

ÖZET

Yüksek Lisans Tezi

BULANIK MANTIK ÇIKARIM SİSTEMİ İLE TREN HIZININ OTOMATİK KONTROLÜ

Erhan ÖZTOK

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalı

Danışman: Prof.Dr.Memmedağa MEMMEDLİ

2010, 85 sayfa

Bu tezde, bulanık mantık ve adaptif ağ yapısına dayalı bulanık çıkarım sistemi (Anfis) ile Otomatik Tren Yönetim Sistemi uygulaması yapılmıştır. Öncelikle Otomatik Tren Yönetim Sistemi üzerine yapılan teorik araştırmalardan yola çıkılarak, trenlerin bulanık mantık ile kontrol ve yönetimi sağlanmıştır. Daha sonra mevcut teknik bir adım daha ileriye götürülerek, yapay sinir ağları ve bulanık mantığın bir arada kullanıldığı Anfis ile (dünyada ilk kez denenerek) modellenmiş ve bir simülatör algoritması geliştirilerek simülasyonu gerçekleştirilmiştir. Bulanık mantık ve Anfis kontrolünün birlikte uygulanması ile Otomatik Tren Yönetim Sistemi gerçek hayatta her bölge ve koşulda çalışabilecek, güvenliği en üst düzeyde olan bir sistem haline getirilmeye çalışılmış ve elde edilen sonuçlar sunulmuştur.

Anahtar Kelimeler: Yapay Sinir Ağları, Bulanık Mantık, Anfis, Otomatik Tren Yönetim Sistemi

ABSTRACT

Master of Science Thesis

AUTOMATIC TRAIN SPEED CONTROL WITH FUZZY LOGIC INFERENCE SYSTEM

Erhan ÖZTOK

**Anadolu University
Graduate School of Sciences
Statistic Program**

Supervisor: Prof.Dr. Memmedağa MEMMEDLİ

2010, 85 page

In this dissertation, fuzzy logic and adaptive network based fuzzy inference system (Anfis) applied to Automatic Train Operation System. Starting from the oretical inquiries on Automatic Train Operation System, a train control and management system based on fuzzy logic is obtained. Later this technique is taken one step further to a madeling and simulation using both artificial intelligence networks and fuzzy logic based Anfis, than modaled and tested for the first time in the world. With usage of fuzzy logic and Anfis control together on Automatic Train Operation System it is attempted to make a reliable system to work on every region and condition with highest security and attained results are presented.

Keywords: Neural Networks, Fuzzy Logic, Anfis, Automatic Train Operation System

TEŐEKKÜR

Yüksek lisans tezim süresince benden yardımlarını esirgemeyen, karşılaştığım sorunlarda bilgi ve deneyimlerini benimle paylaşan değerli hocam ve tez danışmanım Prof. Dr. Memmedağa MEMMEDLİ'ye teşekkürü bir borç bilirim.

Lisans öğrenimim boyunca beni destekleyen Prof. Dr. Ali Fuat YÜZER başta olmak üzere İstatistik Anabilim Dalında görev yapmakta olan Prof. Dr. Embiya AĞAOĞLU ve tüm hocalarıma teşekkürlerimi sunarım.

Yüksek lisansımı tamamlamamda bana büyük destek veren sevgili eşim Özlem ÖZTOK'a ve aileme sonsuz teşekkürler.

Erhan ÖZTOK

Şubat, 2010

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT	iii
TEŞEKKÜR.....	iv
İÇİNDEKİLER.....	v
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	xi
SİMGELER VE KISALTMALAR DİZİNİ.....	xii
1. GİRİŞ.....	1
2. BULANIK KÜMELER VE BULANIK MANTIK	3
2.1. Giriş.....	3
2.2. Bulanık Küme Kuramının Temel Kavramları	4
2.3. Bulanık Küme Üyelik Fonksiyonları	5
2.3.1. Basit Üyelik Fonksiyonu	5
2.3.2. Gaussian Üyelik Fonksiyonu	5
2.3.3. Genelleştirilmiş Gaussian Üyelik Fonksiyonu.....	5
2.3.4. Sigmoidal Üyelik Fonksiyonu	5
2.3.5. Üçgensel Üyelik Fonksiyonu	6
2.3.6. Yamuksal Üyelik Fonksiyonu.....	6
2.4. Bulanık Kümelerde Operasyonlar	9
2.5. Bulanık İlişkiler.....	13
2.6. Bulanık Nedenlilik	14
2.7. Bulanık Mantık	16
3. BULANIK ÇIKARIM SİSTEMİ.....	20
3.1. Bulanık Çıkarım Sisteminin Katmanları	20
3.2. Bulanık Çıkartım Sisteminin Bir Örnek Üzerinde İncelenmesi	22
3.3. Durulandırma Yöntemi.....	26
3.3.1. Maksimum değeri alma	26

3.3.2. Maksimumların Ortalamasını Alma	26
3.3.3. Alan Merkezi Metodu.....	27
4. ADAPTİF AĞ YAPISINA DAYALI BULANIK ÇIKARIM SİSTEMİ ...	28
4.1. Yapay Sinir Ağları	28
4.1.1. Sinir ağlarında öğrenme ve Geri Yayılım Algoritması	29
4.1.2. Hibrid Öğrenme (Off-Line)	30
4.1.3. Hibrid Öğrenme (On-Line).....	31
4.2. Anfis Mimarisi	31
4.2.1. Katman 1	32
4.2.2. Katman 2.....	33
4.2.3. Katman 3	33
4.2.4. Katman 4.....	34
4.2.5. Katman 5.....	34
4.3. Hibrid Öğrenme Algoritması	35
4.4. Anfis Matlab Araçları.....	36
5. OTOMATİK TREN YÖNETİM SİSTEMİ	44
5.1. Otomatik Tren Yönetim Sisteminin Genel İşleyişi.....	44
5.2. Bulanık Çıkarım Sistemi ile Otomatik Tren Yönetim Sistemi	47
5.3. Otomatik Tren Yönetim Sistemi için Bulanık Kurallar	49
5.3.1. Sabit Hız Kontrolü (CSC).....	49
5.3.2. Otomatik Duruş Kontrolü (TASC).....	50
5.4. Otomatik Tren Yönetim Sistemi Bulanık Çıkarım Fonksiyonları.....	50
5.5. Bulanık Çıkarım Sistemi Fonksiyonlarının Matlab Kodları.....	51
5.6. Lokomotif Güç ve Fren Kontrol Bağlantı Şeması	52
5.7. Lokomotif Simülatörü	53
5.7.1. Lokomotif simülatörü matlab kodu.....	54
5.8. Performans İndeksleri - ATO Sistem İlişkisi.....	56
5.9. Performans İndeksleri Üyelik Fonksiyonları.....	57
5.9.1. İz takip indeksi (Traceability)	57
5.9.2. Konfor İndeksi (Comfort).....	59

5.9.3. Güvenlik indeksi (Safety)	61
5.9.4. İşleme zamanı performansı indeksi (Running Time Performance) ..	62
5.9.5. Duruş Mesafesi Performans İndeksi (Stop Gap Performance).....	64
5.9.6. Enerji tasarruf performansı indeksi (Energy Saving Performance) ..	65
5.10. Matlab’de Otomatik Tren Yönetim Sisteminin Tanımlanması	67
5.10.1. Sabit hız kontrol bölgesi.....	67
5.10.2. Güvenlik üyelik fonksiyonu	67
5.10.3. Konfor üyelik fonksiyonu.....	67
5.10.4. Enerji tasarrufu üyelik fonksiyonu.....	67
5.10.5. İzlenebilirlik üyelik fonksiyonu.....	68
5.10.6. Çıkış değişkeni parametreleri	68
5.10.7. Kural listesi.....	69
5.10.8. Kuralların eklenmesi	69
5.10.9. Giriş değişkenleri atamaları.....	69
5.10.10. Bulanık çıkarımının hesaplanması	69
5.10.11. Güç kademesi.....	69
5.10.12. Fren kademesi	70
5.10.13. Duruş mesafesi kontrol bölgesi.....	70
5.10.14. Çalışma zamanı üyelik fonksiyonu	70
5.10.15. Duruş mesafesi üyelik fonksiyonu.....	70
5.10.16. Konfor üyelik fonksiyonu.....	70
5.10.17. Çıkış değişkeni parametreleri	71
5.10.18. Kural listesi.....	71
5.10.19. Kuralların eklenmesi	72
5.10.20. Giriş değişkenleri atamaları.....	72
5.10.21. Bulanık çıkarımının hesaplanması	72
5.10.22. Güç kademesi.....	72
5.10.23. Fren kademesi	72
5.10.24. Ana döngü.....	72
5.10.25. Zamanlayıcı	73
5.10.26. Grafik çizimi	74
5.10.27. ts ve tz zamanlamalarının hesaplanması	74

5.10.28. Simülasyonun çalıştırılması.....	75
5.11. Bulanık Mantık ile Kontrol ve Simülasyon Sonuçları	75
5.12. Anfis ile Kontrol ve Simülasyon Sonuçları	76
5.13. Otomatik Tren Yönetim Sistemi Uygulaması Sonuçları.....	80
6. SONUÇ VE ÖNERİLER	81
KAYNAKLAR.....	82
Ek-1 Eğitim Verisi.....	84

ŞEKİLLER DİZİNİ

2.1.	Üyelik Fonksiyon Grafikleri.....	7
2.2.	Bulanık Kümelerde Destek ve Öz İlişkisi.....	7
2.3.	Min/Max Norm Grafikleri	11
2.4.	Cebir Üçgensel Norm Grafikleri.....	11
2.5.	Bulanık Kümenin Tersisi Grafiği	12
2.6.	Minimum Operatör Grafiği.....	13
2.7.	Çarpım Operatör Grafiği.....	13
2.8.	Bulanık Mantık İşleyiş Aşamaları.....	19
3.1.	Bulanık Çıkarım Sistem Mimarisi.....	20
3.2.	Bulanık Nedensellik Adımları	21
3.3.	X,Y Değişkenlerinin Üyelik Fonksiyonları	22
3.4.	Kesikli Değişkenin min Operatörü ile Üyelik Fonksiyonları	23
3.5.	X Kesikli Değişkenin Çarpım Operatörü ile Üyelik Fonksiyonları.....	24
3.6.	Lukasiewicz Operatörü ile Üyelik Fonksiyonları	25
3.7.	Örnek Durulaştırma İşlemi	25
3.8.	Maksimum Üyelik Metodu Grafiği	26
3.9.	Maksimumların Ortalamasını Alma Metodu Çıkış Değeri.....	26
4.1.	Çok Katmanlı Yapay Sinir Ağı.....	28
4.2.	Anfis Mimarisi	32
4.3.	Anfis Bulanık İlişki Mimarisi	34
4.4.	Anfis Bulanık İlişki Kuralları	35
4.5.	Mamdani ilişkili FIS Editörünün Standart Görüntüsü.....	37
4.6.	Giriş Değerlerinin Üyelik Fonksiyonları.....	37
4.7.	Çıkış Değerlerinin Üyelik Fonksiyonları.....	38
4.8.	Çıkış Değerlerinin Aralıklarının Darlaştırılması.....	38
4.9.	Giriş-Çıkış Kural Tablosu.....	39
4.10.	Giriş-Çıkış Kural İzleme Ekranı	39
4.11.	Giriş-Çıkış Sonuç Yüzey İzleme Ekranı.....	40
4.12.	Sugeno ilişkili FIS Editörünün Standart Görüntüsü.....	40
4.13.	Sugeno ilişkili FIS Editörünün Üyelik Fonksiyon Giriş Ekranı	41

4.14.	Sugeno ilişkili FIS Editörünün Çıkış Birimi Üyelik Fonksiyonları.....	41
4.15.	Sugeno ilişkili FIS Editörünün Çıkış Birimi Üyelik Fonksiyonları.....	42
4.16.	Sugeno ilişkili FIS Editörünün Kural Tanımlamaları	42
4.17.	Sugeno ilişkili FIS Editörünün Kural İzleme Ekranı	43
5.1.	Otomatik Tren Yönetim Sistemi	45
5.2.	Otomatik Tren Yönetim Sistemi Diyagramı.....	46
5.3.	Otomatik Tren Yönetim Sistemi Akış Şeması.....	48
5.4.	Birim Bağlantı Diyagramı	53
5.5.	Lokomotif Simülatörü Dış Etmenler Diyagramı.....	54
5.6.	Performans İndeksleri ve İlişkileri	56
5.7.	İz Takip İndeksi Üyelik Fonksiyonu	59
5.8.	Konfor İndeksi Üyelik Fonksiyonu	60
5.9.	Güvenlik İndeksi Üyelik Fonksiyonu.....	62
5.10.	İşleme Zamanı İndeksi Üyelik Fonksiyonu	63
5.11.	Duruş Mesafesi İndeksi Üyelik Fonksiyonu.....	65
5.12.	Enerji Tassaruf İndeksi Üyelik Fonksiyonu	66
8.13.	Hız - Konum Grafiği	75
5.14.	Bulanık Mantık Kontrollü Anfis Test Grafiği.....	77
5.15.	Bulanık Mantık Kontrollünün Hata Grafiği.....	77
5.16.	Bulanık Mantık Kontrollü Anfis Mimarisi	78
5.17.	Bulanık Mantık Kontrollü Yüzey Grafiği.....	78
5.18.	Bulanık Mantık Kontrollü Bulanık Çıkarım kuralları.....	79
5.19.	Anfis Kontrollü Hata Grafiği	79
5.20.	Anfis Kontrollü Test Grafiği.....	80

