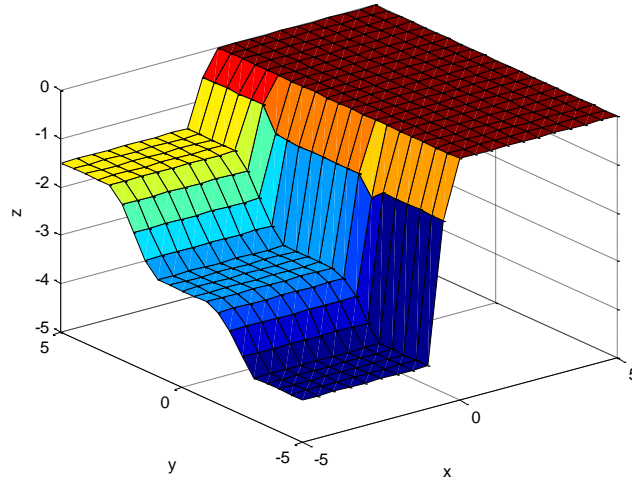
Z(i,j) =sum(Result_MF.*z)/sum(Result_MF);
end

end
figure(2);
surf(X, Y, Z);
xlabel('x'), ylabel('y'), zlabel('z');
title('Mamdani ikinci örneği');

```



Şekil 3.11. Örnek-5 için giriş (a, b) ve çıkış (c) değişkenlerinin üyelik fonksiyonları.



Şekil 3.12. Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.

### 3.6. Örnek-6

İki giriş-tek çıkışlı bir Sugeno bulanık sistem modeli aşağıda verilen ÜFler ve kurallar ile tanımlanmıştır. KİLO={20, 50, 90}, yaş={15, 40, 75} giriş kümelerin olası tüm durumları için bulanık sistem çıkış yüzeydeki noktalara denk düşen FORM değerlerini içeren matrisi, *MİN T-NORM* ve *MAX S-NORM* operatörü kullanarak *ağırlık ortalama* yöntemi ile belirlenmiştir.

KİLO, YAŞ ve ŞEKİL değişkenleri sırasıyla [0,100], [0,80] ve [0,100] aralıklarında tanımlanmıştır.



#### *Giriş-Çıkış değişkeni ÜF tanımları*

$$\mu_{\text{zayıf}}(\text{KİLO}) = \text{YAMUK}(\text{KİLO}, 0, 0, 20, 60)$$

$$\mu_{\text{şişman}}(\text{KİLO}) = \text{YAMUK}(\text{KİLO}, 40, 70, 100, 100)$$

$$\mu_{\text{genç}}(\text{YAŞ}) = \text{YAMUK}(\text{YAŞ}, 0, 0, 20, 60)$$

$$\mu_{\text{yaşlı}}(\text{YAŞ}) = \text{YAMUK}(\text{YAŞ}, 60, 80, 80, 80)$$

### ***Kurallar***

KİLO ZAYİF ve YAŞ GENÇ ise o halde  $\text{ŞEKİL} = 0.5 \cdot \text{KİLO} + 0.5 \cdot \text{YAŞ} + 40$

KİLO ZAYİF ve YAŞ YAŞLI ise o halde  $\text{ŞEKİL} = 0.2 \cdot \text{KİLO} + 0.3 \cdot \text{YAŞ} + 30$

KİLO ŞİŞMAN ve YAŞ GENÇ ise o halde  $\text{ŞEKİL} = 0.1 \cdot \text{KİLO} + 0.5 \cdot \text{YAŞ} + 10$

KİLO ŞİŞMAN ve YAŞ YAŞLI ise o halde  $\text{ŞEKİL} = 0.3 \cdot \text{KİLO} + 0.1 \cdot \text{YAŞ}$

### **Matlab kodu**

```
kilo = 0:10:100;
```

```
yas = 0:10:80;
```

```
mu_kiloZAYIF = trapmf (kilo, [0 0 30 60]);
```

```
mu_kiloSISMAN = trapmf (kilo, [40 70 100 100]);
```

```
subplot(2,1,1)
```

```
plot (kilo, mu_kiloZAYIF, kilo, mu_kiloSISMAN);
```

```
grid
```

```
axis ([0 100 -.1 1.1])
```

```
mu_yasGENC = trapmf (yas, [0 0 20 60]);
```

```
mu_yasYASLI = trapmf (yas, [20 60 80 80]);
```

```
subplot(2,1,2)
```

```
plot (yas, mu_yasGENC, yas, mu_yasYASLI);
```

```
grid
```

```
axis ([0 80 -.1 1.1])
```

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

```
    for j= 1:length(yas)
```

```
        kilo_x= kilo(i);
```

```
        yas_y= yas(j);
```

```
        KILO(i,j)=kilo_x;
```

```
        YAS(i,j)=yas_y;
```

```
        p1= 0.5*kilo_x +0.1*yas_y+40;
```

```
        p2= 0.2*kilo_x + 0.3*yas_y+30;
```

```
        p3= 0.1*kilo_x+ 0.2*yas_y+10;
```

```
        p4= 0.3*kilo_x+ 0.1*yas_y;
```

```
        w1= min(mu_kiloZAYIF(i),mu_yasGENC(j));
```

```
        w2= min(mu_kiloZAYIF(i),mu_yasYASLI(j));
```

```
        w3= min(mu_kiloSISMAN(i),mu_yasGENC(j));
```

```
        w4= min(mu_kiloSISMAN(i),mu_yasYASLI(j));
```

```
        WEIGHT(i,j)= (w1*p1+ w2*p2+w3*p3+w4*p4)/(w1+w2+w3+w4);
```

```
    end
```

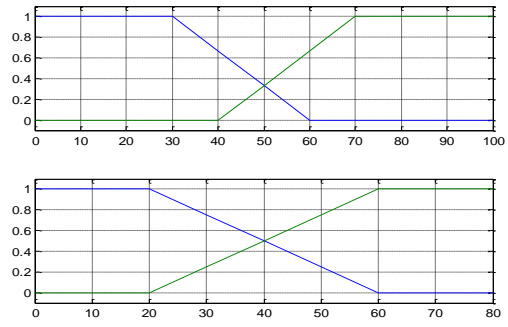
```
end
```

```
figure(2);
```

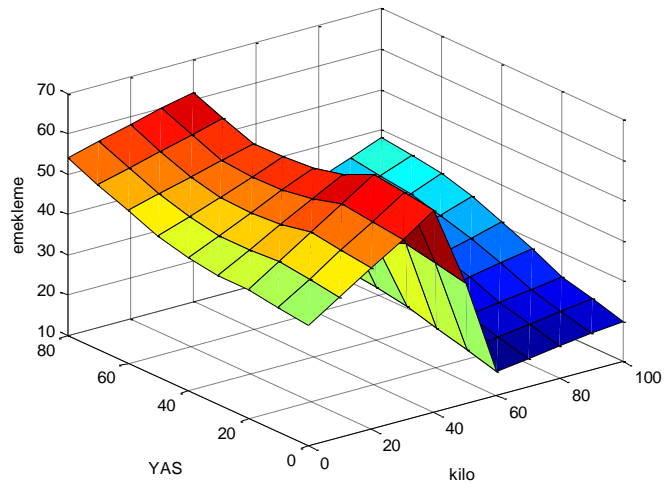
```
surf(KILO, YAS, WEIGHT);
```