ÇİZELGELER DİZİNİ

2.1. Bulanık İçerme Fonksiyon Tablosu.....	15
3.1. X,Y kümelerinin İlişki Matrisi	22
3.2. X Kesikli Değişkenin Min Operatörü ile İlişki Matrisi	23
3.3. Kesikli Değişkenin Çarpım Operatörü ile İlişki Matrisi.....	23
3.4. Lukasiewicz Operatörü ile Aldığı Değerler	24
4.1. Parametre Güncelleme Kuralları	36
4.2. Örnek Matris.....	36

SİMGELER VE KISALTMALAR DİZİNİ

ATO	: Otomatik Tren Yönetimi
CSC	: Sabit Hız Kontrolü
TASC	: Otomatik Duruş Kontrolü
t	: Zaman (saniye)
$x(t)$: Trenin t anındaki konumu (m)
$v(t)$: Trenin t anındaki hızı (km/sa)
$N(t)$: t anındaki Kademe kontrol komutları
$X(t)$: Sonraki istasyonun hedef konumu (m)
V_t	: Hedef hız (km/sa)
T_t	: Hesaplanan çalışma zamanı (sn)
x_d	: Maksimum hız limitinin düştüğü konum (m)
t_s	: x_d noktasına erişme zamanı (sn)
x_k	: Yokuş aşağı ilerlemenin son konumu (m)
$x_z(v)$: TASC bölgesinin başlama noktası (m)
t_z	: $(x_z(v) - x(t))/v(t)$ TASC bölgesindeki zaman (sn)
t_c	: Son kademe değişikliğinden beri geçen zaman (sn)
N_c	: Son kademe değişikliğinin derecesi
N_p	: Seçilen kademe derecesi
$V_p(N_p)$: N_p kademesi seçildiğinde hesaplanan hız (km/sa)
V_e	: İzin verilen hızlanma aralığı (km/sa)
$X_p(N_p)$: N_p kademesi seçildiğinde hesaplanan duruş konumu (m)
x_e	: İzin verilen duruş aralığı (m)

1. GİRİŞ

Bulanık mantık (Fuzzy Logic) kavramı ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Dr. Lotfi A. Zadeh'in bu konu üzerinde makalelerini yayımlamasıyla duyuldu [16]. Yapılan araştırmalar ve uygulamalar ile gün geçtikçe önem kazanan bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş bir düzen olarak tanımlanabilir. Bilindiği gibi istatistikte ve olasılık kuramında, belirsizliklerle değil kesinliklerle çalışılır. Ama insanın yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışması gereklidir.

Bulanık mantığın uygulama alanları çok geniştir. Sağladığı en büyük fayda ise “insana özgü tecrübe ile öğrenme” olayının kolayca modellenebilmesi ve belirsizlikleri matematiksel olarak ifade edilebilmesine olanak tanınmasıdır. Bu nedenle özellikle lineer olmayan sistemlere yaklaşım yapabilmek için uygundur.

Bulanık mantık ilk kez 1973 yılında, Londra'daki Queen Mary College'de profesör olan H. Mamdani tarafından bir buhar makinesinde uygulanmıştır. Ticari olarak ise ilk defa 1980 yılında Danimarka'daki bir çimento fabrikasının fırını kontrol etmede kullanılmıştır.

Bundan sonra bir başka dikkate değer uygulama ise Hitachi firması tarafından 1987 yılında Sendai metrosunda gerçekleştirilmiştir. 14 km boyunca 16 istasyonda duran tren, duruş ve kalkış ivmelerinin en az hissedildiği toplu taşıma aracıdır ve bu sayede trenin istenen konumda durması üç kat daha iyileştirilmiş; kullanılan enerji ise %10 azaltılmıştır [8]. Sendai metrosunda yapılan uygulamanın temelleri, 1983 yılında Japonya Hitachi Sistem Geliştirme Laboratuvarında atılmıştır. Otomatik Tren Yönetim Sistemi (Automatic Train Operation - ATO) üzerinde çalışan araştırmacılar, sistem hakkında kural tabanı oluşturarak bulanık mantık ile tren kontrolü sağlamışlardır [16, 17, 18].

Otomatik Tren Yönetim Sistemi ile; güvenliğin artırılması, rahat ve konforlu bir yolculuk, gecikmelerin azaltılması, enerji ve yakıt tasarrufu sağlanır.

Bu çalışma, 8 bölümden oluşmaktadır. İlk bölüm giriş bölümüdür.

İkinci bölümde, Bulanık Küme Teorisinin temelleri tanıtılmış; üyelik fonksiyonları, operasyonlar, ilişkiler ve nedenlilik kavramları üzerinde durulmuştur.

Üçüncü bölümde, bulanık kümelerden yola çıkılarak bulanık mantık ve uygulama alanlarından bahsedilmiş, literatürde sunulan yaklaşımlar incelenmiştir.

Dördüncü bölümde, Bulanık Çıkarım Sisteminin bölümlerinden bahsedilmiş ve farklı üyelik fonksiyonlarının kullanıldığı bir örnek üzerinde incelenmiştir.

Beşinci bölümde, yapay sinir ağları ve öğrenme algoritmaları üzerinde durulmuştur. İleri Beslemeli Sinir Ağlarında Öğrenme ve Hatırlama, Geri Yayılım Algoritması, Hibrid (Online - Offline) Öğrenme algoritmaları tanıtılmıştır.

Altıncı bölümde, Adaptif ağ yapısına dayalı bulanık çıkarım sistemi tanıtılmış; mimarisi ve öğrenme algoritması üzerinde durulmuştur.

Yedinci bölümde, bulanık mantığın MATLAB ortamında kullanımı örnekler üzerinde tanıtılmıştır.

Sekizinci bölümde Otomatik Tren Yönetim Sistemi uygulama çalışması sunulmuştur. Otomatik Tren Yönetim Sisteminin genel işleyişinden bahsedilmiştir. Ardından tren hız ayarlama ve sabit hız kontrolü ile tren otomatik durma kontrolü şekillerle açıklanmış ve kuralları tanımlanmıştır. Otomatik Tren Yönetim Sistemi çalışma algoritması verilen diyagram ile birlikte ayrıntılı bir şekilde açıklanmıştır. Sistem birimleri ve ilişkileri güç ve fren bağlantı şeması ile tanıtılmıştır. Otomatik Tren Yönetim Sisteminin performans indeksleri olan; iz takip, konfor, güvenlik, işleme zamanı, duruş mesafesi incelenmiştir. Yukarıdaki bilgilerden yola çıkılarak MATLAB ortamında Bulanık Çıkarım Sistemi algoritması oluşturulmuş ve kaynak kodları verilmiştir. Algoritmanın simülasyonu yapılmış ve konum – hız grafikleri üzerinde sonuçları gösterilmiştir. Bunun yanı sıra MATLAB ortamında ANFIS yöntemi kullanılarak Otomatik Tren Yönetim Sisteminin güvenliği arttırılmaya çalışılmıştır. Makinistlerin alışkanlıklarını sisteme aktarmak için adaptif ağ yapısına dayalı bulanık çıkarım sistemi (Sugeno - ANFIS) kullanılmıştır [13].

Sonuç bölümünde ise uygulamanın sonuçları ele alınmış; Otomatik Tren Yönetim Sisteminin güvenliğini arttırıcı sonuçlara ulaşılmıştır.

2. BULANIK KÜMELER VE BULANIK MANTIK

Bu bölümde bulanık kümeler Bulanık Küme Kuramının Temel Kavramları, Bulanık Küme Üyelik Fonksiyonları, Bulanık Kümelerde Operasyonlar, Bulanık İlişkiler ve Bulanık Nedenlilik ifade edilecektir.

2.1. Giriş

Klasik mantık ile bulanık mantık arasındaki fark en açık şekilde kullandıkları küme teorileri incelenmekle görülebilir.

Klasik Doğrusal Programlama problemlerinde kesin sayılarla çalışılır ve bir belirlilik söz konusudur. Halbuki, pratikte modeli oluşturan parametrelere ilişkin net değerlere ulaşmak çoğu zaman mümkün olmamaktadır. Bu belirsizlik durumu, Zadeh (1965) tarafından önerilen “bulanık küme teorisi” ile dilsel olarak “az çok”, “biraz” gibi formlarda ifade edilebilir hale gelmiştir. Kesin olmayan bilginin sayısal gösterimi olan bulanık sayıların kullanılması ile gerçek dünya problemlerinin belirsiz yapısının modellenmesinde kullanılacak yeni matematiksel teknikler geliştirilmiştir. Bu belirsizlik içinde geliştirilen yöntemler, klasik doğrusal programlamadan hayli farklıdır.

Klasik ikili mantığın dayandığı standart küme teorisinde, herhangi bir nesne bir kümeye ya aittir ya da değildir. Bunun ortası yoktur. Yani bir nesnenin bir kümeye aitlik derecesini gösteren "üyelik derecesi" ya "birdir" ya da "sıfırdır". Bu nedenle klasik mantığın doğurduğu klasik kümeler, evrensel kümenin elemanlarını, kümeye ait olanlar ve ait olmayanlar diye ikiye böler. Yani klasik kümeler, tabiattakinin aksine, yaşadığımız dünyayı siyah/beyaz, doğru/yanlış, iyi/kötü gibi kategorize ederek ikiye bölen birbirine zıt ikili kavramlarla inşa eder. Halbuki, gerçek dünya hiç de öyle siyah ve beyazdan ibaret değildir; orada siyah ile beyazın arasında sonsuz renk tonu vardır. Konuşma dilinde ifade edilen ve üzerinde çalıştığımız çoğu sınıflandırmada kullanılan, genel olarak kesin sınırlarla tanımlanamayan ve kişiden kişiye farklı yorumlanan “güzel”, “uzun”, “sıcak”, “acı” gibi belirsiz kavramlara bulanık değişkenler denir. Günlük hayatta kullanırken sözel değişkenlerin etki derecesini “biraz”, “çok az”, “orta”, “çok”,

“çok fazla” gibi sıfatlarla artırır veya azaltırız. Bu tür sıfatlara bulanık değişkenlerin bulanık terimleri denir. Örneğin “acı” bulanık değişkeni “çok acı”, “orta acılıkta”, “acı” ve “tatlı” gibi bulanık terimlerle nitelenebilir. Bulanık terimlerle ifade edilen “Özlem çok güzel.”, “Hava aşırı sıcak.”, “Biber az acı.” gibi ifadeler, kesin hüküm belirtmediğinden, klasik mantık önerme olarak kabul etmez ve bu kavramlarla da klasik manada küme tanımlanamaz. İşte, bu tür önermelere bulanık önermeler ve bunlarla uğraşan mantığa da bulanık mantık denir [3].

Bulanık önermelerin doğruluğu ve yanlışlığı hakkında kesin bir şey söylenemeyeceğinden dolayı bunların doğruluk değeri $[0,1]$ gerçel sayılar kümesinden bir sayıyla derecelendirilir. Bir bulanık önerme derecesine göre hem doğru hem de yanlış olabilir. Bulanık bir önerme için “doğru değildir” denmiş ise bu “yanlıştır” anlamına gelmez. Bir önerme 0.8 derecesinde doğru ise aynı önerme 0.2 derecesinde de yanlıştır. Örneğin, “Özlem çirkindir” önermesi 0.5 derecesinde doğru ise aynı derecede de yanlıştır.

Bulanık önermeleri oluşturan bulanık terimlerin her biri bir bulanık küme ile modellenir. O halde bir bulanık önermenin oluşturduğu bir bulanık küme çalışma yapılan alana ait her bir elemana kümedeki aitlik derecesini temsil eden $[0,1]$ aralığındaki gerçel sayılardan bir değer atayarak tanımlanır. Bulanık küme kuramının temel kavramları devam eden başlıkta incelenecektir [1, 2, 3, 5].

2.2. Bulanık Küme Kuramının Temel Kavramları

Tanım 1: (Bulanık Küme) X evrensel tanım kümesi üzerinde A bulanık kümesi, X uzayından birim aralığa bir dönüşüm olan $\mu_A: X$ üyelik fonksiyonları ile tanımlanır:

$$A = \{(x, \mu_A(x)), x \in X\}, \forall x \in X \quad (2.1)$$

$$\mu_A: X \rightarrow [0,1].$$

A bulanık kümesindeki her eleman $x \in X$ ve üyelik fonksiyonu $\mu_A: X \in [0,1]$ ile gösterilir. Burada X kümesi evrensel kümedir ve $A \subseteq X$ olarak yazılabilir.

Bulanık küme kuramı, bir elemanın bir kümeye kısmi üyeliğine olanak sağlar. Eğer üyelik derecesi olarak adlandırılan üyelik fonksiyonunun değeri bire

eşitse x elemanı bulanık kümeye tamamen aittir. Eğer bu değer sıfır ise, x bulanık kümeye ait değildir. Eğer üyelik derecesi sıfır ile bir arasında ise x bulanık kümenin kısmi üyesidir.

2.3. Bulanık Küme Üyelik Fonksiyonları

2.3.1. Basit Üyelik Fonksiyonu

Basit üyelik fonksiyonu sadece evrensel kümenin \bar{x} noktasında 1 değerini alan, diğer noktalarda 0 olan üyelik fonksiyonu ile tanımlanır:

$$\mu_{Basit}(x) = \begin{cases} 1 & x = \bar{x} \\ 0 & x \neq \bar{x} \end{cases} \quad (2.2)$$

2.3.2. Gaussian Üyelik Fonksiyonu

Gaussian üyelik fonksiyonu genişlik ve merkez olmak üzere 2 parametreleri ile tanımlanır:

$$\mu_{Gauss}(x) = \exp\left(-\left(\frac{x-\bar{x}}{\sigma}\right)^2\right) \quad (2.3)$$

2.3.3. Genelleştirilmiş Gaussian Üyelik Fonksiyonu

Genelleştirilmiş çan üyelik fonksiyonu a genişlik, c merkez ve b eğim olmak üzere 3 parametre ile tanımlanır:

$$\mu_{Bell}(x) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \quad (2.4)$$

2.3.4. Sigmoidal Üyelik Fonksiyonu

Sigmoidal üyelik fonksiyonları $x=c$ kesişim noktasındaki a eğim parametresi ile tanımlanır:

$$\mu_{Sigm}(x) = \frac{1}{1+\exp[-a(x-c)]} \quad (2.5)$$

2.3.5. Üçgensel Üyelik Fonksiyonu

Üçgensel üyelik fonksiyonu $\{a, b, c\}$ parametreleri ile tanımlanır:

$$\mu_{\text{Üçgensel}}(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (2.6)$$

Diğer alternatif formül;

$$\mu_{\text{Üçgensel}}(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (2.7)$$

2.3.6. Yamuksal Üyelik Fonksiyonu

Yamuksal üyelik fonksiyonu $\{a, b, c, d\}$ parametreleri ile tanımlanır:

$$\mu_{\text{Yamuksal}}(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (2.8)$$

Diğer alternatif formül:

$$\mu_{\text{Yamuksal}}(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}\right), 0\right) \quad (2.9)$$

Tanım 2: (Yükseklik) A bulanık kümesinin yüksekliği, elemanlarının üyelik değerlerinin supremumudur.

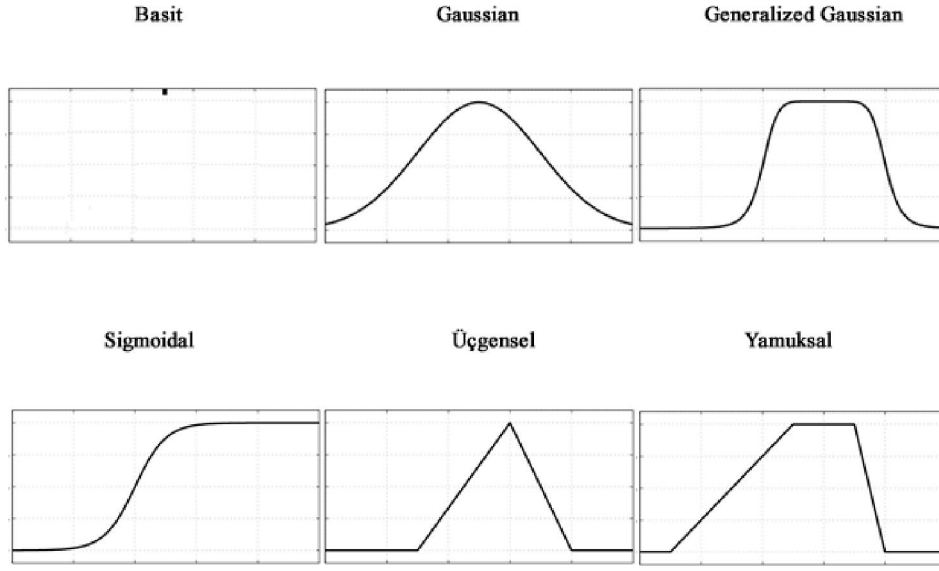
$$\text{Yükseklik}(A) = \sup_{x \in X} \mu_A(x) \quad (2.10)$$

Tanım 3: (Destek) X uzayında tanımlı A bulanık kümesinin desteği, üyelik derecesi sıfırdan farklı olan tüm elemanlarının oluşturduğu kesin alt kümedir.

$$\text{Destek}(A) = \{x \mid \mu_A(x) > 0\} \quad (2.11)$$

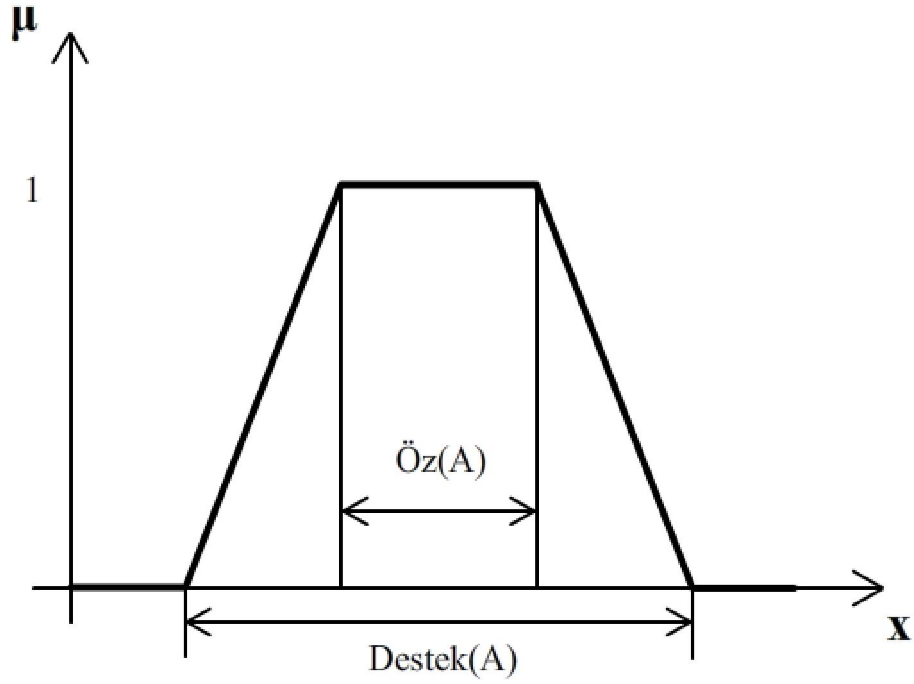
Tanım 4: (Öz) X uzayında tanımlı A bulanık kümesinin özü, üyelik derecesi bire eşit olan tüm elemanlarının oluşturduğu kesin alt kümedir.

$$\text{Öz}(A) = \{x \mid \mu_A(x) = 1\} \quad (2.12)$$



Şekil 2.1 Üyelik Fonksiyon Grafikleri

Literatürde, A bulanık kümesinin özü bazen göbek ya da $\ker(A)$ olarak gösterilir.



Şekil 2.2 Bulanık Kümelerde Destek ve Öz İlişkisi

Tanım 5: (Konveks bulanık küme) $x_1, x_2 \in X^n$ ve $\gamma \in [0,1]$ ise A bulanık kümesi konveksdir.

$$\mu_A(\gamma x_1 + (1 - \gamma)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\} \quad (2.13)$$

Tanım 6: (α -kesim) A bulanık kümesinin A_α ile gösterilen α -kesimi, X uzayında üyelik dereceleri α 'ya eşit veya büyük olan tüm elemanlarından oluşan kesin alt kümesidir.

$$A_\alpha = \{x | \mu_A(x) \geq \alpha, \alpha \in [0,1]\} \quad (2.14)$$

Ayrıca α -kesim operatörü, α -kesim(A, α) şeklinde de gösterilir.

Tanım 7: (Tam α -kesim) Her $x \in A_\alpha$ için $\mu_A(x) \neq \alpha$ ise A_α α -kesimi tamdır.

Tanım 8: A bulanık kümesini oluşturan A bulanık sayılar $X \subset R^N$ koşulunu sağlıyorsa A bulanık kümesi normal ve konveksdir.

Tanım 9: (Normal bulanık küme) Eğer $\exists x \in X$ için $\mu_A(x) = 1$ ise A bulanık kümesi normaldir. $norm(A)$ operatörü bir bulanık kümenin normalizasyonudur.

$$A' = norm(A) \leftrightarrow \mu_{A'}(x) = \frac{\mu_A(x)}{hgt(A)}, \forall x \quad (2.15)$$

Tanım 10: A bulanık kümesi simetrik ise sahip olduğu üyelik fonksiyonu da c noktasına göre simetriktir.

$$\mu_A(c + \Delta x) = \mu_A(c - \Delta x) \quad (2.16)$$

$$c + \Delta x \in X, c - \Delta x \in X$$

Tanım 11: (Kardinalite) $A = \{\mu_A(x_i)/x_i | i = 1, 2, \dots, n\}$ şeklindeki A bulanık kümesinin kardinalitesi üyelik derecelerinin toplamı olarak tanımlıdır:

$$|A| = \int_x \mu_A(x) dx$$

$$|A| = \sum_{i=1}^n \mu_A(x_i) \quad (2.17)$$

Tanım 12: A bulanık kümesinin genişlemesi (dilation) şu şekilde tanımlanır:

$$\mu_{DIL(A)}(x) = (\mu_A(x))^{0.5}, x \in X \quad (2.18)$$

Tanım 13: A bulanık kümesinin silindirik uzantısı $X \times Y$ bulanık kümesi şu şekilde tanımlanır:

$$ce(A, X \times Y) = \int_{X \times Y} \frac{\mu_A(x)}{(x,y)} \quad (2.19)$$

2.4. Bulanık Kümelerde Operasyonlar

Aşağıda bulanık kümelerde kesişim, birleşim ve ters operasyonlarına ilişkin temel tanımlar bulunmaktadır.

Tanım 14: (Bulanık kümelerin kesişimi) X uzayında A ve B iki bulanık küme olsun. A ve B kümelerinin kesişimi olan C bulanık kümesi $C = A \cap B$ şeklinde gösterilir öyle ki her $x \in X$ için:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.20)$$

Minimum operatörü ayrıca “ \wedge ” ile de gösterilir.

$$\mu_C(x) = \mu_A(x) \wedge \mu_B(x)$$

Tanım 15: (Bulanık kümelerin birleşimi) X uzayında A ve B iki bulanık küme olsun. A ve B kümelerinin kesişimi olan C bulanık kümesi $C = A \cup B$ şeklinde gösterilir öyle ki her $x \in X$ için:

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.21)$$

Maximum operatörü ayrıca “ \vee ” ile de gösterilir.

$$\mu_C(x) = \mu_A(x) \vee \mu_B(x)$$

İki bulanık kümenin bulanık kesişimi, birim aralıkta tanımlı ikili operasyon ile daha geniş olarak belirlenebilir. Örneğin; $T: [0,1] \times [0,1] \rightarrow [0,1]$ formundaki bir fonksiyon. T fonksiyonun bulanık kesişim olarak nitelendirilebilmesi için bazı özelliklere sahip olması gerekir. t -norm olarak bilinen fonksiyonlar kesişim için gerekli olan bu özelliklere sahiptirler.