```
xlabel('kilo'), ylabel('YAS'), zlabel('emekleme');
```

```
title('Bulanık sistemin ağırlık yöntemi');
```



Şekil 3.12. Modelinin giriş değişkenlerinin üyelik fonksiyonları.



Şekil 3.13. Tanımlanan bulanık sistemin gerçeklediği giriş-çıkış yüzeyi.

#### 4. VERİYE DAYALI BULANIK SİSTEM MODELLEME

Gerçek hayattaki problemlerin klasik mantığa dayalı yani var-yok veya '1' , '0' mantığı ile çözmek çoğu zaman problemimizi çözmez. Bundan dolayı bulanık mantık sistemlerine ihtiyaç duyulmaktadır. Bunun için sistemlerin matematiksel modelleri birçok bilim dalında önemli hale gelmiştir. Fakat bu sistem modellerinin problemi çözebilmesi için olabildiğince gerçeğe uygun ve yakın olması gerekir. Yoksa daha sonra önemli sorunlar ortaya çıkabilir. Tasarlanacak model ne çok basit ne de çok karmaşık olmalıdır. İnsanın düşüncüsü ele alındığında birçok model arayışı ortaya çıkmıştır. Bulanık model zaten bu yöntemlerden biridir. Veriye dayalı bir sistemde temel olarak eğitim ve test aşamaları vardır. Yani sistemimizin eğitimi için bir data eğitim setine ve eğitimimizi tamamladıktan sonra ağın iyi eğitilip eğitilmediğini kontrol etmek için de bir test verisine ihtiyaç vardır. Ağın eğitim aşamasında giriş parametreleri ve ağırlıkları sisteme girdi olarak verilir ve sonuçta mantıklı doğruya yakın en az hataya sahip bir sonuç beklenir. Sonuçta çıkan değer ağın başarısını test eder.

Bu bölümde veriye dayalı modelleme için kullanılacak yöntem (Babuska, 1998) açıklanacaktır. Modelleme için kullanılmak üzere elimizde  $N$  adet giriş-çıkış çiftleri  $(x_i, y_i) \mid i=1,2,\dots,N$  olduğunu düşünelim. Burada,  $x_i \in R^p$  giriş vektörü ve  $y_i$  skaler çıkış olarak tanımlıdır.  $X \in R^{N \times p}$  satırlarında  $x_k^T$  giriş vektörlerini barındıran matris,  $\mathcal{Y} \in R^N$  modellemede kullanılacak istenen çıkış  $y_k$  örneklerini içeren bir vektörü gösterebilir.

$$X = [x_1, \dots, x_N]^T, \quad \mathcal{Y} = [y_1, \dots, y_N]^T \quad (4.1)$$

Bu çalışmada, modelleme aracı olarak kullanılacak TS yapısının sonuç (kural) parametrelerinin en küçük kareler yöntemiyle kestirimi kullanılmıştır. Bu yaklaşımda TS bulanık modelin şart parametreleri, girişlere ait ÜFlerin ilgili değişken değişim aralığında homojen bir şekilde yerleştirilmesiyle belirlenmektedir. Sıfırıncı dereceden (tek ton) ve birinci dereceden TS bulanık modellerinin sırasıyla eşitlik (4.2) ve eşitlik (4.3)'de verilen formüllerinden de görüleceği üzere  $a_i, b_i$  sonuç/kural parametrelerinin doğrusal bir ifadesi olduğu anlaşılır.

$$y = \frac{\sum_{i=1}^K \beta_i b_i}{\sum_{i=1}^K \beta_i} \quad (4.2)$$

$$y = \frac{\sum_{i=1}^K \beta_i y_i}{\sum_{i=1}^K \beta_i} = \frac{\sum_{i=1}^K \beta_i (a_i^T x + b_i)}{\sum_{i=1}^K \beta_i} \quad (4.3)$$

Bu nedenle bu parametreler mevcut veriden en küçük kareler tekniğiyle kestirilebilir.  $\Gamma_i \in R^{N \times N}$ ,  $k$  diyagonal elemanı olarak eşitlik (4.4)'de tanımlanan  $k$  inci giriş için normalleştirilmiş üyelik derecesini ( $\gamma_i(x_k)$ ) içeren diyagonal matrisini gösterebilir.  $X$  giriş matrisine birim kolon ekleyerek genişletilmiş  $X_e = [X, 1]$  matrisi oluşturulur. Bu işlemin sonrasında, eşitlik (4.5)'te tanımlanan  $X' \in R^{N \times KN}$  matrisi elde edilir.

$$\gamma_i(x) = \beta_i(x) / \sum_{j=1}^K \beta_j(x) \quad (4.4)$$

$$X' = [\Gamma_1 \times X_e \quad \Gamma_2 \times X_e \quad \dots \quad \Gamma_k \times X_e] \quad (4.5)$$

TSK modelinin sonuç/kural parametreleri  $(a_i, b_i)$  tek parametre vektöründe ( $\theta \in R^{K(p+1)}$ ) eşitlik (4.6)'da verildiği gibi toplanmıştır.

$$\theta = [a_1^T b_1, a_2^T b_2, \dots, a_k^T b_k]^T \quad (4.6)$$

$X$ ,  $\mathcal{Y}$  verileri verildiğinde, (2) ve (3) denklemleri  $\mathcal{Y} = X'\theta + \varepsilon$  matris formunda yazılabilir. Bilinen doğrusal cebir işlemleriyle  $\theta$  parametre vektörü eşitlik (4.7) ile çözebilir.

$$\theta = [(X')^T X']^{-1} (X')^T \mathcal{Y} \quad (4.7)$$

Eşitlik (7) en az tahmin hatası veren en uygun (optimal) en küçük kareler çözümüdür. Bununla birlikte bu çözüm, yerel modellerin parametreleri olarak bulanık sistem sonuç parametrelerinin tahminlerini sapmalı olarak tutabilir. Yerel model parametrelerinin doğru bir tahmini isteniyorsa, kural başına uygulanan eşitlik (4.8)'deki ağırlıklı en küçük kareler yaklaşımı kullanılabilir.

$$[a_i^T, b_i]^T = [X_e^T \Gamma_i X_e]^{-1} X_e^T \Gamma_i \mathcal{Y} \quad (4.8)$$

Bu durumda, her bir kuralın parametreleri birbirinden bağımsız olarak tahmin edilir ve böylece kuralların etkileşimleri tarafından herhangi bir sapma belirmez. Eşitlik (7) ve (8),  $K$  adet kurala sahip TSK bulanık modeli için doğrudan uygulanabilir.

Yukarıda verilen yöntemde eşitlik (5) ve (8)'de kullanılan  $\Gamma_i \in R^{N \times N}$  diyagonal bir matris olup eşitlik (4.9)'da verildiği gibi bir yapıya sahiptir (Babuska, 1998).

$$\Gamma_i = \begin{bmatrix} \gamma_{i(x_1)} & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \gamma_{i(x_N)} \end{bmatrix}_{N \times N} \quad (4.9)$$

#### 4.1. Veriye Dayalı Modelleme Örnekleri

Bu alt bölümde, yukarıda tanımlanan veriye dayalı bulanık sistem modelleme yönteminin etkinliğini göstermek amacıyla beş adet örnek dinamik sistemin Sugeno bulanık modeli ile modellenmesi gösterilmiştir. Her bir örneğin tanımı ve elde edilen bulanık sistem ve başarımları aşağıda kısaca verilmiştir.

##### 4.1.1. Dinamik sistem modelleme örnek 1

Dinamik sistemin matematik fonksiyonu (4.10) de (Narendra ve Parthasarathy, 1990) verilmiştir. Bu sistem için giriş olarak (4.11) dizisi kullanarak eğitim verisi bulunmuştur. Bu sistemin başarısını kontrol etmek için (4.12)'teki giriş dizisi kullanıldı ve test verisi bulundu.

$$y(k) = \frac{y(k-1).y(k-2).(y(k-1)+2.5)}{1+y^2(k-1)+y^2(k-2)} + u(k) \quad (4.10)$$

$$u(k) = \cos \frac{2\pi k}{100} \quad (4.11)$$

$$u(k) = \sin \frac{2\pi k}{100} \quad (4.12)$$

Aşağıda bu dinamik sistemin TSK bulanık modeli ilçe modellenmesinde kullanılan Matlab kodu verilmiştir. Sistem girişi için  $y(k-1)$ ,  $y(k-2)$ ,  $u(k)$  girişlerini  $x_1, x_2, x_3$  olarak alındı ve ağırlık girişine uygulandı. Sistem üç girişli bir çıkışlı olacak şekilde tasarlanmıştır. *PIveri.mat* veri kümesinden alınan veriler  $x_1, x_2, x_3$  e eğitim için verildi. Daha sonra *PIveritest.mat* veri kümesinden veriler ile eğittiğimiz ağırlık başarısını test ederek ağırlık ne kadar doğruya yakın eğitildiği test edildi. Bu örnek için açıklayıcı bazı şekiller Şekil 4.1- 4.3'de verilmiştir.

```

load P1veri.mat
x1=veri(:,3);% 1st input for training
x2=veri(:,1);% 2nd input for training
x3=veri(:,2);% 3th input for training
x=[x1 x2 x3];
y=veri(:,4);%output for training

%% prepare the inputs/output for testing
load P1veritest.mat
x1t=veri(:,3);% 1st input for testing
x2t=veri(:,1);% 2nd input for testing
x3t=veri(:,2);% 3th input for testing
xt=[x1t x2t x3t]; %input for testing
yt=veri(:,4); %output for testig
%% Ploting
subplot(2,2,1); plot (x1); xlabel('k'); ylabel('u(k)'); title('for training');
subplot(2,2,3); plot (y); xlabel('k'); ylabel('y(k)');

subplot(2,2,2); plot (x1t); xlabel('k'); ylabel('u(k)'); title('for testing');
subplot(2,2,4); plot (yt); xlabel('k'); ylabel('y(k)');

teta=[];%parameter array/matrix

Xe= [x ones(size(x1))]; %%% extended input

%% membership funtions deffination
%for the first input ,u(k)
x1_array=-1:0.1:1;
A1 = gaussmf(x1_array, [0.4 -0.5]);
A2 = gaussmf(x1_array, [0.4 0.5]);
figure(2)
subplot(3,1,1); plot(x1_array, A1, x1_array, A2);
xlabel('u(k)');
title('MFs for u(k) input');

%for the second input ,y(k-1)
x2_array=-2:0.1:5;
B1 = gaussmf(x2_array, [0.8 -0.5]);
B2 = gaussmf(x2_array, [1.5 3]);
subplot(3,1,2); plot(x2_array, B1, x2_array, B2);
xlabel('y(k-1)');
title('MFs for y(k-1) input');

%for the third input ,y(k-2)
x3_array=-2:0.1:5;
C1 = gaussmf(x3_array, [0.8 -0.5]);
C2 = gaussmf(x3_array, [1.5 3]);
subplot(3,1,3); plot(x3_array, C1, x3_array, C2);
xlabel('y(k-2)');
title('MFs for y(k-2) input');

%% %% TRAINING TRANSACTIONS
%% %% computing gamma values
for n= 1:length(x1)

```

```

%finding firing strength (activity degree) of each rule
w1=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [0.8 -0.5])*gaussmf(x3(n), [0.8 -0.5]);
w2=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [0.8 -0.5])*gaussmf(x3(n), [1.5 3]); %A1 B1 C2
w3=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [1.5 3])*gaussmf(x3(n), [0.8 -0.5]); %A1 B2 C1
w4=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [1.5 3])*gaussmf(x3(n), [1.5 3]); %A1 B2 C2
w5=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [0.8 -0.5])*gaussmf(x3(n), [0.8 -0.5]); %A2 B1
w6=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [0.8 -0.5])*gaussmf(x3(n), [1.5 3]); %A2 B1 C2
w7=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [1.5 3])*gaussmf(x3(n), [0.8 -0.5]); %A2 B2 C1
w8=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [1.5 3])*gaussmf(x3(n), [1.5 3]); %A2 B2 C2
%finding normalized firing strength (activity degree) of each rule
gamma1(n)= w1/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma2(n)= w2/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma3(n)= w3/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma4(n)= w4/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma5(n)= w5/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma6(n)= w6/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma7(n)= w7/(w1+w2+w3+w4+w5+w6+w7+w8);
gamma8(n)= w8/(w1+w2+w3+w4+w5+w6+w7+w8);

end

Gama1= diag(gamma1);
Gama2= diag(gamma2);
Gama3= diag(gamma3);
Gama4= diag(gamma4);
Gama5= diag(gamma5);
Gama6= diag(gamma6);
Gama7= diag(gamma7);
Gama8= diag(gamma8);
%% weighted least-squares approach appliedper rule
teta=[inv(Xe'*Gama1*Xe)*Xe'*Gama1*y inv(Xe'*Gama2*Xe)*Xe'*Gama2*y
inv(Xe'*Gama3*Xe)*Xe'*Gama3*y inv(Xe'*Gama4*Xe)*Xe'*Gama4*y...
inv(Xe'*Gama5*Xe)*Xe'*Gama5*y inv(Xe'*Gama6*Xe)*Xe'*Gama6*y
inv(Xe'*Gama7*Xe)*Xe'*Gama7*y inv(Xe'*Gama8*Xe)*Xe'*Gama8*y];

%% END OF TRAINING
%% PERFORMANCE FOR TRAINING DATA with founded rule parameter

for i= 1:length(x1)

p1= teta(1,1)*x1(i)+teta(2,1)*x2(i)+teta(3,1)*x3(i)+ teta(4,1);
p2= teta(1,2)*x1(i)+teta(2,2)*x2(i)+teta(3,2)*x3(i)+ teta(4,2);
p3= teta(1,3)*x1(i)+teta(2,3)*x2(i)+teta(3,3)*x3(i)+ teta(4,3);
p4= teta(1,4)*x1(i)+teta(2,4)*x2(i)+teta(3,4)*x3(i)+ teta(4,4);
p5= teta(1,5)*x1(i)+teta(2,5)*x2(i)+teta(3,5)*x3(i)+ teta(4,5);
p6= teta(1,6)*x1(i)+teta(2,6)*x2(i)+teta(3,6)*x3(i)+ teta(4,6);
p7= teta(1,7)*x1(i)+teta(2,7)*x2(i)+teta(3,7)*x3(i)+ teta(4,7);
p8= teta(1,8)*x1(i)+teta(2,8)*x2(i)+teta(3,8)*x3(i)+ teta(4,8);

w1=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i),[0.8 -0.5])*gaussmf(x3(i), [0.8 -0.5]);%A1 B1 C1
w2=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i),[0.8 -0.5])*gaussmf(x3(i), [1.5 3]); %A1 B1 C2
w3=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i),[1.5 3])*gaussmf(x3(i), [0.8 -0.5]); %A1 B2 C1
w4=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i),[1.5 3])*gaussmf(x3(i), [1.5 3]); %A1 B2 C2
w5=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i),[0.8 -0.5])*gaussmf(x3(i), [0.8 -0.5]); %A2 B1 C1
w6=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i),[0.8 -0.5])*gaussmf(x3(i), [1.5 3]); %A2 B1 C2
w7=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i),[1.5 3])*gaussmf(x3(i), [0.8 -0.5]);%A2 B2 C1
w8=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i),[1.5 3])*gaussmf(x3(i), [1.5 3]); %A2 B2 C2

```