Tanım 16: (t -norm / Bulanık kesişim) Bir t -norm T , her $a, b, c, d \in [0, 1]$ için en azından aşağıdaki koşulları sağlayan birim aralıkta tanımlı bir ikili operasyondur:

1) Sınır koşulu

$$T\{a, 0\} = 0, \quad T\{a, 1\} = a \quad (2.22)$$

2) Monotonluk

$$T\{a, c\} \leq T\{b, d\}, \quad a \leq b \text{ ve } c \leq d \quad (2.23)$$

3) Değişme Özelliği

$$T\{a, b\} = T\{b, a\} \quad (2.24)$$

4) Birleşme Özelliği

$$T\{T\{a, b\}, c\} \leq T\{a, T\{b, c\}\} \quad (2.25)$$

Benzer olarak a ve b parametreleri ile t-norm şu şekilde gösterilir:

$$T\{a, b\} = a^T * b \quad (2.26)$$

Birleşme özelliğini kullanarak $n > 2$ durumlarda Tanım 16 genişletilerek;

$$T\{a_1, a_2, \dots, a_n\} = T\{a\} = a_1^T * a_2^T * \dots * a_n^T \quad (2.27)$$

Benzer şekilde t-conorm olarak adlandırılan fonksiyonlar da bulanık birleşim için kullanılabilirler. $S: [0,1] \times [0,1] \rightarrow [0,1]$ formundaki fonksiyon için:

Tanım 17: (t-conorm / Bulanık birleşim) Bir t-conorm S, her $a, b, c, d \in [0, 1]$ için en azından aşağıdaki koşulları sağlayan birim aralıkta tanımlı bir ikili operasyondur:

1) Sınır koşulu

$$S\{a, 0\} = a, \quad S\{a, 1\} = 1 \quad (2.28)$$

2) Monotonluk

$$S\{a, c\} \leq S\{b, d\}, \quad a \leq b \text{ ve } c \leq d \quad (2.29)$$

3) Değişme Özelliği

$$S\{a, b\} = S\{b, a\} \quad (2.30)$$

4) Birleşme Özelliği

$$S\{S\{a, b\}, c\} \leq S\{a, S\{b, c\}\} \quad (2.31)$$

Benzer olarak a ve b parametreleri ile t-conorm şu şekilde gösterilir:

$$S\{a, b\} = a^S * b \quad (2.32)$$

Birleşme özelliğini kullanarak $n > 2$ durumlarda Tanım 17 genişletilerek;

$$S\{a_1, a_2, \dots, a_n\} = S\{a\} = a_1^S * a_2^S * \dots * a_n^S \quad (2.33)$$

Tanım 18: Bir t-conorm ve t-norm çifttir ve aşağıdaki koşulları sağlar:

$$S_{i=1}^n \{a_i\} = 1 - T_{i=1}^n \{1 - a_i\} \quad (2.34)$$

$$T_{i=1}^n \{a_i\} = 1 - S_{i=1}^n \{1 - a_i\} \quad (2.35)$$

Tanım 19: A ve B iki bulanık kümenin kesişimi t-norm ile ifadesi şu şekildedir:

$$\mu_{A \cap B}(x) = \mu_A(x)^T * \mu_B(x) \quad (2.36)$$

$A_1, A_2, \dots, A_n \subseteq X$ n tane bulanık kümenin $A = A_1 \cap A_2 \dots, A_n = \bigcap_{i=1}^n A_i$ kesişimi:

$$\mu_A(x) = T_{i=1}^n \mu_{A_i}(x) \quad (2.37)$$

Sıkça kullanılan t-normlar ise:

Standart kesişim: $T(a, b) = \min(a, b)$

(Cebirsel) çarpım: $T(a, b) = ab$

Lukasiewicz kesişimi: $T(a, b) = \max(0, a + b - 1)$

Tanım 20: A ve B iki bulanık kümenin birleşimi t-conorm ile ifadesi şu şekildedir:

$$\mu_{A \cup B}(x) = \mu_A(x)^S * \mu_B(x) \quad (2.38)$$

$A_1, A_2, \dots, A_n \subseteq X$ n tane bulanık kümenin $A = A_1 \cup A_2 \dots, A_n = \cup_{i=1}^n A_i$ birleşimi;

$$\mu_A(x) = S_{i=1}^n \mu_{A_i}(x) \quad (2.39)$$

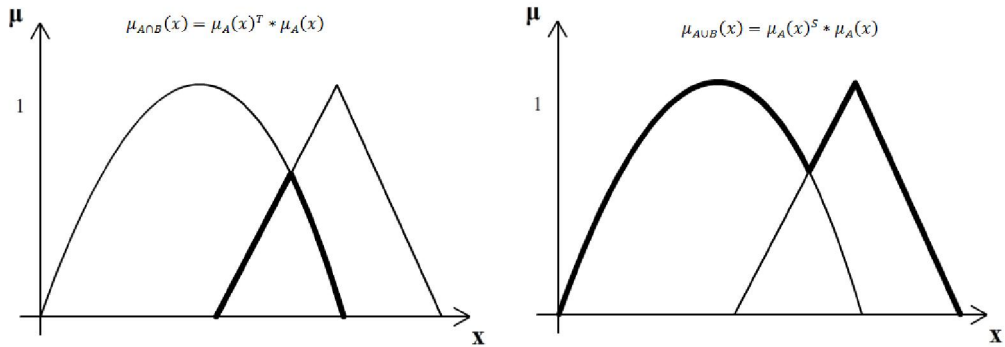
Sıkça kullanılan t-conormlar ise:

Standart birleşim: $S(a, b) = \max(a, b)$

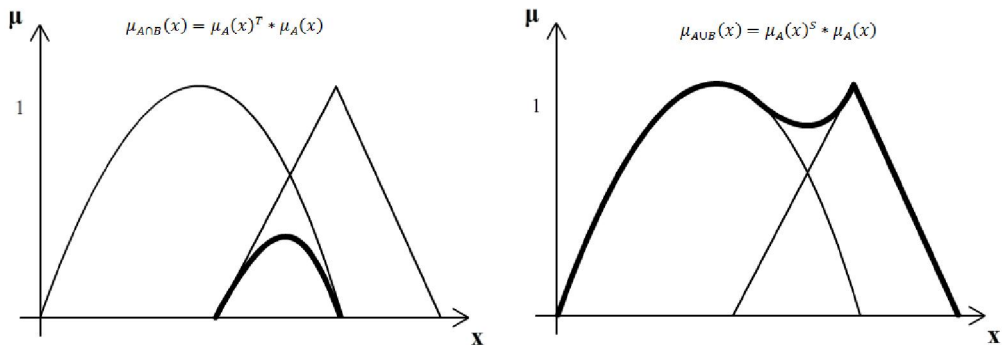
(Cebirsel) toplam: $S(a, b) = a + b - ab$

Lukasiewicz birleşimi: $S(a, b) = \min(1, a + b)$

Aşağıdaki şekillerde min/max ve cebir üçgensel normlar gösterilmiştir.



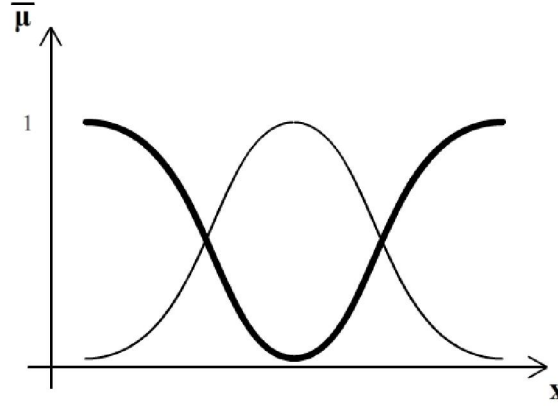
Şekil 2.3 Min/Max Norm Grafikleri



Şekil 2.4 Cebir Üçgensel Norm Grafikleri

Tanım 21: (Bir bulanık kümenin tersi) X uzayında A bir bulanık küme olsun. A bulanık kümesinin tersi \bar{A} ile gösterilir ve her $x \in X$ için aşağıdaki gibi tanımlıdır:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.40)$$



Şekil 2.5 Bulanık Kümenin Tersisi Grafiği

Tanım 22: A ve B iki bulanık kümenin kartezyen çarpımı $A \subseteq X, B \subseteq Y, A \times B$ şu şekilde tanımlanır:

$$\mu_{A \times B}(x, y) = \min \{ \mu_A(x), \mu_B(y) \} \quad (2.41)$$

yada,

$$\mu_{A \times B}(x, y) = \mu_A(x) \cdot \mu_B(y) \quad (2.42)$$

$x \in X, y \in Y$

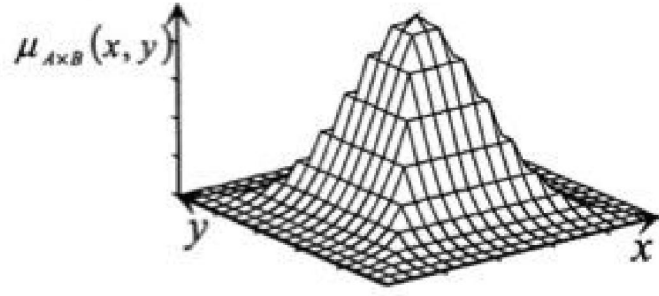
İkiden fazla bulanık kümenin kartezyen çarpımı $A_1 \subseteq X, A_2 \subseteq X, \dots, A_n \subseteq X, A_1 \times A_2 \times \dots \times A_n$ şu şekilde tanımlanır:

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, x_2, \dots, x_n) = \min \{ \mu_{A_i}(x_i) \} \quad (2.43)$$

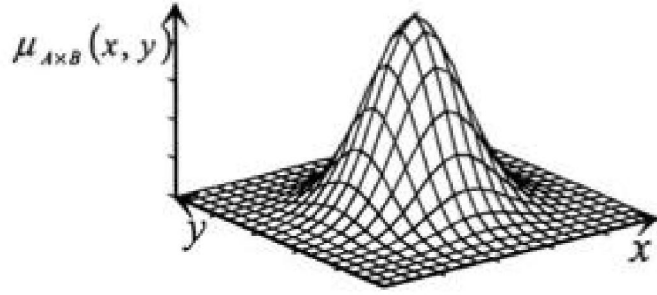
ya da,

$$\mu_{A_1 \times A_2 \times \dots \times A_n}(x_1, x_2, \dots, x_n) = \prod_{i=1}^n \mu_{A_i}(x_i) \quad (2.44)$$

Şekil 2.6 ve Şekil 2.7 minimum operatörü ve çarpım operatörü ile çizilen grafikler gösterilmiştir.



Şekil 2.6 Minimum Operatör Grafiği



Şekil 2.7 Çarpım Operatör Grafiği

Bulanık ilişkiler ve bulanık nedenlilik devam eden tanımlarda incelenecektir [1, 2, 3, 5].

2.5. Bulanık İlişkiler

Tanım 23: X ve Y iki evrensel küme olsun. İlişki R ile gösterilir ise aşağıdaki şekilde tanımlanır: Kesikli,

$$R = \sum_{X \times Y} \mu_R(y) / (x, y) \quad (2.45)$$

ya da sürekli,

$$R = \int_{X \times Y} \mu_R(y) / (x, y) \quad (2.46)$$

Tanım 24: R, $X \times Y$ evrensel kümesinde bulanık ilişki olsun. Y üzerinde R'nin üyelik derecesi aşağıdaki şekilde tanımlanır:

$$proj(R; Y) = \int_{y \in Y} \frac{\sup_{x \in X} \mu_R(x, y)}{y} \quad (2.47)$$

$\mu_R(x, y)$, (x, y) ikilisinin R üyelik derecesini ifade eder.

Tanım 25: R ve S, $X \times Y$ ve $Y \times Z$ evrensel kümelerinde bulanık komşular olsun. İlişki R'nin sup-T kompozisyonu aşağıdaki şekilde tanımlanır:

$$\mu_{R \times S}(x, z) = \sup_{y \in Y} \{\mu_R(x, y)^T * \mu_S(y, z)\}, \quad RoS \subseteq X \times Z \quad (2.48)$$

Tanım 26: İlişki R'nin sup-T kompozisyonu $A \subseteq X$ ve bulanık ilişki $R \subseteq X \times Y$ ise bulanık küme;

$$B = AoR \subseteq Y \quad (2.49)$$

$$\mu_B(y) = \mu_{AoR}(y) = \sup_{x \in X} \{\mu_A(x)^T * \mu_R(x, y)\} \quad (2.50)$$

Benzer olarak yukardaki tanımları doğrultusunda;

$$B = AoR = \text{proj}\{(ce(A; X \times Y) \cap R); Y\} \quad (2.51)$$

2.6. Bulanık Nedenlilik

Sonuç çıkarma kuralları, bulanık küme ve bulanık komşuluğun bir kompozisyonu olarak oluşturulur. Örneğin bir bulanık nedenlilik ya da kural,

$$\text{Eğer } x = A \text{ ise } y = B \quad (2.52)$$

ifadesi ile oluşturulur. A' dilsel giriş değeri, B' çıkış bulanık kümesi ve R ilişki olsun. Burada A, B, A', B' bulanık kümeler ve x, y dilsel değişkenlerdir.

B' çıkış bulanık kümesi;

$$B' = A'oR = A'o(A \rightarrow B) \quad (2.53)$$

ve,

$$\mu_{B'}(y) = \mu_{A'oR}(y) = \sup_{x \in X} \{\mu_{A'}(x)^T * \mu_R(x, y)\} \quad (2.54)$$

olarak tanımlanabilir. Bulanık komşuluğun üyelik fonksiyonlar ile tanımı;

$$\mu_R(x, y) = \mu_{A \rightarrow B}(x, y) \quad (2.55)$$

$$\mu_{A \rightarrow B}(x, y) = I(\mu_A(x), \mu_B(y)) \quad (2.56)$$

Burada, $I(\cdot)$ bulanık implikasyonu (fuzzy implication) olarak bilinir.