```

train_result(i)=(w1*p1+
w2*p2+w3*p3+w4*p4+w5*p5+w6*p6+w7*p7+w8*p8)/(w1+w2+w3+w4+w5+w6+w7+w8);
    end

    figure(3);
    %subplot(2,1,1),
    plot(train_result,'b. ');
    title('training performance of fuzzy system');
    hold on
    plot(y,'k'); hold off
    xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%   subplot(2,1,2) ; plot(x1,'k');
%   xlabel('k'), ylabel('u(k)');

%% END of PERFORMANCE FOR TRAINING DATA with founded rule parameter
%% TESTING TRANSACTIONS
%% PERFORMANCE FOR TESTING DATA with founded rule parameter

    for i= 1:length(x1t)

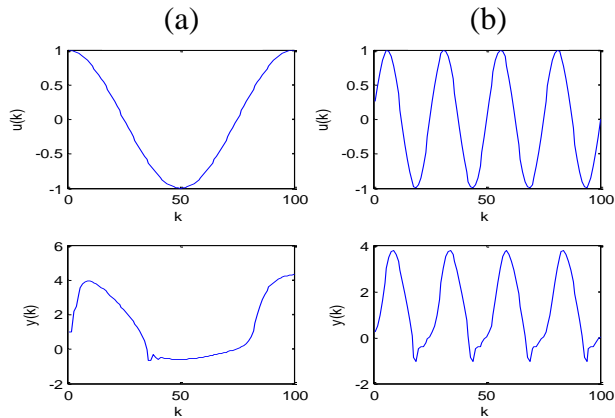
p1= teta(1,1)*x1t(i)+teta(2,1)*x2t(i)+teta(3,1)*x3t(i)+ teta(4,1);
p2= teta(1,2)*x1t(i)+teta(2,2)*x2t(i)+teta(3,2)*x3t(i)+ teta(4,2);
p3= teta(1,3)*x1t(i)+teta(2,3)*x2t(i)+teta(3,3)*x3t(i)+ teta(4,3);
p4= teta(1,4)*x1t(i)+teta(2,4)*x2t(i)+teta(3,4)*x3t(i)+ teta(4,4);
p5= teta(1,5)*x1t(i)+teta(2,5)*x2t(i)+teta(3,5)*x3t(i)+ teta(4,5);
p6= teta(1,6)*x1t(i)+teta(2,6)*x2t(i)+teta(3,6)*x3t(i)+ teta(4,6);
p7= teta(1,7)*x1t(i)+teta(2,7)*x2t(i)+teta(3,7)*x3t(i)+ teta(4,7);
p8= teta(1,8)*x1t(i)+teta(2,8)*x2t(i)+teta(3,8)*x3t(i)+ teta(4,8);

w1=gaussmf(x1t(i), [0.4 -0.5])*gaussmf(x2t(i),[0.8 -0.5])*gaussmf(x3t(i), [0.8 -0.5]);%A1 B1 C
w2=gaussmf(x1t(i), [0.4 -0.5])*gaussmf(x2t(i),[0.8 -0.5])*gaussmf(x3t(i), [1.5 3]); %A1 B1 C2
w3=gaussmf(x1t(i), [0.4 -0.5])*gaussmf(x2t(i),[1.5 3])*gaussmf(x3t(i), [0.8 -0.5]); %A1 B2 C1
w4=gaussmf(x1t(i), [0.4 -0.5])*gaussmf(x2t(i),[1.5 3])*gaussmf(x3t(i), [1.5 3]); %A1 B2 C2
w5=gaussmf(x1t(i), [0.4 0.5])*gaussmf(x2t(i),[0.8 -0.5])*gaussmf(x3t(i), [0.8 -0.5]); %A2 B1 C
w6=gaussmf(x1t(i), [0.4 0.5])*gaussmf(x2t(i),[0.8 -0.5])*gaussmf(x3t(i), [1.5 3]); %A2 B1 C2
w7=gaussmf(x1t(i), [0.4 0.5])*gaussmf(x2t(i),[1.5 3])*gaussmf(x3t(i), [0.8 -0.5]);%A2 B2 C1
w8=gaussmf(x1t(i), [0.4 0.5])*gaussmf(x2t(i),[1.5 3])*gaussmf(x3t(i), [1.5 3]); %A2 B2 C2

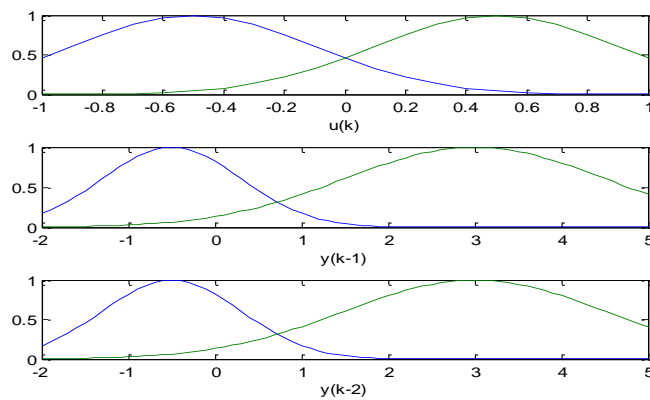
test_result(i)=(w1*p1+
w2*p2+w3*p3+w4*p4+w5*p5+w6*p6+w7*p7+w8*p8)/(w1+w2+w3+w4+w5+w6+w7+w8);
    end

    figure(4);
    %subplot(2,1,1),
    plot(test_result,'r* ');
    title('testing performance of fuzzy system');
    hold on
    plot(yt,'k'); hold off
    xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%% END of PERFORMANCE FOR TESTING DATA with founded rule parameter

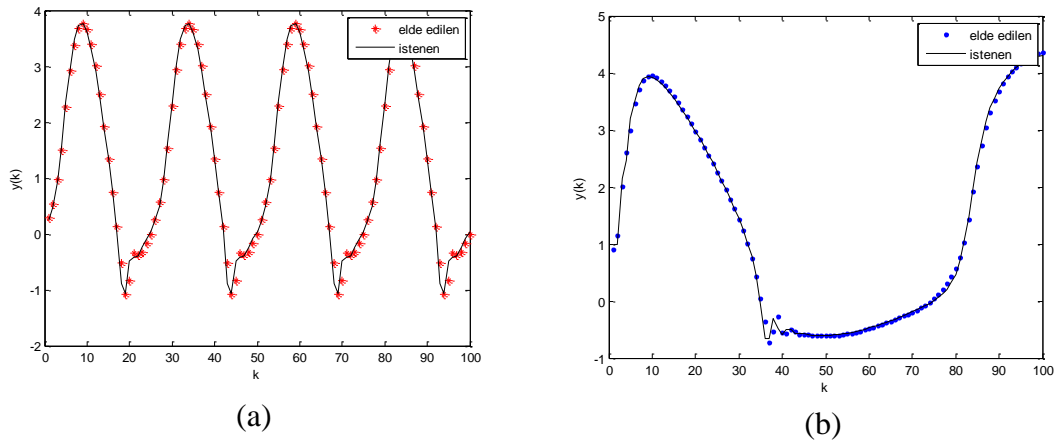
```



Şekil 4.1. ÖDS 1 için eğitim (a) ve test (b) giriş  $u(k)$  ve çıkış  $y(k)$  dizileri.



Şekil 4.2. ÖDS 1'in modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.



Şekil 4.3. Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.

#### 4.1.2. Dinamik sistem modelleme örnek 2

Dinamik sistemin matematiksel tanımı (4.13)'de (Narendra ve Parthasarathy, 1990) verilmiştir. Bu sistem için giriş olarak (4.14) ile elde edilen dizi kullanarak eğitim verisi oluşturulmuştur. Bu sistemin başarısını kontrol etmek için (4.15)'teki giriş dizisi

kullanıldı ve test verisi elde edildi. Bu sistemin  $u(k)$ ,  $y(k)$ ,  $y(k-1)$  girişleri için ikişer üyelik fonksiyonu ile sekiz kural ve toplamda 32 parameterli bir sistem tasarlanmıştır.

$$y(k + 1) = \frac{y(k)}{1+y^2(k)} + u^3(k) \quad (4.13)$$

$$u(k) = \cos \frac{2\pi k}{100} \quad (4.14)$$

$$u(k) = \sin \frac{2\pi k}{100} \quad (4.15)$$

```

%% prepare the inputs/output for training
load P2veri.mat
x1=veri(:,3);% 1st input for training
x2=veri(:,2);% 2nd input for training
x3=veri(:,1);% 3th input for training
x=[x1 x2 x3];
y=veri(:,4);%output for training

%% prepare the inputs/output for testing
load P2veritest.mat
x1t=veri(:,3);% 1st input for testing
x2t=veri(:,2);% 2nd input for testing
x3t=veri(:,1);% 3th input for testing
xt=[x1t x2t x3t]; %input for testing
yt=veri(:,4); %output for testig
%% Ploting
subplot(2,2,1); plot (x1); xlabel('k'); ylabel('u(k)');
subplot(2,2,3); plot (y); xlabel('k'); ylabel('y(k)');

subplot(2,2,2); plot (x1t); xlabel('k'); ylabel('u(k)');
subplot(2,2,4); plot (yt); xlabel('k'); ylabel('y(k)');

teta=[];%parameter array/matrix

Xe= [x ones(size(x1))]; %%% extended input
%% membership funtions deffination
%for the first input ,u(k)
x1_array=-1:0.1:1;
A1 = gaussmf(x1_array, [0.4 -0.5]);
A2 = gaussmf(x1_array, [0.4 0.5]);
figure(2)
subplot(3,1,1); plot(x1_array, A1, x1_array, A2);
xlabel('u(k)');

%for the second input ,y(k-1)
x2_array=-2:0.1:2;
B1 = gaussmf(x2_array, [0.8 -1]);
B2 = gaussmf(x2_array, [0.8 1]);
subplot(3,1,2); plot(x2_array, B1, x2_array, B2);
xlabel('y(k)');

```

```

%for the third input ,y(k-2)
x3_array=-2:0.1:2;
    C1 = gaussmf(x3_array, [0.8 -1]);
    C2 = gaussmf(x3_array, [0.8 1]);
    subplot(3,1,3); plot(x3_array, C1, x3_array, C2);
    xlabel('y(k-1)');

%% TRAINING TRANSACTIONS
% %% computing gamma values
for n= 1:length(x1)
    %finding firing strength (activity degree) of each rule
    w1=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [0.8 -1])*gaussmf(x3(n), [0.8 -1]);
    w2=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [0.8 -1])*gaussmf(x3(n), [0.8 1]); %A1 B1 C2
    w3=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [0.8 1])*gaussmf(x3(n), [0.8 -1]); %A1 B2 C1
    w4=gaussmf(x1(n), [0.4 -0.5])*gaussmf(x2(n), [0.8 1])*gaussmf(x3(n), [0.8 1]); %A1 B2 C2
    w5=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [0.8 -1])*gaussmf(x3(n), [0.8 -1]); %A2 B1 C1
    w6=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [0.8 -1])*gaussmf(x3(n), [0.8 1]); %A2 B1 C2
    w7=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [0.8 1])*gaussmf(x3(n), [0.8 -1]); %A2 B2 C1
    w8=gaussmf(x1(n), [0.4 0.5])*gaussmf(x2(n), [0.8 1])*gaussmf(x3(n), [0.8 1]); %A2 B2 C2
    %finding normalized firing strength (activity degree) of each rule
    gamma1(n)= w1/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma2(n)= w2/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma3(n)= w3/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma4(n)= w4/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma5(n)= w5/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma6(n)= w6/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma7(n)= w7/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma8(n)= w8/(w1+w2+w3+w4+w5+w6+w7+w8);

end

Gama1= diag(gamma1);
Gama2= diag(gamma2);
Gama3= diag(gamma3);
Gama4= diag(gamma4);
Gama5= diag(gamma5);
Gama6= diag(gamma6);
Gama7= diag(gamma7);
Gama8= diag(gamma8);

%% weighted least-squares approach appliedper rule
teta=[inv(Xe'*Gama1*Xe)*Xe'*Gama1*y inv(Xe'*Gama2*Xe)*Xe'*Gama2*y
inv(Xe'*Gama3*Xe)*Xe'*Gama3*y inv(Xe'*Gama4*Xe)*Xe'*Gama4*y...
    inv(Xe'*Gama5*Xe)*Xe'*Gama5*y inv(Xe'*Gama6*Xe)*Xe'*Gama6*y
inv(Xe'*Gama7*Xe)*Xe'*Gama7*y inv(Xe'*Gama8*Xe)*Xe'*Gama8*y];

%% END OF TRAINING

%% PERFORMANCE FOR TRAINING DATA with founded rule parameter

for i= 1:length(x1)

p1= teta(1,1)*x1(i)+teta(2,1)*x2(i)+teta(3,1)*x3(i)+ teta(4,1);
p2= teta(1,2)*x1(i)+teta(2,2)*x2(i)+teta(3,2)*x3(i)+ teta(4,2);
p3= teta(1,3)*x1(i)+teta(2,3)*x2(i)+teta(3,3)*x3(i)+ teta(4,3);
p4= teta(1,4)*x1(i)+teta(2,4)*x2(i)+teta(3,4)*x3(i)+ teta(4,4);

```

```

p5= teta(1,5)*x1(i)+teta(2,5)*x2(i)+teta(3,5)*x3(i)+ teta(4,5);
p6= teta(1,6)*x1(i)+teta(2,6)*x2(i)+teta(3,6)*x3(i)+ teta(4,6);
p7= teta(1,7)*x1(i)+teta(2,7)*x2(i)+teta(3,7)*x3(i)+ teta(4,7);
p8= teta(1,8)*x1(i)+teta(2,8)*x2(i)+teta(3,8)*x3(i)+ teta(4,8);

w1=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i),[0.8 -1])*gaussmf(x3(i), [0.8 -1]);%A1 B1 C1
w2=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i),[0.8 -1])*gaussmf(x3(i), [0.8 1]); %A1 B1 C2
w3=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i), [0.8 1])*gaussmf(x3(i), [0.8 -1]); %A1 B2 C1
w4=gaussmf(x1(i), [0.4 -0.5])*gaussmf(x2(i), [0.8 1])*gaussmf(x3(i), [0.8 1]); %A1 B2 C2
w5=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i), [0.8 -1])*gaussmf(x3(i), [0.8 -1]); %A2 B1 C1
w6=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i), [0.8 -1])*gaussmf(x3(i), [0.8 1]); %A2 B1 C2
w7=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i), [0.8 1])*gaussmf(x3(i), [0.8 -1]);%A2 B2 C1
w8=gaussmf(x1(i), [0.4 0.5])*gaussmf(x2(i), [0.8 1])*gaussmf(x3(i), [0.8 1]); %A2 B2 C2

    train_result(i)=(w1*p1+
w2*p2+w3*p3+w4*p4+w5*p5+w6*p6+w7*p7+w8*p8)/(w1+w2+w3+w4+w5+w6+w7+w8);
    end

    figure(3);
    plot(train_result,'g');
    hold on
    plot(y,'k'); hold off
    xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')

%% END of PERFORMANCE FOR TRAINING DATA with founded rule parameter
%% TESTING TRANSACTIONS
%% PERFORMANCE FOR TESTING DATA with founded rule parameter

    for i= 1:length(x1t)

p1= teta(1,1)*x1t(i)+teta(2,1)*x2t(i)+teta(3,1)*x3t(i)+ teta(4,1);
p2= teta(1,2)*x1t(i)+teta(2,2)*x2t(i)+teta(3,2)*x3t(i)+ teta(4,2);
p3= teta(1,3)*x1t(i)+teta(2,3)*x2t(i)+teta(3,3)*x3t(i)+ teta(4,3);
p4= teta(1,4)*x1t(i)+teta(2,4)*x2t(i)+teta(3,4)*x3t(i)+ teta(4,4);
p5= teta(1,5)*x1t(i)+teta(2,5)*x2t(i)+teta(3,5)*x3t(i)+ teta(4,5);
p6= teta(1,6)*x1t(i)+teta(2,6)*x2t(i)+teta(3,6)*x3t(i)+ teta(4,6);
p7= teta(1,7)*x1t(i)+teta(2,7)*x2t(i)+teta(3,7)*x3t(i)+ teta(4,7);
p8= teta(1,8)*x1t(i)+teta(2,8)*x2t(i)+teta(3,8)*x3t(i)+ teta(4,8);

w1=gaussmf(x1t(i),[0.4 -0.5])*gaussmf(x2t(i),[0.8 -1])*gaussmf(x3t(i), [0.8 -1]);%A1 B1 C1
w2=gaussmf(x1t(i),[0.4 -0.5])*gaussmf(x2t(i),[0.8 -1])*gaussmf(x3t(i), [0.8 1]); %A1 B1 C2
w3=gaussmf(x1t(i),[0.4 -0.5])*gaussmf(x2t(i), [0.8 1])*gaussmf(x3t(i), [0.8 -1]); %A1 B2 C1
w4=gaussmf(x1t(i),[0.4 -0.5])*gaussmf(x2t(i), [0.8 1])*gaussmf(x3t(i), [0.8 1]); %A1 B2 C2
w5=gaussmf(x1t(i),[0.4 0.5])*gaussmf(x2t(i), [0.8 -1])*gaussmf(x3t(i), [0.8 -1]); %A2 B1 C1
w6=gaussmf(x1t(i),[0.4 0.5])*gaussmf(x2t(i), [0.8 -1])*gaussmf(x3t(i), [0.8 1]); %A2 B1 C2
w7=gaussmf(x1t(i), [0.4 0.5])*gaussmf(x2t(i), [0.8 1])*gaussmf(x3t(i), [0.8 -1]);%A2 B2 C1
w8=gaussmf(x1t(i), [0.4 0.5])*gaussmf(x2t(i), [0.8 1])*gaussmf(x3t(i), [0.8 1]); %A2 B2 C2

    test_result(i)=(w1*p1+
w2*p2+w3*p3+w4*p4+w5*p5+w6*p6+w7*p7+w8*p8)/(w1+w2+w3+w4+w5+w6+w7+w8);
    end

    figure(4);
    plot(test_result,'r-');
    hold on

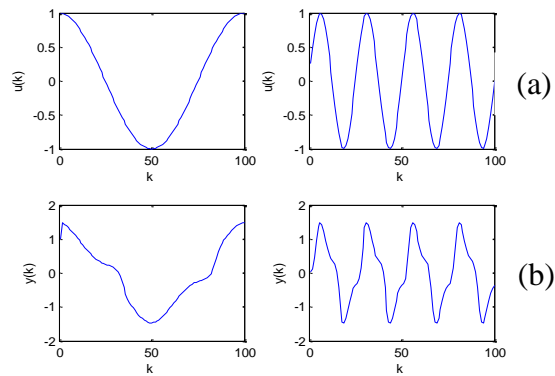
```

```

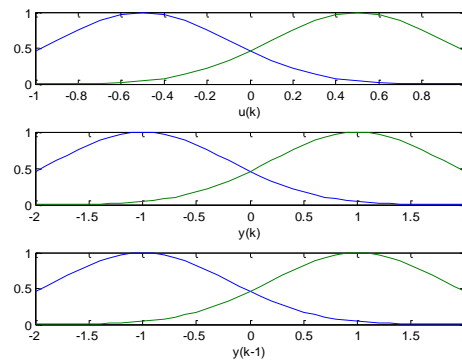
plot(yt,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%% END of PERFORMANCE FOR TESTING DATA with founded rule parameter

```

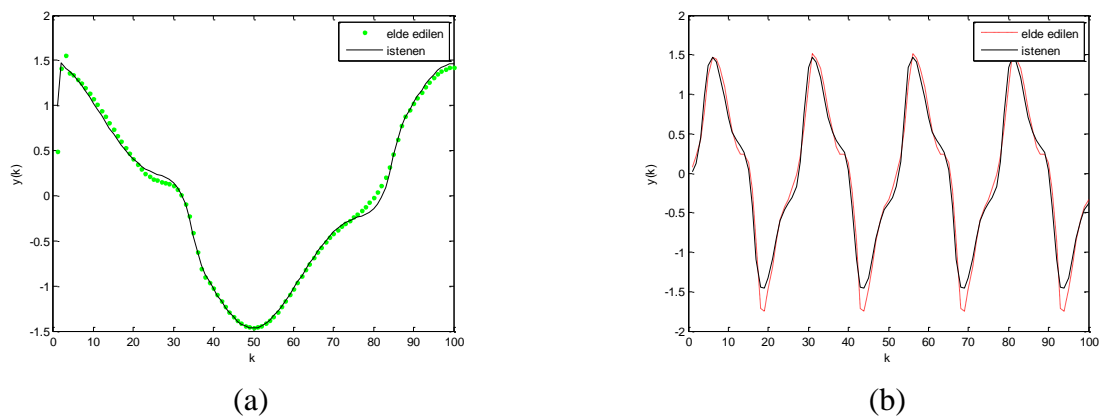
Bulank sistemimiz ile ilgili grafiksel bilgiler Şekil 4.4-4.6’da verilmiştir.



Şekil 4.4. ÖDS 2 için eğitim (a) ve test (b) giriş  $u(k)$  ve çıkış  $y(k)$  dizileri.



Şekil 4.5. ÖDS 2'nin modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.



Şekil 4.6. Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.

### 4.1.3 Dinamik sistem modelleme örnek 3

Dinamik sistemin tanımı (4.16) de (Babuska) verilmiştir. Bu sistem eğitimi ve testinde kullanmak üzere  $[1 \ -1]$  aralığında rasgele genlikli darbelerden oluşan  $u(k)$  girişleri ile eğitim ve test verisi hazırlanmıştır. Giriş darbelerinin süresi 10 örnekleme periyodu olacak şekilde ayarlanmıştır. Sistemde  $u(k)$  ve  $y(k)$  girişler olarak alınmıştır.

$$y(k + 1) = y(k) + u(k)e^{-3|y(k)|} \quad (4.16)$$