Tanım 27: $I: [0,1]^2 \rightarrow [0,1]$ fonksiyonunun bulanık implikasyonu aşağıdaki koşulları sağlar;

$$(K1) \text{ Eğer } a_1 \leq a_3, I(a_1, a_2) \geq I(a_3, a_2), \quad a_1, a_2, a_3 \in [0,1]$$

$$(K2) \text{ Eğer } a_2 \leq a_3, I(a_1, a_2) \leq I(a_1, a_3), \quad a_1, a_2, a_3 \in [0,1]$$

$$(K3) I(0, a_2) = 1, \quad a_2 \in [0,1]$$

$$(K4) I(a_1, 1) = 1, \quad a_1 \in [0,1]$$

$$(K5) I(1, 0) = 0$$

Yukardaki kurallara uyan bazı bulanık içerme fonksiyonları aşağıda listelenmiştir.

No	Ad	I(a,b)
1	Kleene-Dienes (Binary)	$\max\{1 - a, b\}$
2	Lukasiewicz	$\min\{1, 1 - a + b\}$
3	Reichenbach	$1 - a + a \cdot b$
4	Fodor	$\begin{cases} 1 & a \leq b \\ \max\{1 - a, b\} & a > b \end{cases}$
5	Rescher	$\begin{cases} 1 & a = 0 \\ 0 & a > 0 \end{cases}$
6	Goguen	$\begin{cases} 1 & a = 0 \\ \min\{1, b/a\} & a > 0 \end{cases}$
7	Gödel	$\begin{cases} 1 & a \leq b \\ b & a > b \end{cases}$
8	Yager	$\begin{cases} 1 & a = 0 \\ b^* & a > 0 \end{cases}$
9	Zadeh	$\max\{\min\{a, b\}, 1 - a\}$
10	Willmott	$\min\left\{\begin{array}{l} \max\{1 - a, b\} \\ \max\{a, 1 - b, \min\{1 - a, b\}\} \end{array}\right\}$
11	Dubois-Prade	$\begin{cases} 1 - a & b = 0 \\ b & a = 1 \\ 1 & \text{d. d} \end{cases}$
12	Mamdani (Max-Min)	$\min\{a, b\}$
13	Mamdani (Product)	$a \cdot b$

Tablo 2.1 Bulanık İçerme Fonksiyon Tablosu

2.7. Bulanık Mantık

Klasik mantık (Aristo Mantığı), kesin sonuçlar çıkarma üzerine kurulmuştur. Bu, birlerle ifade edilen, yanlış-doğru önermeler veya var-yok gibi kesin gerçeklerle yapılıdır. Örneğin "su iki hidrojen bir oksijen atomundan oluşur" önermesi kesindir ve doğruluk değeri birdir. Fakat oluşturacağımız önermeler her durumda bu kadar kesin sonuçlara götürmeyebilir. Benzer olarak, "su renksizdir" önermesi tam doğru olmamakla birlikte yanlış da değildir. Öyleyse insan beyni gerçek hayatta, karşılaştığı karar verme problemlerini nasıl çözer? İnsan beyni "koyu sarı", "sıcak su" veya "yüksek basınç" gibi matematiksel olarak kesin olmayan, belirsiz ya da değer yargıları içeren "bulanık dilsel niteleyicileri" kullanarak, sağduyulu kararlar verebilir ve sonuç çıkartabilir. Bulanık mantık, temel olarak yaklaşıklık ve kesin olmama gibi insan düşüncesinde ve doğada var olan belirsizliği kullanmaktadır ve kesin doğru veya kesin yanlış yerine dereceli doğruluk ya da yanlışlık belirten tanımlar kullanır. Böylelikle bulanık mantıkta, günlük konuşmalarda kullanılan belirsiz ifadeler tanımlanabilmekte ve bu dilsel niteleyiciler kullanılarak insan beyninin benzeri sonuç çıkarma işlemleri gerçekleştirilmektedir. Bu sayede bulanık mantık bilgisayarlara dolayısıyla kontrol sistemlerine, insan düşünme ve karar verebilme yeteneğinin kazandırılmasında başarılı bir tekniktir [8].

Bulanık mantık kavramı aslında yeni değildir. 1930'ların başlarında ilk üç-değerli mantık sistemini Lukasiewicz ortaya çıkarmıştır. Polonya'lı mantıkçı Jan Lukasiewicz, önermelerin sadece bir veya sıfır doğruluk değeri alabildiği klasik mantıktan farklı olarak, önermelerin bir ve sıfır arasında da kesirli doğruluk değeri alabileceği "çok değerli" mantık ilkelerini oluşturdu. Daha sonra kuantum felsefecisi Max Black yayımlanan bir makalesinde sürekli değerlere sahip mantığı tanımlamıştır. Belirsizlik kavramını, matematiksel olarak modelleyen ise Azeri kökenli Amerikalı Matematikçi Lütüf Askerzade Zadeh'tir. L.Zadeh makalesinde, bir kümenin tüm elemanlarına Lukasiewicz'in mantığını uygulayarak bulanık kümeler için eksiksiz bir cebir geliştirdi. Daha sonra 1965'de Lotfi A. Zadeh bu alana adını veren "Bulanık Kümeler" yazısını yayımladı [1]. L.Zadeh bu teorisinde dil ve insanın zeka ilişkisini matematiksel olarak modellemiş ve bulanık

mantığın gerçek problemlerin modellemesinde iyi bir teknik olduğunu göstermiştir. L.Zadeh için kuramda geçen bulanık sözcüğü matematiksel bir niceliği ifade eder ve gerçek dünyada cereyan eden olaylar kesikli değil sürekli olduğu görüşünü benimsemiştir. L.Zadeh'ye göre bulanık küme ve bulanık mantık özetle şunu ifade eder:

“Kesinlik diye bir şey yoktur. Mutlak kesin olan hiçbir şey yoktur. Her şey matematiksel olarak ifade edersek, 0 ile 1 arasındaki sınırdadır değişmektedir.“ [6].

Bulanık mantığı ilk olarak 1972 yılında İngiltere’de İran kökenli Ebrahim Mamdani’nin bir buhar makinesi üzerinde uygulamıştır. İlk ticari uygulaması ise, 1980’de, Danimarka’da bir çimento fabrikasının entegrasyonunda kullanılmıştır.

Bulanık mantığın uygulama alanları kontrol sistemleriyle sınırlı değildir. Geliştirilen bulanık mantık ile mühendislik, tıp ve ekonomi gibi farklı alanlarda sürekli sistemleri modellemek için kullanılmış ve standart matematiksel modellerden daha verimli ve kesin sonuçlar ürettiği görülmüştür .

Bulanık Mantık, insanın karar verme ve değerlendirme süreçlerini algoritmik biçimde temsil ederken, insan mantığının esnekliğini örnek olarak problemlerin çözümünü mümkün kılmaktadır. Bulanık Mantık, makinelere zeka kazandırmıştır. İnsanoğlu bunu kullanarak güvenliği ve performansı artırıcı ona karşı işletme ve bakım maliyetini azaltıcı, iş gücü ve zaman tasarrufu sağlayan sistemler geliştirmiştir. Örnek olarak bulanık mantık günümüzde;

- ✓ Metroların sinyalizasyonunda,
- ✓ Televizyon alıcılarında,
- ✓ Bilgisayar disklerinin okuma kontrol birimlerinde,
- ✓ Klima, çamaşır makineleri, elektrikli süpürgelerde,
- ✓ Buzdolabı klimalandırma sistemlerinde,
- ✓ Trafik ışıklarında,
- ✓ Otomobillerin ABS fren sistemleri, motorları ve süspansiyonlarında,
- ✓ Asansörlerde,
- ✓ GPS navigasyon yön bulma yazılımlarında,
- ✓ Füze atış kontrol sistemlerinde,

- ✓ Robotlarda,
- ✓ Görüntü işleme ve kameralarda uygulamalarını görmekteyiz.

Bulanık mantık ve bunun uygulaması olarak “Bulanık Yapay Zeka” gelecekte insanlar ile bilgisayarlar arasında kurulacak olan yakın ilişkide büyük bir rol oynayacağı beklenmektedir.

Bulanık mantık kuramı, bulanık kümeler üzerine inşa edilmiştir. Daha önce bahsettiğimiz gibi klasik mantık küme elemanlarını evrensel kümeye ait olan ve olmayan diye ayırır. Bunun tersine bulanık mantık küme içinde var-yok çıkarımından daha çok ait olmayı ya da olmamayı $[0,1]$ gerçel sayılar kümesinden derecesi ile ifade eder.

Konuşma dilinde kullandığımız “tatlı”, ”çirkin” ve “kısa” gibi sınırları tanımlanamayan kavramlara *bulanık değişken* denir. Bu değişkenlerin etki derecelerini pekiştirdiğimiz “çok fazla”, “fazla”, “orta”, “az”, “çok az” gibi sıfatlar ise *bulanık terimler* olarak adlandırılır. “Özlem çok güzel”, “Yemek çok tuzlu” gibi ifadeler ise *bulanık önerme* ve uygulanılan mantığa ise *bulanık mantık* adı verilir. Bulanık önermelerin içindeki bulanık terimlerin her biri bulanık kümeye aittir. Matematiksel olarak önceki bulanık küme konusunda işlendiği gibi bulanık önermeler, X evrensel tanım kümesindeki bir bulanık A kümesinin $\mu_A: X \rightarrow [0,1]$ şeklinde haritalanmasıdır. Bulanık A kümesi, X evrensel kümesinde bulunan her elemanı ve üyelik derecesi ile birlikte oluşturduğu ikililer kümesidir. Şu şekilde ifade edilir:

$$A = \{(x, \mu_A(x)), x \in X\}, \forall x \in X \quad (2.57)$$

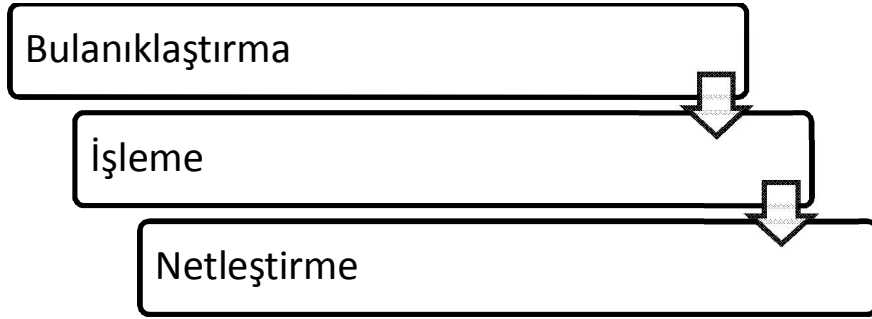
Farklı olarak, bir bulanık kümenin elemanlarının üyelik dereceleri toplamı, olasılıktaki gibi, 1 etmek zorunda değildir. Bir bulanık kümenin üyelik dereceleri toplamına bu kümenin kardinalitesi denir. Bir elemanın bir A bulanık kümesi ile bunun tümleyeni bulanık kümesine aitlik dereceleri toplamı 1 dir. Bu nedenle, bir bulanık A kümesinin tümleyeni $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ ile ifade edilir [9].

Bir bulanık kümesi kendi aitlik fonksiyonu ile açık olarak temsil edilebilir. Aitlik fonksiyonu 0 ile 1 arasındaki her değeri alabilir. Böyle bir aitlik fonksiyonu ile “kesinlikle ait” veya “kesinlikle ait değil” arasında istenilen değer verilebilmektedir.

Bir bulanık önerme derecesine göre hem doğru ve hem de yanlış olabilir. Bulanık bir önerme için “doğru değildir” denmiş ise bu “yanlıştır” anlamına gelmez. Bir önerme 0.2 derecesinde doğru ise aynı önerme 0.8 derecesinde de yanlıştır. Örneğin, “Özlem çirkindir” önermesi 0.5 derecesinde doğru ise aynı derecede de yanlıştır. Bu özellikten dolayı, klasik mantıkta problem olan paradokslar, hem “doğru” hem “yanlış”, ya da ne “doğru” ne de “yanlış” doğruluk değerine sahip önermeler, bulanık mantıkta doğruluk değerleri olarak biraz da olsa doğrulara indirgenmiş olurlar.

Bir bulanık mantığın işleyiş aşamalarını şu şekilde özetleyebiliriz [7]:

1. Bulanıklaştırma (Fuzzification)
2. Davranış Tanımlama
3. Netleştirme, Durulaştırma veya Bulanıklığın Giderilmesi (Defuzzification)

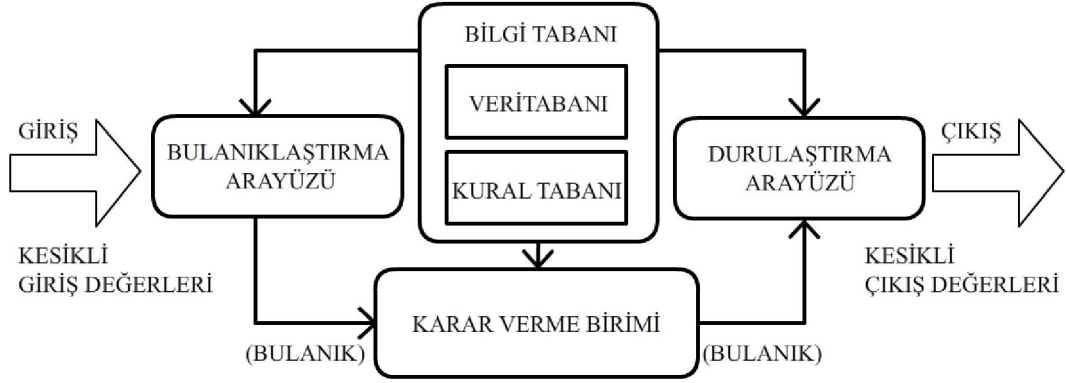


Şekil 2.8 Bulanık Mantık İşleyiş Aşamaları

İlk iki evre bulanık küme tanımları ve işlemlerini içerir. Bulanıklaştırma bulanık kümelerin genelleştirilmesidir. İşleme adımında kümeler arası özel bağlantılar kurulur. Çözüm için tek bir değer elde edilmesi netleştirme aşamasında yapılır.