```

%% prepare the inputs/output for training
load P3veri.mat

x1= veri(:,2); % first input for training
x2= veri(:,1); %second input for trainig
x=[x1 x2];
y= veri(:,3); %output for training
%% prepare the input/output for testing
load P3veritest.mat
x1t= veri (:,2); %first input for testing
x2t= veri (:,1); %second input for testing
xt=[x1t x2t]; % input for testig
yt= veri(:,3); %% output for testing

%% Ploting
subplot(2,2,1); plot (x1); xlabel('k'); ylabel('u(k)'); title('for training');
subplot(2,2,3); plot (y); xlabel('k'); ylabel('y(k)');

subplot(2,2,2); plot (x1t); xlabel('k'); ylabel('u(k)'); title('for testing');
subplot(2,2,4); plot (yt); xlabel('k'); ylabel('y(k)');

teta=[];%parameter array/matrix

Xe= [x ones(size(x1))]; %%% extended input

%% membership funtions defination
%% for the first input u(k)
x1_array= -2:0.1:2;
A1= gaussmf(x1_array, [0.5 -0.7]);
A2= gaussmf(x1_array, [0.5 0.7]);
figure (2)
subplot(2,1,1); plot(x1_array, A1, x1_array, A2);
xlabel('u(k)');

x2_array= -3:0.1:3;
B1= gaussmf(x2_array, [0.8 -1]);
B2= gaussmf(x2_array, [0.8 1]);
subplot (2,1,2); plot (x2_array, B1, x2_array,B2);
xlabel('y(k)');

%% TRAINING TRANSACTIONS
%% %% computing gamma values
for n= 1:length(x1)
    %finding firing strength (activity degree) of each rule

```

```

w1=gaussmf(x1(n), [0.5 -0.7])*gaussmf(x2(n), [0.8 -1]);%A1 B1
w2=gaussmf(x1(n), [0.5 -0.7])*gaussmf(x2(n), [0.8 1]); %A1 B2
w3=gaussmf(x1(n), [0.5 0.7])*gaussmf(x2(n), [0.8 -1]); %A2 B1
w4=gaussmf(x1(n), [0.5 0.7])*gaussmf(x2(n), [0.8 1]); %A2 B2

% finding normalized firing strength (activity degree) of each rule
gamma1(n)= w1/(w1+w2+w3+w4);
gamma2(n)= w2/(w1+w2+w3+w4);
gamma3(n)= w3/(w1+w2+w3+w4);
gamma4(n)= w4/(w1+w2+w3+w4);

end

Gama1= diag(gamma1);
Gama2= diag(gamma2);
Gama3= diag(gamma3);
Gama4= diag(gamma4);
%% weighted least-squares approach applied per rule
teta=[inv(Xe'*Gama1*Xe)*Xe'*Gama1*y inv(Xe'*Gama2*Xe)*Xe'*Gama2*y
inv(Xe'*Gama3*Xe)*Xe'*Gama3*y inv(Xe'*Gama4*Xe)*Xe'*Gama4*y];

% END OF TRAINING

% PERFORMANCE FOR TRAINING DATA with founded rule parameter

for i= 1:length(x1)
    p1= teta(1,1)*x1(i)+teta(2,1)*x2(i);
    p2= teta(1,2)*x1(i)+teta(2,2)*x2(i);
    p3= teta(1,3)*x1(i)+teta(2,3)*x2(i);
    p4= teta(1,4)*x1(i)+teta(2,4)*x2(i);

    w1=gaussmf(x1(i), [0.5 -0.7])*gaussmf(x2(i), [0.8 -1]);%A1 B1
    w2=gaussmf(x1(i), [0.5 -0.7])*gaussmf(x2(i), [0.8 1]); %A1 B2
    w3=gaussmf(x1(i), [0.5 0.7])*gaussmf(x2(i), [0.8 -1]); %A2 B1
    w4=gaussmf(x1(i), [0.5 0.7])*gaussmf(x2(i), [0.8 1]); %A2 B2

    train_result(i)=(w1*p1+ w2*p2+w3*p3+w4*p4)/(w1+w2+w3+w4);
end

figure(3);
plot(train_result,'b. ');
hold on
plot(y,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%% END of PERFORMANCE FOR TRAINING DATA with founded rule parameter
% TESTING TRANSACTIONS
%% PERFORMANCE FOR TESTING DATA with founded rule parameter

for i= 1:length(x1t)

    p1= teta(1,1)*x1t(i)+teta(2,1)*x2t(i);
    p2= teta(1,2)*x1t(i)+teta(2,2)*x2t(i);
    p3= teta(1,3)*x1t(i)+teta(2,3)*x2t(i);
    p4= teta(1,4)*x1t(i)+teta(2,4)*x2t(i);

w1=gaussmf(x1t(i), [0.5 -0.7])*gaussmf(x2t(i), [0.8 -1]);%A1 B1

```

```

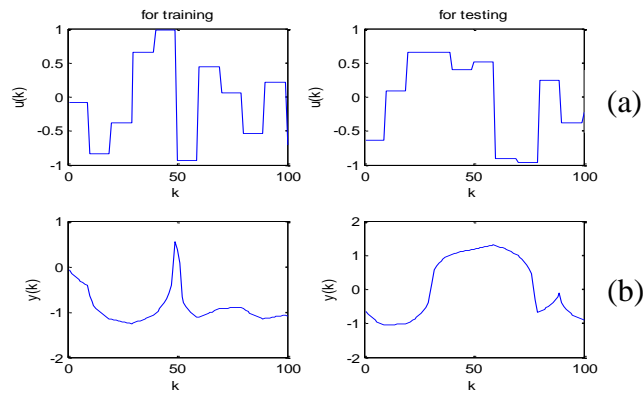
w2=gaussmf(x1t(i), [0.5 -0.7])*gaussmf(x2t(i), [0.8 1]); %A1 B2
w3=gaussmf(x1t(i), [0.5 0.7])*gaussmf(x2t(i), [0.8 -1]); %A2 B1
w4=gaussmf(x1t(i), [0.5 0.7])*gaussmf(x2t(i), [0.8 1]); %A2 B2

test_result(i)=(w1*p1+ w2*p2+w3*p3+w4*p4)/(w1+w2+w3+w4);
end

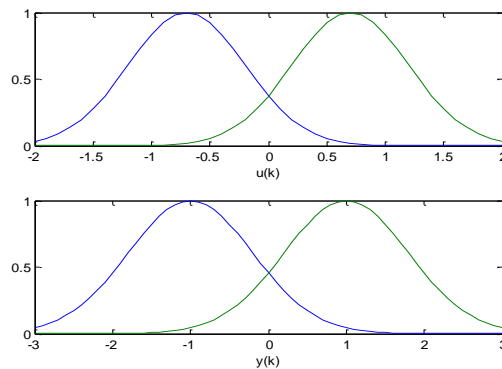
figure(4);
plot(test_result,'r-'),
hold on
plot(yt,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%% END of PERFORMANCE FOR TESTING DATA with founded rule parameter

```

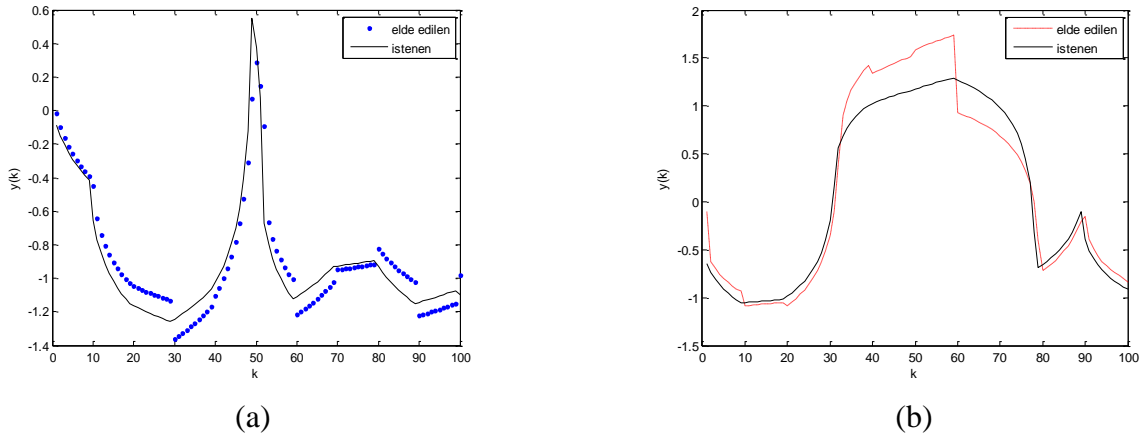
Bulanık sistemimiz ile ilgili grafiksel bilgiler Şekil 4.7-4.9’da verilmiştir.



Şekil 4.7. ÖDS 3 için eğitim (a) ve test (b) giriş  $u(k)$  ve çıkış  $y(k)$  dizileri.



Şekil 4.8. ÖDS 3’in modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.



Şekil 4.9. Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.

#### 4.1.4. Dinamik sistem modelleme örnek 4

Dinamik sistemin matematiksel tanımı (4.17) de (Oussar, Rivals ve Dreyfus, 1998) verilmiştir. Bu sistem  $[-5 \ 5]$  aralığında sürekli darbelerden oluşan  $u(k)$  girişleri ile eğitim ve test verisi olarak tasarlanmıştır. Sistem 10 örnekleme periyodu ile çalışacak şekilde hazırlanmıştır. Sistemde  $u(k)$ ,  $y(k)$  ve  $y(-1)$  girişler olarak alınmıştır. Bu sistemde 32 parametre eğitim verisi için belirlenmiştir.

$$y(k + 1) = \frac{24 + y(k)}{30} + y(k) - 0.8 \frac{u(k)^2}{1 + u(k)^2} y(k - 1) + 0.5u(k) \quad (4.17)$$

```

%%prapare the input/output training
load P4veri.mat
x1= veri(:,3); % first input for training
x2= veri(:,2); % second input for training
x3= veri(:,1); % third input fpr training
x=[x1 x2 x3]; % inputs for training
y= veri(:,4); %output for training
%% prapare the input output for testing
load P4veritest.mat
x1t= veri(:,3); %first input for testing
x2t=veri(:,2); %second input for testing
x3t= veri(:,1); %thrd input for testing
xt= [x1t x2t x3t]; %inputs for testing
yt= veri(:,4); %output for testing
%% plotting
subplot(2,2,1); plot(x1); xlabel('k'); ylabel('u(k)');
subplot(2,2,3); plot(y); xlabel('k'); ylabel('y(k)');

```

```

subplot(2,2,2); plot(x1t); xlabel('k'); ylabel ('u(k)');
subplot(2,2,4); plot(yt); xlabel('k'); ylabel ('y(k)');
teta=[]; %% parameter array/matrix
Xe= [x ones(size(x1))]; %%% extended input

%% membership funtions deffination
%for the first input ,u(k)
x1_array= -1:0.1:4;
    A1 = gaussmf(x1_array, [0.8 0]);
    A2 = gaussmf(x1_array, [1 2.5]);
    figure(2)
    subplot(3,1,1); plot(x1_array, A1, x1_array, A2);
    xlabel('u(k)');

%for the second input ,y(k)
x2_array= -1:0.1:4;
    B1 = gaussmf(x2_array, [0.8 0]);
    B2 = gaussmf(x2_array, [1 2.5]);
    subplot(3,1,2); plot(x2_array, B1, x2_array, B2);
    xlabel('y(k)');

%for the third input ,y(k-1)
x3_array= -1:0.1:4;
    C1 = gaussmf(x3_array, [0.8 0]);
    C2 = gaussmf(x3_array, [1 2.5]);
    subplot(3,1,3); plot(x3_array, C1, x3_array, C2);
    xlabel('y(k-1)');

%% TRAINING TRANSACTIONS
%% %% computing gamma values
for n= 1:length(x1)
    %finding firing strength (activity degree) of each rule
    w1=gaussmf(x1(n),[0.8 0])*gaussmf(x2(n),[0.8 0])*gaussmf(x3(n),[0.8 0]);%A1 B1 C1
    w2=gaussmf(x1(n), [0.8 0])*gaussmf(x2(n), [0.8 0])*gaussmf(x3(n), [1 2.5]); %A1 B1 C2
    w3=gaussmf(x1(n), [0.8 0])*gaussmf(x2(n), [1 2.5])*gaussmf(x3(n), [0.8 0]); %A1 B2 C1
    w4=gaussmf(x1(n), [0.8 0])*gaussmf(x2(n), [1 2.5])*gaussmf(x3(n), [1 2.5]); %A1 B2 C2
    w5=gaussmf(x1(n), [1 2.5])*gaussmf(x2(n), [0.8 0])*gaussmf(x3(n), [0.8 0]); %A2 B1 C1
    w6=gaussmf(x1(n), [1 2.5])*gaussmf(x2(n), [0.8 0])*gaussmf(x3(n), [1 2.5]); %A2 B1 C2
    w7=gaussmf(x1(n), [1 2.5])*gaussmf(x2(n), [1 2.5])*gaussmf(x3(n), [0.8 0]);%A2 B2 C1
    w8=gaussmf(x1(n), [1 2.5])*gaussmf(x2(n), [1 2.5])*gaussmf(x3(n), [1 2.5]); %A2 B2 C2
    %finding normalized firing strength (activity degree) of each rule
    gamma1(n)= w1/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma2(n)= w2/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma3(n)= w3/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma4(n)= w4/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma5(n)= w5/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma6(n)= w6/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma7(n)= w7/(w1+w2+w3+w4+w5+w6+w7+w8);
    gamma8(n)= w8/(w1+w2+w3+w4+w5+w6+w7+w8);

end

Gama1= diag(gamma1);
Gama2= diag(gamma2);
Gama3= diag(gamma3);
Gama4= diag(gamma4);
Gama5= diag(gamma5);

```

```

Gama6= diag(gamma6);
Gama7= diag(gamma7);
Gama8= diag(gamma8);

%% weighted least-squares approach applied per rule
teta=[inv(Xe'*Gama1*Xe)*Xe'*Gama1*y inv(Xe'*Gama2*Xe)*Xe'*Gama2*y
inv(Xe'*Gama3*Xe)*Xe'*Gama3*y inv(Xe'*Gama4*Xe)*Xe'*Gama4*y...
inv(Xe'*Gama5*Xe)*Xe'*Gama5*y inv(Xe'*Gama6*Xe)*Xe'*Gama6*y
inv(Xe'*Gama7*Xe)*Xe'*Gama7*y inv(Xe'*Gama8*Xe)*Xe'*Gama8*y];

%% END OF TRAINING

%% PERFORMANCE FOR TRAINING DATA with founded rule parameter

for i= 1:length(x1)

p1= teta(1,1)*x1(i)+teta(2,1)*x2(i)+teta(3,1)*x3(i)+ teta(4,1);
p2= teta(1,2)*x1(i)+teta(2,2)*x2(i)+teta(3,2)*x3(i)+ teta(4,2);
p3= teta(1,3)*x1(i)+teta(2,3)*x2(i)+teta(3,3)*x3(i)+ teta(4,3);
p4= teta(1,4)*x1(i)+teta(2,4)*x2(i)+teta(3,4)*x3(i)+ teta(4,4);
p5= teta(1,5)*x1(i)+teta(2,5)*x2(i)+teta(3,5)*x3(i)+ teta(4,5);
p6= teta(1,6)*x1(i)+teta(2,6)*x2(i)+teta(3,6)*x3(i)+ teta(4,6);
p7= teta(1,7)*x1(i)+teta(2,7)*x2(i)+teta(3,7)*x3(i)+ teta(4,7);
p8= teta(1,8)*x1(i)+teta(2,8)*x2(i)+teta(3,8)*x3(i)+ teta(4,8);
w1=gaussmf(x1(i), [0.8 0])*gaussmf(x2(i), [0.8 0])*gaussmf(x3(i), [0.8 0]);%A1 B1 C1
w2=gaussmf(x1(i), [0.8 0])*gaussmf(x2(i), [0.8 0])*gaussmf(x3(i), [1 2.5]); %A1 B1 C2
w3=gaussmf(x1(i), [0.8 0])*gaussmf(x2(i), [1 2.5])*gaussmf(x3(i), [0.8 0]); %A1 B2 C1
w4=gaussmf(x1(i), [0.8 0])*gaussmf(x2(i), [1 2.5])*gaussmf(x3(i), [1 2.5]); %A1 B2 C2
w5=gaussmf(x1(i), [1 2.5])*gaussmf(x2(i), [0.8 0])*gaussmf(x3(i), [0.8 0]); %A2 B1 C1
w6=gaussmf(x1(i), [1 2.5])*gaussmf(x2(i), [0.8 0])*gaussmf(x3(i), [1 2.5]); %A2 B1 C2
w7=gaussmf(x1(i), [1 2.5])*gaussmf(x2(i), [1 2.5])*gaussmf(x3(i), [0.8 0]);%A2 B2 C1
w8=gaussmf(x1(i), [1 2.5])*gaussmf(x2(i), [1 2.5])*gaussmf(x3(i), [1 2.5]); %A2 B2 C2

train_result(i)=(w1*p1+
w2*p2+w3*p3+w4*p4+w5*p5+w6*p6+w7*p7+w8*p8)/(w1+w2+w3+w4+w5+w6+w7+w8);
end
figure(3);
plot(train_result,'g');
hold on
plot(y,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')

%% END of PERFORMANCE FOR TRAINING DATA with founded rule parameter
%% TESTING TRANSACTIONS
%% PERFORMANCE FOR TESTING DATA with founded rule parameter

for i= 1:length(x1t)

p1= teta(1,1)*x1t(i)+teta(2,1)*x2t(i)+teta(3,1)*x3t(i)+ teta(4,1);
p2= teta(1,2)*x1t(i)+teta(2,2)*x2t(i)+teta(3,2)*x3t(i)+ teta(4,2);
p3= teta(1,3)*x1t(i)+teta(2,3)*x2t(i)+teta(3,3)*x3t(i)+ teta(4,3);
p4= teta(1,4)*x1t(i)+teta(2,4)*x2t(i)+teta(3,4)*x3t(i)+ teta(4,4);
p5= teta(1,5)*x1t(i)+teta(2,5)*x2t(i)+teta(3,5)*x3t(i)+ teta(4,5);
p6= teta(1,6)*x1t(i)+teta(2,6)*x2t(i)+teta(3,6)*x3t(i)+ teta(4,6);
p7= teta(1,7)*x1t(i)+teta(2,7)*x2t(i)+teta(3,7)*x3t(i)+ teta(4,7);
p8= teta(1,8)*x1t(i)+teta(2,8)*x2t(i)+teta(3,8)*x3t(i)+ teta(4,8);

```

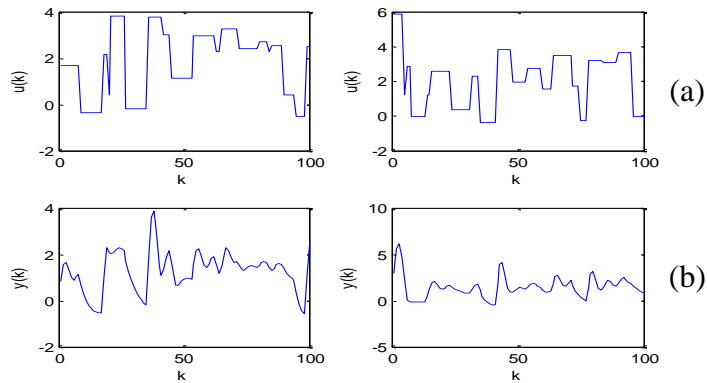
```

w1=gaussmf(x1t(i), [0.8 0])*gaussmf(x2t(i), [0.8 0])*gaussmf(x3t(i), [0.8 0]);%A1 B1 C1
w2=gaussmf(x1t(i), [0.8 0])*gaussmf(x2t(i), [0.8 0])*gaussmf(x3t(i), [1 2.5]); %A1 B1 C2
w3=gaussmf(x1t(i), [0.8 0])*gaussmf(x2t(i), [1 2.5])*gaussmf(x3t(i), [0.8 0]); %A1 B2 C1
w4=gaussmf(x1t(i), [0.8 0])*gaussmf(x2t(i), [1 2.5])*gaussmf(x3t(i), [1 2.5]); %A1 B2 C2
w5=gaussmf(x1t(i), [1 2.5])*gaussmf(x2t(i), [0.8 0])*gaussmf(x3t(i), [0.8 0]); %A2 B1 C1
w6=gaussmf(x1t(i), [1 2.5])*gaussmf(x2t(i), [0.8 0])*gaussmf(x3t(i), [1 2.5]); %A2 B1 C2
w7=gaussmf(x1t(i), [1 2.5])*gaussmf(x2t(i), [1 2.5])*gaussmf(x3t(i), [0.8 0]);%A2 B2 C1
w8=gaussmf(x1t(i), [1 2.5])*gaussmf(x2t(i), [1 2.5])*gaussmf(x3t(i), [1 2.5]); %A2 B2 C2