3. BULANIK ÇIKARIM SİSTEMİ

Bulanık çıkarım sistemi, bulanık küme kavramı ve buna bağlı bulanık mantığın temelleri üzerine kurulmuş problemlerin çözümüne yönelik geliştirilen sistemlerdir. Bulanık çıkarım sistemini oluşturan bölümler şu şekildedir [12]:



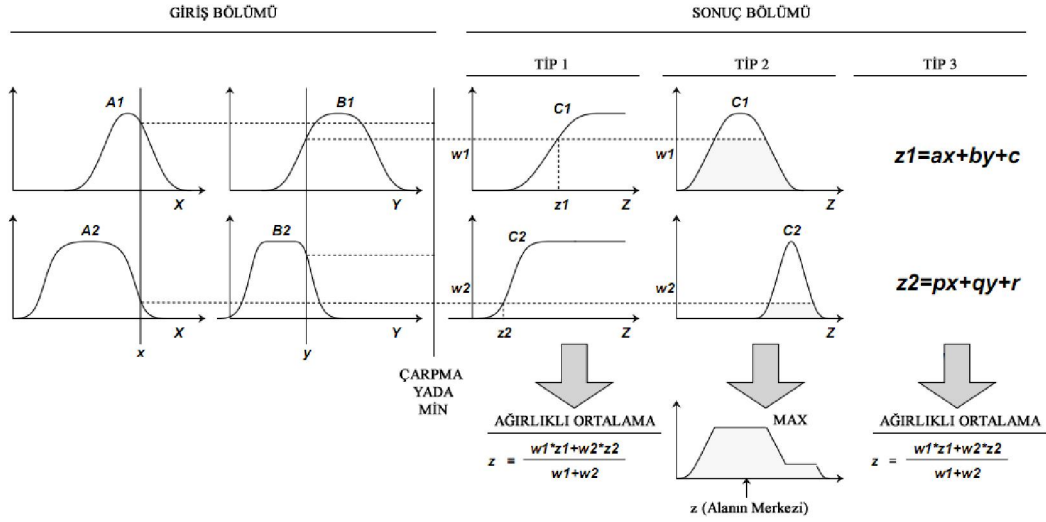
Şekil 3.1 Bulanık Çıkarım Sistem Mimarisi

3.1. Bulanık Çıkarım Sisteminin Katmanları

Bulanık çıkarım sistemi beş bloktan oluşmaktadır [12].

- ✓ **Kural Tabanı:** Bulanık neden-sonuç kurallarını içerir,
- ✓ **Veritabanı:** Bulanık kurallar içerisindeki bulanık kümelerin üyelik fonksiyonlarını tanımlar,
- ✓ **Karar Verme Birimi:** Kurallar üzerindeki çıkarım operasyonlarını sağlar,
- ✓ **Bulanıklaştırma Arayüzü:** Giriş değerlerinin dilsel değişken dereceleri ile karşılaştırıp bulanıklaştırma arayüzüdür,
- ✓ **Durulaştırma Arayüzü:** Çıkarım sonuçlarını kesikli değerlere döndüren arayüzdür.

Bu bölümde, bulanık çıkarım sistemini iki girişli ve tek çıkışlı yapı ile incelenecektir. Bulanık çıkarım sistemimin gerçekleştirdiği bulanık nedensellik adımları aşağıdaki diyagramda sunulmuştur [12]:



Şekil 3.2 Bulanık Nedensellik Adımları

Bulanık çıkarım sistemi giriş ve sonuç bölümleri altında incelenilirse, A ve B bulanık kümeleri ait oldukları $\mu_{A1}(x)$, $\mu_{A2}(x)$, $\mu_{B1}(y)$, $\mu_{B2}(y)$ üyelik fonksiyonlarındaki aitlik derecelerini; T-norm (Mamdani çarpma, min ve diğer) operatör kullanılarak sonuç bölümüne aktarılır. Giriş bölümünde yapılan bu dönüşüm bulanıklaştırma olarak adlandırılır. Yukarıdaki şekilde görüleceği gibi min operatör kullanılırsa x kesikli değişkeni $\mu_{A2}(x)$ üyelik fonksiyonu ile , y kesikli değişken $\mu_{B1}(y)$ üyelik fonksiyonu ile bulanıklaştırılmış ve sonuç bölümünde w1 ve w2 ağırlıkları olarak girdi sağlanmıştır. Sonuç bölümünde genelde üç farklı tipden birisi kullanılarak sonuç elde edilir. Bu aşamaya durulaştırma adı verilmektedir.

Tip1: z çıkış değişkeninin üyelik fonksiyonları $\mu_{C1}(z)$, $\mu_{C2}(z)$ olsun. Giriş bölümünden gelen w1 ve w2 ağırlıkların üyelik fonksiyonlarının aitlik dereceleri ağırlıklı ortalaması alınarak z çıkış değeri elde edilir.

Tip2: z çıkış değişkeninin üyelik fonksiyonları $\mu_{C1}(z)$, $\mu_{C2}(z)$ olsun. Giriş bölümünden gelen w1 ve w2 ağırlıkların üyelik fonksiyonlarından elde edilen aitlik derecelerinin maksimumu çıkış değeri olur.

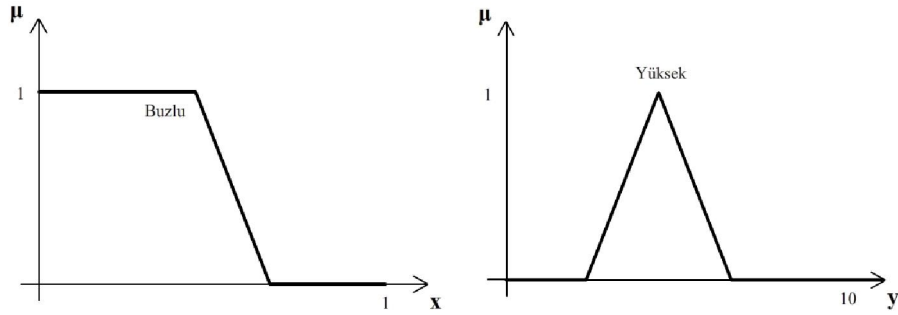
Tip3: z1 ve z2 çıkış değişkenleri giriş değişkenlerinin belli parametrelerle oluşturduğu lineer denklemler olsun. Bu çıkış değişkenlerinin giriş bölümünden gelen w1 ve w2 parametreleri ile ağırlık ortalamaları çıkış değeri olarak elde edilir.

3.2. Bulanık Çıkartım Sisteminin Bir Örnek Üzerinde İncelenmesi

Bir girişli bir çıkışlı bulanık çıkarım sistemi için x kesikli giriş değişkeni buzlu, y kesikli çıkış değişkeni yüksek olarak tanımlansın. x değişkeninin üyelik fonksiyonu $\mu_{Buzlu}(x)$, y değişkeninin üyelik fonksiyonu $\mu_{Yüksek}(y)$ ise t-norm min operatörünün kullanıldığı Mamdani implikasyonu,

$$\mu_R(x, y) = \min \{ \mu_{Buzlu}(x), \mu_{Yüksek}(y) \} \quad (3.1)$$

şeklinde ifade edilir. x ve y değişkenleri için üyelik fonksiyonları Şekil 11 gibidir.



Şekil 3.3 X, Y Değişkenlerinin Üyelik Fonksiyonları

Bulanık kural EĞER x=Buzlu ise y=Yüksek olarak tanımlansın.

$$x = \{0.0, 0.2, 0.4, 0.6, 0.8\}$$

$$y = \{1.0, 2.0, 3.0, 4.0, 5.0\}$$

x ve y kümelerinin ilişki matrisi ;

R / x - μ	1.0 - 0	2.0 - 0.5	3.0 - 1	4.0 - 0.5	5.0 - 0
0.0 - 1	0	0.5	1	0.5	0
0.2 - 1	0	0.5	1	0.5	0
0.4 - 1	0	0.5	1	0.5	0
0.6 - 0.5	0	0.5	0.5	0.5	0
0.8 - 0	0	0	0	0	0

Tablo 3.1 X, Y kümelerinin İlişki Matrisi

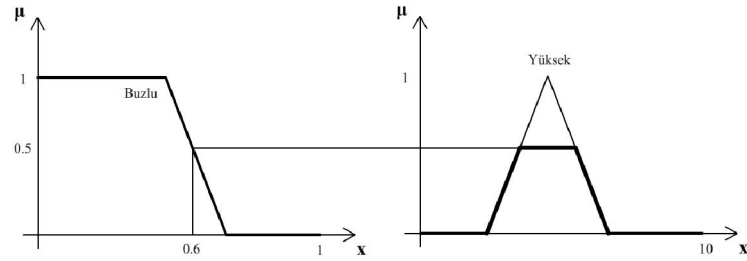
x kesikli deęişkenin 0.6 deęeri için t-norm min operatörü ile ilişki matrisinden çıkartımı şu şekilde tanımlanır.

$$\mu_R(x, y) = \min\{\mu_{Buzlu}(0.6), \mu_{Yüksek}(y)\} = \min\{0.5, \mu_{Yüksek}(y)\} \quad (3.2)$$

R / x - μ	1.0 - 0	2.0 - 0.5	3.0 - 1	4.0 - 0.5	5.0 - 0
0.0 - 1	0	0.5	1	0.5	0
0.2 - 1	0	0.5	1	0.5	0
0.4 - 1	0	0.5	1	0.5	0
0.6 - 0.5	0	0.5	0.5	0.5	0
0.8 - 0	0	0	0	0	0

Tablo 3.2 X Kesikli Deęişkenin Min Operatörü ile İlişki Matrisi

x kesikli deęişkenin 0.6 deęerindeki grafiksel gösterimi aşağıdaki gibidir:



Şekil 3.4 Kesikli Deęişkenin min Operatörü ile Üyelik Fonksiyonları

Benzer olarak t-norm operatörlerinden çarpım (product) kullanılarak elde edilen ilişki matrisi aşağıdaki gibidir:

$$\mu_R(x, y) = \mu_{Buzlu}(x) \cdot \mu_{Yüksek}(y) \quad (3.3)$$

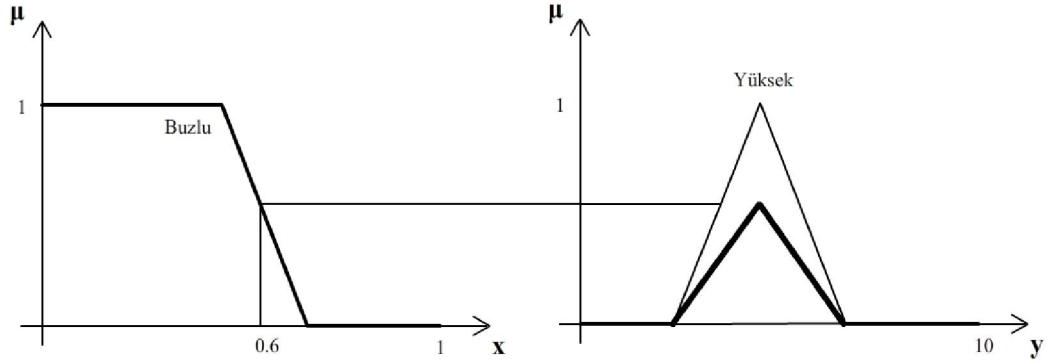
R / x - μ	1.0 - 0	2.0 - 0.5	3.0 - 1	4.0 - 0.5	5.0 - 0
0.0 - 1	0	0.5	1	0.5	0
0.2 - 1	0	0.5	1	0.5	0
0.4 - 1	0	0.5	1	0.5	0
0.6 - 0.5	0	0.25	0.5	0.25	0
0.8 - 0	0	0	0	0	0

Tablo 3.3 Kesikli Deęişkenin Çarpım Operatörü ile İlişki Matrisi

x kesikli deęişkeninin 0.6 deęeri için t-norm çarpım operatörü ile ilişki matrisinden çıkartımı şu şekilde tanımlanır.

$$\mu_R(x, y) = \mu_{Buzlu}(0.6) \cdot \mu_{Yüksek}(y) \quad (3.4)$$

Sonuç kümesi = (0, 0.25, 0.5, 0.25, 0) olarak elde edilir.



Şekil 3.5 X Kesikli Deęişkeninin Çarpım Operatörü ile Üyelik Fonksiyonları

Benzer olarak t-norm operatörlerinden Lukasiewicz kullanılarak elde edilen ilişki matrisi aşağıdaki gibidir:

$$\mu_R(x, y) = \max \{0, \mu_{Buzlu}(x) + \mu_{Yüksek}(y) - 1\} \quad (3.5)$$

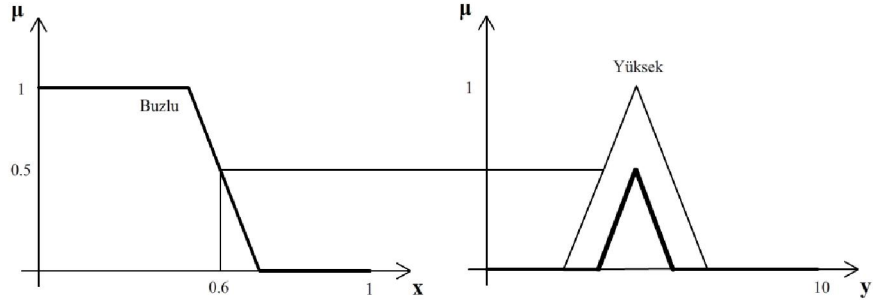
R / x - μ	1.0 - 0	2.0 - 0.5	3.0 - 1	4.0 - 0.5	5.0 - 0
0.0 - 1	0	0.5	1	0.5	0
0.2 - 1	0	0.5	1	0.5	0
0.4 - 1	0	0.5	1	0.5	0
0.6 - 0.5	0	0	0.5	0	0
0.8 - 0	0	0	0	0	0

Tablo 3.4 Lukasiewicz Operatörü ile Aldığı Deęerler

x kesikli deęişkeninin 0.6 deęeri için t-norm çarpım operatörü ile ilişki matrisinden çıkartımı şu şekilde tanımlanır.

$$\mu_R(x, y) = \max \{0, \mu_{Buzlu}(0.6) + \mu_{Yüksek}(y) - 1\} \quad (3.6)$$

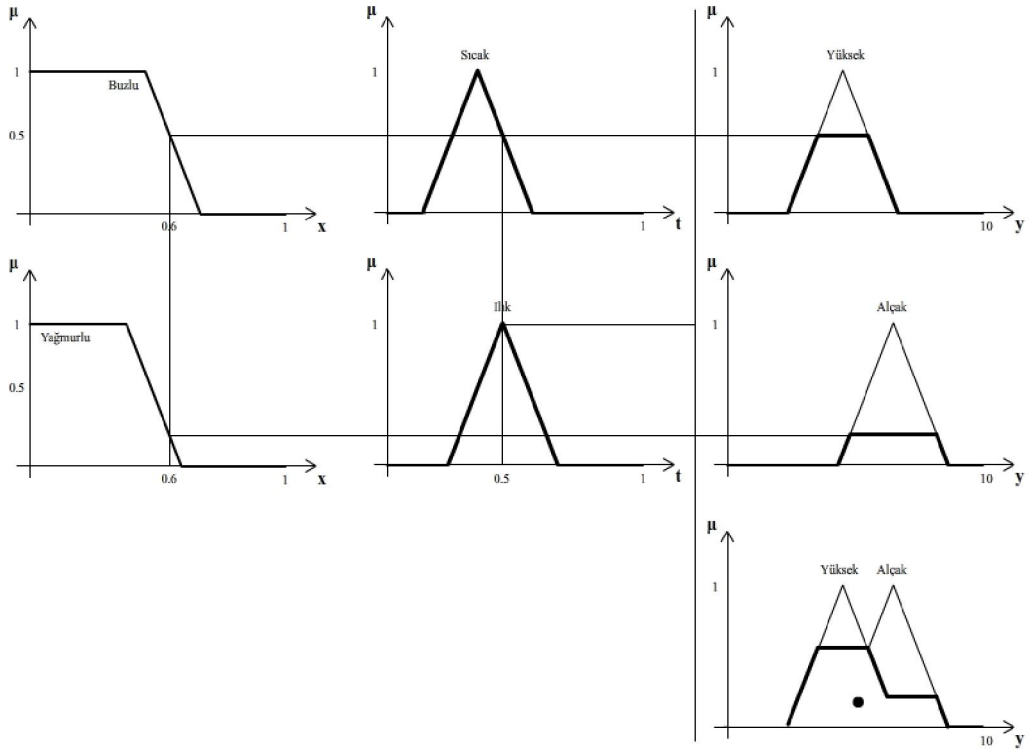
Sonuç kümesi = (0, 0, 0.5, 0, 0) olarak elde edilir.



Şekil 3.6 Lukasiewicz Operatörü ile Üyelik Fonksiyonları

İki girişli bir çıkışlı bulanık çıkarım sistemi için yukardaki örneğe ek olarak t kesikli giriş değişkeni Sıcak olarak tanımlansın. t değişkeninin üyelik fonksiyonu $\mu_{Sıcak}(t)$ olsun.

X değişkeninin 0.6 değerine göre iki üyelik fonksiyon değerleri Mamdani'nin Max-Min implikasyonuna göre karşılaştırılırsa çıkış değişkenin alçak üyelik fonksiyonu seçilir. Aynı şekilde t değişkeninin 0.5 değeri için min olan çıkış değişkeninin yüksek üyelik fonksiyonu seçilir. İki üyelik fonksiyonu max metoduna göre birleştirildiğinde aşağıdaki şekilde görülen birleşim kümesi elde edilmiş olur. Bu noktadan sonra durulaştırma işlemine başlanır.

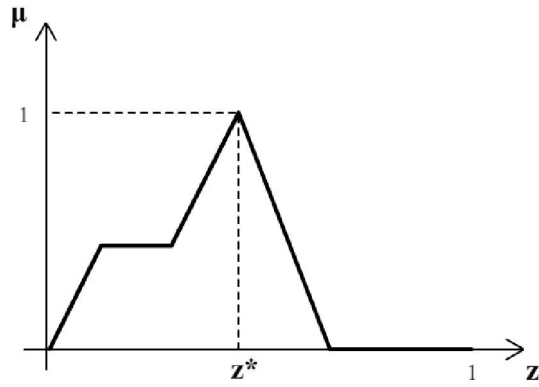


Şekil 3.7 Örnek Durulaştırma İşlemi

3.3. Durulandırma Yöntemi

3.3.1. Maksimum değeri alma

Bulanık küme içinde en büyük değere sahip eleman “kesin değer” olarak alınır. Fakat birden fazla maksimum değeri olması halinde bu algoritma ile karar vermek oldukça güçleşir. Ayrıca konveks olmayan bulanık kümelerde yanlış sonuçlar çıkabilmektedir. $\mu_A(z^*) \geq \mu_A(z)$ koşuluna göre;

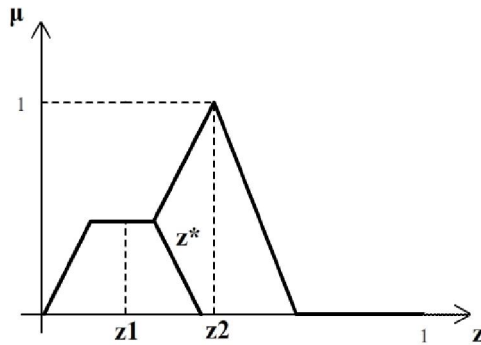


Şekil 3.8 Maksimum Üyelik Metodu Grafiği

3.3.2. Maksimumların Ortalamasını Alma

Birden fazla maksimum noktası olması halinde çözüm üreten bir yöntemdir. Maksimum değerlerin ortalaması “kesin değer” olarak alınır. Bu netleştirme metodunda;

$$z^* = \sum_{j=1}^k \frac{z_j}{k} \quad (3.7)$$



Şekil 3.9 Maksimumların Ortalamasını Alma Metodu Çıkış Değeri

3.3.3. Alan Merkezi Metodu

Alan merkezi ya da ağırlık merkezi de denilen bu yöntem en çok kullanılan netleştirme yöntemlerinden biridir ve ağırlık merkezi hesaplanarak yapılmaktadır.

$$z^* = \frac{\sum_{j=1}^k \mu_z(w_j) w_j}{\sum_{j=1}^k \mu_z(w_j)} \quad (3.8)$$

Burada, aktif olan kuralların bulanık çıkışlarına ilişkin üyelik fonksiyon değerleri ile skaler ağırlıkları çarpılarak toplamları alınır. Elde edilen değer, üyelik fonksiyon değerlerinin toplamına bölünmesiyle sayısal denetim işareti bulunur [12, 14].

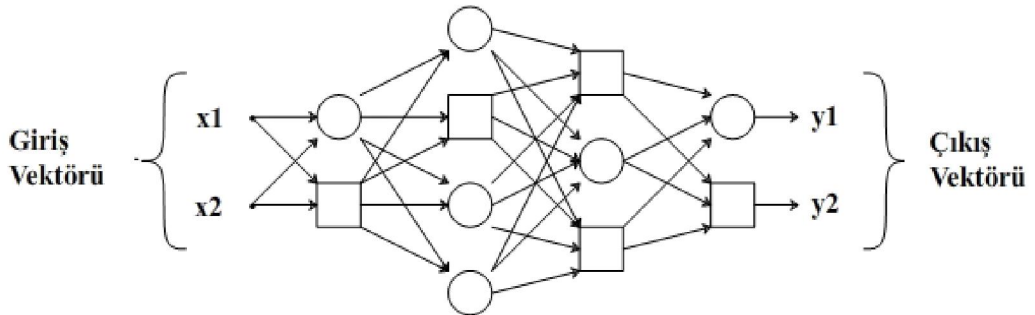
4. ADAPTİF AĞ YAPISINA DAYALI BULANIK ÇIKARIM SİSTEMİ

Önceki konularda bulanık küme , bulanık mantık, bulanık çıkarım, yapay sinir ağları hakkında sistem mimarileri ve çalışma mantıkları üzerinde durduk. Adaptif ağ yapısına dayalı bulanık çıkarım sistemi (ANFIS) ise bu bilgi ve tekniklerin üzerinde kurulu, bulanık mantık ve yapay sinir ağının oluşturduğu sistemdir. Eğitim olarak hibrid öğrenme algoritmasını kullanmaktadır [13].

4.1. Yapay Sinir Ağları

Yapay sinir ağları (YSA) insan beynindeki sinir sistemine benzer işleyişi taklit ederek karmaşık fonksiyonlara yakınsayabilmektedir. YSA sistem verileri ile eğitilerek öğrenme yeteneği kazanmakta ve karmaşık sistemlerin modellenmesinde oldukça önemli bir araç haline gelmektedir. [10, 11, 12].

İleri beslemeli sinir ağları (Feed Forward Network-FFN) yapay sinir ağları arasında yaygın olarak tercih edilen tekniklerden birdir. FFN Şekil 18’de gibi bir veya daha fazla gizli katmandan oluşan ileri beslemeli sinir ağlarıdır. Gizli katmanlar giriş ve çıkış katmanları arasında bağlantılar oluşturur ve gerekli işlemleri yerine getirir. Çok katmanlı yapılarda a1. katmanın çıkış sinyalleri a2. katmanın giriş sinyalleri olarak kullanılır. m adet giriş düğümü, ilk gizli katmanında t1 adet nöron, ikinci gizli katmanında t2 adet nöron ve çıkış katmanında k adet nöron bulunan çok katmanlı ileri besleme sinir ağı m-t1-t2-k ağı olarak adlandırılır.



Şekil 4.1 Çok Katmanlı Yapay Sinir Ağı

4.1.1. Sinir ağlarında öğrenme ve Geri Yayılım Algoritması

Sinir Ağlarının gerçekleştirdiği iki temel fonksiyon, öğrenme ve hatırlamadır. Öğrenme, ağırlık değerlerinin, bir giriş vektörüne karşılık istenilen çıkış vektörünü sağlamak üzere uyarlanmasıdır. Benzer olarak belirli bir girişe, ağırlık değerlerine uygun bir çıkış üretmesi de hatırlama olarak tanımlanır.