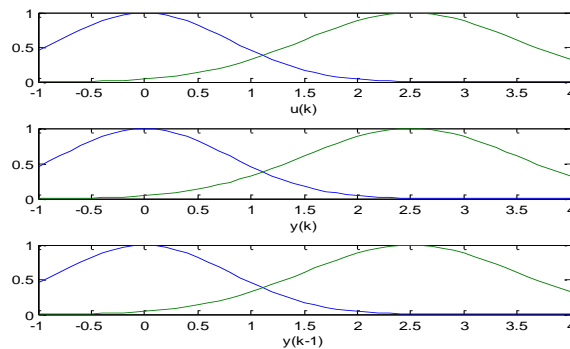
test_result(i)=(w1*p1+
w2*p2+w3*p3+w4*p4+w5*p5+w6*p6+w7*p7+w8*p8)/(w1+w2+w3+w4+w5+w6+w7+w8);
end
figure(4);
plot(test_result,'r-');
hold on
plot(yt,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%% END of PERFORMANCE FOR TESTING DATA with founded rule parameter

```

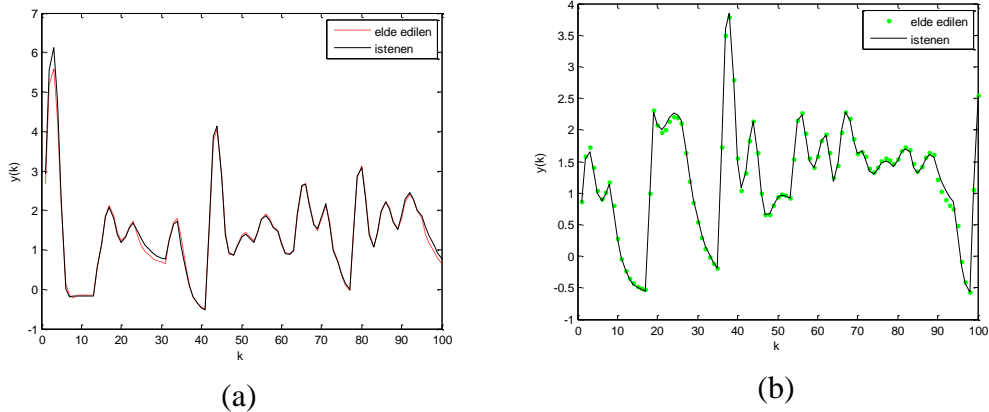
Bu örnek için grafiksel bilgiler Şekil 4.10-4.12’da verilmiştir



Şekil 4.10. ÖDS 4 için eğitim (a) ve test (b) giriş  $u(k)$  ve çıkış  $y(k)$  dizileri.



Şekil 4.11. ÖDS 4’ün modellenmesinde kullanılan giriş değişkenlerinin üyelik fonksiyonları.



**Şekil 4.12.** Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.

#### 4.1.5. Dinamik sistem modelleme örnek 5

Dinamik sistemin tanımı (4.18)'de ( Sastry, Santharam ve Unnikrishnan, 1994) verilmiştir. Bu sistem için  $[-2 \ 2]$  aralığında sürekli darbelerden oluşan  $u(k)$  girişleri ile eğitim ve test verisi hazırlanmıştır. Sistem rasgele örnekleme periyodu ile çalışacak şekilde hazırlanmıştır. Sistemde  $u(k)$  ve  $y(k)$  girişler olarak alınmıştır. Bu sistemde 12 parametre eğitim verisi kullanılarak belirlenmiştir.

$$y(k+1) = 0.5 * \left( \frac{y(k)}{1 + y^2(k)} + (1 + u(k))u(k)(1 - u(k)) \right) \quad (4.18)$$

```
load P5veri.mat
x1= veri(:,2); % first input for training
x2= veri(:,1); %second input for trainig
x=[x1 x2];
y= veri(:,3); %output for training
%% prepare the input/output for testing
load P5veritest.mat
x1t= veri (:,2); %first input for tetig
x2t= veri (:,1); %second input for testing
xt=[x1t x2t]; % input for testig
yt= veri(:,3); %% output for testing
%%% plotting
subplot(2,2,1); plot(x1); xlabel('uk'); ylabel('y(k)'); title('for training');
subplot (2,2,3); plot(y); xlabel('uk'); ylabel ('y(k)');

subplot(2,2,2); plot(x1t); xlabel('uk'); ylabel('y(k)'); title('for training');
subplot (2,2,4); plot(yt); xlabel('uk'); ylabel ('y(k)');

teta=[]; %% parameter array/matrix
```

```

Xe= [x ones(size(x1))]; %% extended input
%% membership funtions defination
%% for the first input u(k)
x1_array= -2:0.1:2;
A1= gaussmf(x1_array, [0.5 -0.8]);
A2= gaussmf(x1_array, [0.5 0.8]);
figure (2)
subplot(2,1,1); plot(x1_array, A1, x1_array, A2);
xlabel('u(k)');
title ('MFs for u(k) input');

x2_array= -3:0.1:4;
B1= gaussmf(x2_array, [0.8 -1]);
B2= gaussmf(x2_array, [0.8 1]);
subplot (2,1,2); plot (x2_array, B1, x2_array,B2);
xlabel('y(k)');
title ('MFs for y(k) input');
% TRAINING TRANSACTIONS
% %% computing gamma values
for n= 1:length(x1)
    %finding firing strength (activity degree) of each rule
    w1=gaussmf(x1(n), [0.5 -0.8])*gaussmf(x2(n), [0.8 -1]);%A1 B1
    w2=gaussmf(x1(n), [0.5 -0.8])*gaussmf(x2(n), [0.8 1]); %A1 B2
    w3=gaussmf(x1(n), [0.5 0.8])*gaussmf(x2(n), [0.8 -1]); %A2 B1
    w4=gaussmf(x1(n), [0.5 0.8])*gaussmf(x2(n), [0.8 1]); %A2 B2

    %finding normalized firing strength (activity degree) of each rule
    gamma1(n)= w1/(w1+w2+w3+w4);
    gamma2(n)= w2/(w1+w2+w3+w4);
    gamma3(n)= w3/(w1+w2+w3+w4);
    gamma4(n)= w4/(w1+w2+w3+w4);

end

Gama1= diag(gamma1);
Gama2= diag(gamma2);
Gama3= diag(gamma3);
Gama4= diag(gamma4);

%% weighted least-squares approach appliedper rule
teta=[inv(Xe'*Gama1*Xe)*Xe'*Gama1*y inv(Xe'*Gama2*Xe)*Xe'*Gama2*y
inv(Xe'*Gama3*Xe)*Xe'*Gama3*y inv(Xe'*Gama4*Xe)*Xe'*Gama4*y];

%% END OF TRAINING

%% PERFORMANCE FOR TRAINING DATA with founded rule parameter

for i= 1:length(x1)

    p1= teta(1,1)*x1(i)+teta(2,1)*x2(i);
    p2= teta(1,2)*x1(i)+teta(2,2)*x2(i);
    p3= teta(1,3)*x1(i)+teta(2,3)*x2(i);
    p4= teta(1,4)*x1(i)+teta(2,4)*x2(i);

    w1=gaussmf(x1(i), [0.5 -0.8])*gaussmf(x2(i), [0.8 -1]);%A1 B1
    w2=gaussmf(x1(i), [0.5 -0.8])*gaussmf(x2(i), [0.8 1]); %A1 B2

```

```

w3=gaussmf(x1(i), [0.5 0.8])*gaussmf(x2(i), [0.8 1]); %A2 B1
w4=gaussmf(x1(i), [0.5 0.8])*gaussmf(x2(i), [0.8 1]); %A2 B2

train_result(i)=(w1*p1+ w2*p2+w3*p3+w4*p4)/(w1+w2+w3+w4);
end

figure(3);
plot(train_result,'b-'),
title('training performance of fuzzy system');
hold on
plot(y,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')

%% END of PERFORMANCE FOR TRAINING DATA with founded rule parameter
% TESTING TRANSACTIONS
%% PERFORMANCE FOR TESTING DATA with founded rule parameter

for i= 1:length(x1t)

    p1= teta(1,1)*x1t(i)+teta(2,1)*x2t(i);
    p2= teta(1,2)*x1t(i)+teta(2,2)*x2t(i);
    p3= teta(1,3)*x1t(i)+teta(2,3)*x2t(i);
    p4= teta(1,4)*x1t(i)+teta(2,4)*x2t(i);

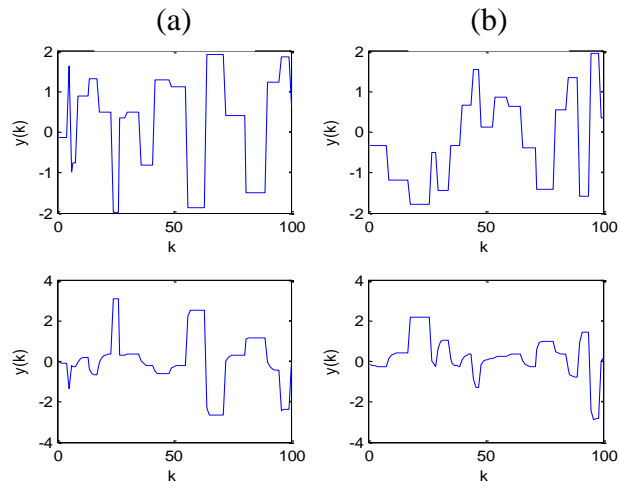
w1=gaussmf(x1t(i), [0.5 -0.8])*gaussmf(x2t(i), [0.8 -1]);%A1 B1
w2=gaussmf(x1t(i), [0.5 0.8])*gaussmf(x2t(i), [0.8 1]); %A1 B2
w3=gaussmf(x1t(i), [0.5 0.8])*gaussmf(x2t(i), [0.8 -1]); %A2 B1
w4=gaussmf(x1t(i), [0.5 0.8])*gaussmf(x2t(i), [0.8 1]); %A2 B2

    test_result(i)=(w1*p1+ w2*p2+w3*p3+w4*p4)/(w1+w2+w3+w4);
end

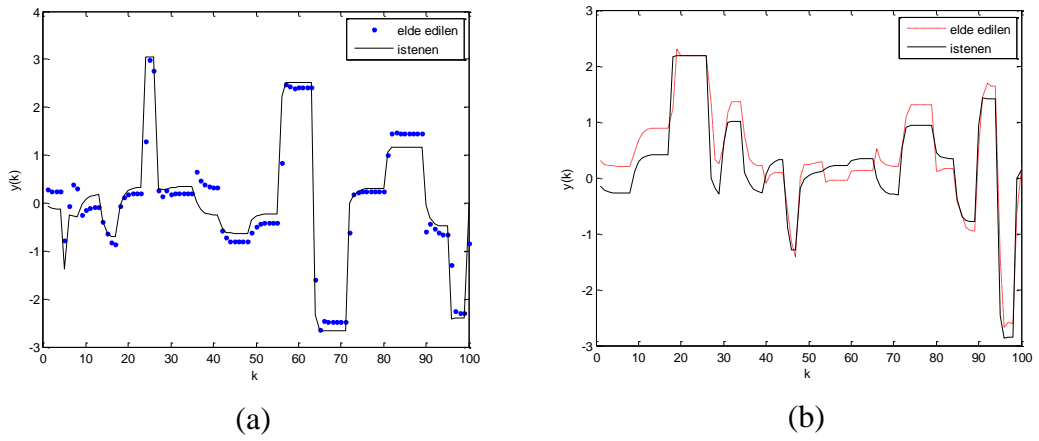
figure(4);
plot(test_result,'r-'),
title('testing performance of fuzzy system');
hold on
plot(yt,'k'); hold off
xlabel('k'), ylabel('y(k)'); legend('elde edilen', 'istenen')
%% END of PERFORMANCE FOR TESTING DATA with founded rule parameter

```

Bulanık sistemin eğitim ve testi ile ilgili grafiksel bilgiler Şekil 4.13-4.14’te verilmiştir



Şekil 4.13. ÖDS 5 için eğitim (a) ve test (b) giriş  $u(k)$  ve çıkış  $y(k)$  dizileri.



Şekil 4.14. Eğitim (a) ve test (b) verisi için elde edilen modelleme başarımı.

## 5. SONUÇ ve ÖNERİLER

Bu tezde problemlerin bulanık mantık yolu ile ve çeşitli modeller kullanarak çözümlenmesi üzerinde öncelikle çalışılmıştır. Bu amaçla 6 adet örnek bulanık sistem tanımlanmış olup bu sistemlerin gerçeklediği giriş-çıkış yüzeyle Matlab programı ile elde edilmiştir. Bu örneklerde yaygın olarak kullanılan Tagaki-Sugeno ve Mamdani bulanık çıkarım modelleri kullanılmıştır.

Çalışmanın ikinci aşamasında Tagaki-Sugeno tipi bulanık modelin parametrelerinin belirlenmesi için şablon bir yöntem tanımlanmıştır. Bu yöntemin etkinliği literatürden alınan 5 farklı dinamik sistemin Tagaki-Sugeno tipi bulanık model ile modellenmesi üzerinde gösterilmiştir. Tanımlanan bu yöntem en küçük kareler yöntemine dayalı olarak geliştirilmiş bir yöntemdir. Bu yöntem eldeki giriş-çıkış verilerinden hareketle bulanık sistemin parametrelerini belirlemektedir.

Bu çalışmada sunulan yöntemde, bulanık sistemin şart (giriş üyelik fonksiyonları) parametreleri üyelik fonksiyonlarının ilgili uzayda homojen dağılımı sağlanarak belirlenmektedir. Kural parametreleri ise en küçük kareler kestirimine dayalı olarak geliştirilen kural bazında geçerli bir cebirsel eşitlik ile belirlenmektedir. Özetle bu çalışmada, veriye dayalı bulanık modelleme için şablon bir öğrenme yöntemi tanımlanmış olup bu yöntemin kullanımı kod bazında bir örnek ile açıkça gösterilmiştir. Tanımlanan modelleme yönteminin başarımı beş farklı eğrisel örnek dinamik sistem modellemesi üzerinde de ayrıca kapsamlı olarak gösterilmiştir.

Bu dinamik sistem modelleme görevi için önce bulanık sistemin öğrenmesinde kullanmak üzere eğitim veri seti hazırlanmıştır. Bu eğitim veri seti ile bulanık sistemin parametreleri tanımlanan şablon yöntem ile belirlenmiştir. Böylece oluşturulan dinamik sistem bulanık modelinin başarımını gözlemek amacı ile bir test veri seti de hazırlanmıştır. Bu test verisi ile öğrenmesi tamamlanan bulanık model başarımı test edilmiştir. Şeki 4.3, Şekil 4.6, Şekil 4.9, Şekil 4.12 ve Şekil 4.14'de verilen grafiksel karşılaştırmalı sonuçlardan görüleceği üzere yöntemin başarımı oldukça tatmin edici seviyede bulunmuştur. Burada verilen şablon modelleme yöntemi ile yeteri sayıda giriş-çıkış örneği bulunan muhtelif problemler için bulanık model elde edilebilir

## KAYNAKLAR

- Alıcı, M., ve Karatepe, E., “ Bulanık Mantık ve Matlab Uygulamaları” İzmir (2012)
- Bülent Çobanoğlu “Bulanık Mantık ve Bulanık Küme Teorisi” , 2000, Intl: <https://cobanoglu.wikispaces.com/file/view/bulanikmantik.pdf>, (Ziyaret edilme tarihi 12.04.2017).
- Can Bozkurt, ‘ bulanık mantık uygulama Örnekleri’, Intl: <http://bulanikmantikbmc.blogspot.com.tr/p/bulank-mantk-uygulama-ornekleri.html>, (ziyaret edilme tarihi 20.02.2017).
- Elmas, Ç., “Yapay Zeka Uygulamaları, Yapay Sinir Ağları•Bulanık Mantık•Genetik Algoritma”, *Seçkin Yayınları*, Ankara (2011)
- Ertunç, H.M “bulanık mantığa giriş”, Intl: [https://endstriorganizasyon.files.wordpress.com/2016/03/bulanc4b1k\\_mantc4b1k\\_giric59f\\_sunu.pdf](https://endstriorganizasyon.files.wordpress.com/2016/03/bulanc4b1k_mantc4b1k_giric59f_sunu.pdf), (2012).
- İnaç, T., “BBO Algoritmesinin Optimizasyon Başaramının İncelenmesi” Yüksek Lisans Tezi, *Bilecik Şeyh Edebali Üniversitesi*, Bilecik (2014).
- K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, 1(1):4-27(1990).
- Karakuzu, C.,“Ders 5 : Bulanık modeller”uthern Brazil (Coleoptera, Cantharidae)”, *BM545Bulanık Sistem ve Uygulamaları ders notu, Bilecik Şeyh Edebali Üniveritesi, Bilecik*, (2012).
- Karakuzu, C., & Demirci, O. , “Fuzzy logic based smart traffic light simulator design and hardware implementation.”, *Applied Soft Computing*, 10(1): 66-73( 2010).
- L. Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes”, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1):28-44( 1973).

- Nevra Aydın, “Bulanık Mantık Nasıl Çalışır?”, Intl:  
<http://www.elektrikport.com/teknik-kutuphane/bulanik-mantik-nasil-calisir-1-bolum/14849#ad-image-0> , (Ziyaret tarihi 10.02.2017) .
- Novruz Allahverdi, “Bulanık Kümeler”, 2001, Intl:  
<http://farabi.sutef.gen.tr/bulanik/bulanik/bolum02.htm> (Ziyaret edilme tarihi 25.03.2017).
- P. S. Sastry, G. Santharam ve K. P. Unnikrishnan, “Memory neuron networks for identification and control of dynamical systems”, *IEEE Transaction on Neural Networks*, 5( 2): 306 – 319(1994).
- R. Babuska, “Fuzzy Systems, Modeling and Identification”, [Çevrimiçi].  
 Available:[https://www.researchgate.net/profile/Robert\\_Babuska/publication/228769192\\_Fuzzy\\_Systems\\_Modeling\\_and\\_Identification/links/02e7e5223310e79d19000000.pdf?origin=publication\\_list&ev=srch\\_pub\\_xdl](https://www.researchgate.net/profile/Robert_Babuska/publication/228769192_Fuzzy_Systems_Modeling_and_Identification/links/02e7e5223310e79d19000000.pdf?origin=publication_list&ev=srch_pub_xdl). [Erişildi: 22 12 2016].
- R. Babuška, Fuzzy Modeling for Control, Kluwer Academic Publishers, 1998.
- Sarı, M., Murat, Y. Ş., ve Kırabalı, M., “Bulanık Modelleme Yaklaşımı ve Uygulamaları” Intl:  
<https://birimler.dpu.edu.tr/app/views/panel/ckfinder/userfiles/16/files/Dergiler/9/09.pdf>
- Serhat Yılmaz, “bulanık sistemler”, Intl:  
[http://www.yarbis1.yildiz.edu.tr/web/userCourseMaterials/eakdogan\\_784d798d417e811a3ba5a79ff8c46a30.pdf](http://www.yarbis1.yildiz.edu.tr/web/userCourseMaterials/eakdogan_784d798d417e811a3ba5a79ff8c46a30.pdf) , (Ziyaret edilme tarihi 17.02.2017).
- Şükrü Köşe, “Yapay Zekâ Bulanık Mantık Yöntemi”, 2017, Intl:  
<http://80.251.40.59/science.ankara.edu.tr/ozbek/bulanik-1.htm>, (Ziyaret edilme tarihi 09.03.2017).
- T. Takagi ve M. Sugeno, “Fuzzy identification of systems and its application to modeling and control”, *IEEE Trans. on Systems, Man, and Cybernetics*, %1 / %2SMC-15(1):116-132(1985).

Yıldırım, Ö., “Sezgisel Arama Algoritma Tabanlı Bulanık Sistem Optimizasyonu” ,  
Yüksek Lisans Tezi, *Bilecik Şeyh Edebali Üniversitesi*, Bilecik (2012).

Şen, Z “ Mühendislikte Bulanık Mantık ile Mdelleme Prensipleri”, *Su Vakfı*, İstanbul  
(2004).

Baykal, N. Ve Beyan, T., “Bulanık Mantık Uzman Sistemler ve Denetleyiciler”  
*Bıçaklar Kitabevi*, Ankara (2004).

ZAdeh, L. A., “Fuzzy Sets”, *Information and Cotrol*, 8:338-335(1965).

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Adı Soyadı: Abdoulaye Abdramane MAKHAÏLA

Dağum Yeri ve Tarihi: 07/04/1985 N'Djamena-Tchad



### Eğitim Durumu

Lisans Öğrenimi : Kral Faysal Üniversitesi Mühendislik Fakültesi, Teknik ve Bilgisayar Mühendisliği (2010).

Bildiği Yabancı Diller : Arapça, İngilizce, Fransızca

Bilimsel Faaliyetleri :

### İş Deneyimi

**Stajlar** : 2010 BİSİC bankasında (N'Djamena-Tchad)

**Projeler** :

**Çalıştığı Kurumlar** : ACHABIA Lisesinde matematik ve İngilizce öğretmenliği (N'Djamena-Tchad)

### Akademik Çalışmalar:

Cihan KARAKUZU, Abdoulaye Abdramane MAKHAÏLA, “*Veriye Dayalı Bulanık Mantık Tabanlı Dinamik Sistem Modelleme*”, Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, Cilt:4, Sayı:1, 2017 ISSN: 2458-7575 (<http://edergi.bilecik.edu.tr/index.php/fbd>).

### İletişim

**Adres** : Uçar str, Uçar cite apt no:6 /Bahçelievler dist. Bilecik/Türkiye

**Tel** : (+90) 5395894759

**E-Posta Adresi** : makailaabdoulaye@yahoo.fr

**Temuz 2017**