İleri beslemeli sinir ağlarının giriş-çıkış değerlerinin ne kadar sağlandığının bir ölçüsü olarak, ağırlık çıkış katmanındaki her sinire ait hata sinyallerinin karelerinin toplamından oluşan bir uygunluk fonksiyonu oluşturulur. Eğitimin k 'inci tekrarında ağırlık çıkış katmanındaki i 'inci sinirinin çıkış değeri y_i , bu sinirden vermesi istenilen değer d_i ile gösterilirse, i . sinirinin hata değeri:

$$e_i = d_i - y_i(k) \quad (4.1)$$

denklemini elde edilir. Uygunluk fonksiyonu da;

$$E = \frac{1}{2} \sum_i e_i^2(k) = \frac{1}{2} \sum_i (d_i - y_i(k))^2 \quad (4.2)$$

denklemini ile ifade edilir. Geri yayılım algoritmasının amacı uygunluk fonksiyonunu minimum yapmaktır. Uygunluk fonksiyonu ağırlık değerlerine bağlı olduğundan, algoritma ağırlıklarının en uygun biçimde değiştirilmesi işlemlerinden oluşmaktadır. YSA'daki her w_{ij} ağırlık değerinin değişme miktarı gradyan-alçalma (gradient-descend) yöntemi olarak adlandırılan

$$\Delta w_{ij} = -\eta \frac{\partial E(\vec{w})}{\partial w_{ij}} \quad (4.3)$$

denkleminde yola çıkarak bulunmaktadır [12]. Burada η öğrenme katsayısıdır.

$$\eta = \frac{k}{\sqrt{\sum \left(\frac{\partial E}{\partial w_{ij}} \right)^2}} \quad (4.4)$$

k parametresi adım büyüklüğüdür. k parametresinin değişimi sonuca yakınsamanın hızını değiştirir. Tüm gizli katman nöronları için değişiklik yapıldıktan sonra yeni girişler uygulanır ve süreç yeniden başlar. Hedeflenen hata kriterine ulaşılan dek iterasyon devam eder. Hata kriterine ulaşıldığında eğitim tamamlanmış olur.

4.1.2. Hibrid Öğrenme (Off-Line)

Gradyan-Düşümü yöntemi sinir ağlarında sık kullanılan bir tekniktir. Ancak genelde yavaş çalışmaktadır ve yerel minimumlara takılmaktadır. Bundan dolayı hibrid öğrenme algoritmaları geliştirilmiştir. Bunlardan gradyan yöntemi ve en küçük kareler yönteminin birleşimi olan offline öğrenme algoritmasıdır [13].

Basit olarak ağın bir çıkışı olsun;

$$\text{Çıkış} = F(\vec{I}, S), \quad (4.5)$$

Burada \vec{I} giriş değişkenlerinin kümesini, S parametrelerin kümesini gösterebilir. Hibrid öğrenme algoritmasında, sinir ağına ait parametreler, giriş ve çıkış parametreleri olarak iki kısımda ele alınır. Toplam parametre kümesi; $S = S1 + S2$ şeklinde ifade edilecek olursa, $S1$ giriş parametrelerine, $S2$ de çıkış parametrelerine karşılık gelir. Hibrid algoritmanın ilk aşaması ağa ait giriş parametreleri, ikinci aşamada ise ağın sonuç parametreleri güncellenir. Hibrid öğrenme algoritmasının ileri yön geçişini oluşturan ve en küçük kareler tahmini (LSE) yöntemi kullanılmak suretiyle gerçekleştirilen kısmında, ağın girişindeki üyelik fonksiyonlarına ait parametreler ya da kısaca $S1$ ile belirtilen giriş parametreleri sabit tutulur. Böylece ağın çıkışı, $S2$ parametre kümesi içinde bulunan çıkış parametrelerinin, lineer bir kombinasyonu haline gelir. Modellenen sisteme ait P tane giriş-çıkış verisi ya da eğitim verisi ile S giriş parametrelerinin verilen değerleri kullanılarak, ağın çıkışı:

$$A\theta = B \quad (4.6)$$

şeklinde ifade edilebilir. Matrisel eşitlikte yer alan θ vektörü, $S2$ çıkış parametrelerinin elemanlarından oluşan, bilinmeyen vektördür. Bu denklem, standart lineer en küçük-kareler (least-squares) problemini gösterir ve θ için en iyi çözüm, $\|A\theta - B\|^2$ nin minimum değeri olan, en küçük kareler tahmini (LSE) θ^* dir.

$$\theta^* = (A^T A)^{-1} A^T B \quad (4.7)$$

Burada A matrisinin i .satırını a_i^T ile B matrisinin i .elemanı b_i^T olsun. θ vektörü :

$$\theta_{i+1} = \theta_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T \theta_i) \quad (4.8)$$

$$S_{i+1} = S_i + \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}, \quad i = 0, 1, \dots, P-1 \quad (4.9)$$

Burada en küçük kareler tahmincisi θ^*, θ_p 'ye eşittir.

Hibrid öğrenme algoritmasının geri yön geçişinde, S2 çıkış parametreleri sabit tutulur ve ağın çıkışındaki hata sinyali geriye doğru yayılarak, eğim düşümü (gradient descent) yöntemiyle giriş parametreleri güncellenir. Geri yön geçişinde giriş parametreleri için güncelleme formülü aşağıdaki gibidir [14],

$$\Delta w_{ij} = -\eta \frac{\partial E(\vec{w})}{\partial w_{ij}} \quad (4.10)$$

4.1.3. Hibrid Öğrenme (On-Line)

Eğer veri sunulduğu her defasında parametreler güncelleniyorsa bu tür hibrid algoritmaya online öğrenme adı verilir. Yukardaki formüle unutma faktörü eklenirse, λ , 0 ile 1 aralığında değişmektedir [13];

$$\begin{aligned} \theta_{i+1} &= \theta_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T \theta_i) \\ S_{i+1} &= \frac{1}{\lambda} \left[S_i + \frac{S_i a_{i+1} a_{i+1}^T S_i}{\lambda + a_{i+1}^T S_i a_{i+1}} \right], \quad i = 0, 1, \dots, P-1 \end{aligned} \quad (4.11)$$

4.2. Anfis Mimarisi

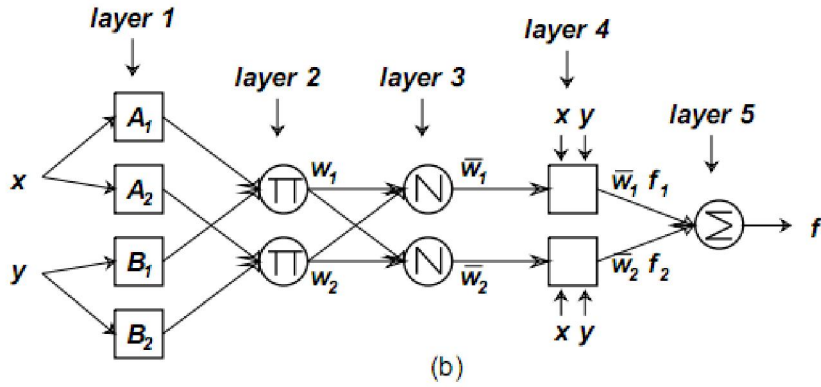
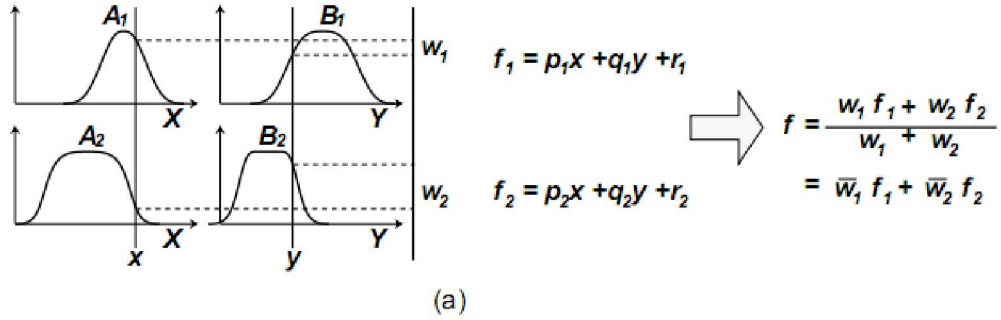
Anfis, Sugeno tipindeki bir modele dayalı olarak işlem yapan adaptif ağ yapılı bir bulanık çıkarım yöntemidir. Yöntem belirli giriş-çıkış veri kümesinden yararlanarak uygun bulanık kural kümesini oluşturması ve adaptif özellikli ağ yapısı ile sistem parametrelerinin ayarlanmasına dayanır. Anfis mimarisini anlayabilmek için iki adet bulanık kuraldan oluşan birinci derece Sugeno model üzerinde x , y giriş ve z çıkış değişkenleri ile iki kural tanımlayalım [15]:

$$\text{Kural 1: Eğer } x = A_1 \text{ ve } y = B_1, \text{ ise } f_1 = p_1 x + q_1 y + r_1,$$

$$\text{Kural 2: Eğer } x = A_2 \text{ ve } y = B_2, \text{ ise } f_2 = p_2 x + q_2 y + r_2,$$

Buradaki x ve y , sistem girişleri; A_i ve B_i , Bulanık Kümeler; f_i , bulanık kural çıkış değeri ve p_i, q_i ve r_i ise eğitim işlemi ile belirlenen sistem parametreleridir. Şekil 4.1 birinci derece Sugeno tip bulanık model ve bu modeli tanımlayan ANFIS yapısı verilmiştir. Beş tabakalı bir ağdan meydana gelen temel

yapıda görülen daire biçimli gösterimler sabit düğümleri ve kare biçimli gösterimler ise adaptif düğümleri ifade etmektedir.



Şekil 4.2 Anfis Mimarisi

(a) Tip 3 Bulanık İlişki

(b) Tip 3 ANFIS Mimarisi

4.2.1. Katman 1

Bulanıklaştırma katmanı olarak adlandırılır. Giriş değerlerini bulanık kümelere ayırmada Jang'ın ANFIS modeli, üyelik fonksiyonu şekli olarak geliştirilmiş Bell aktivasyon fonksiyonunu kullanmaktadır. Katmandaki her i düğümü için;

$$O_i^1 = \mu_{A_i}(x) \quad (4.12)$$

Fonksiyonda x i.nodun giriş değeri, A_i dilsel değişken (Sıcak, küçük, v.b.) ifade etmektedir. Diğer taraftan, O_i^1 A_i değişkenin üyelik fonksiyonu ve x giriş değerinin aitlik derecesini göstermektedir. $\mu_{A_i}(x)$ üyelik fonksiyonu için

genelde maksimum değeri 1, minimum değeri 0 olan çan eğrisi tercih edilir. Fonksiyon eşitliği:

$$\mu_{Ai}(x) = \frac{1}{1 + \left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}} \quad (4.13)$$

ya da Gaussian fonksiyonu;

$$\mu_{Ai}(x) = \exp\left[-\left(\frac{x-c_i}{a_i}\right)^2\right] \quad (4.14)$$

$\{a_i, b_i, c_i\}$ parametre kümesini ifade eder.

4.2.2. Katman 2

Kural katmanıdır. Bu katmandaki her bir düğüm, Sugeno bulanık mantık çıkarım sistemine göre oluşturulan kuralları ve sayısını ifade etmektedir. Burada, her bir düğümün çıkışı sinyalleri birbirleri ile çarpıp ve w_i ağırlıklarını hesaplar. Örneğin;

$$w_i = \mu_{Ai}(x) \times \mu_{Bi}(y), i=1,2. \quad (4.15)$$

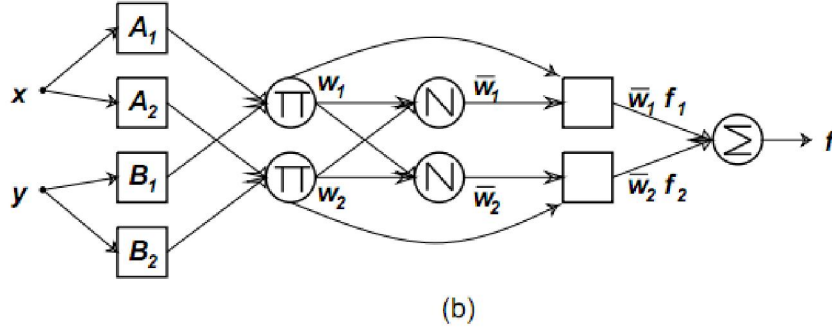
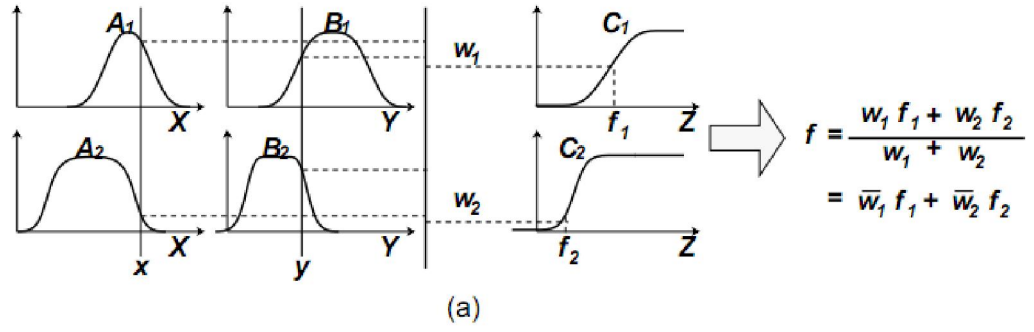
Her bir düğüm noktası kuralın ateşleme gücünü gösterir.

4.2.3. Katman 3

Normalizasyon katmanıdır. Bu katmandaki her bir düğüm, kural katmanından gelen tüm düğümleri giriş değeri olarak kabul etmekte ve her bir kuralın normalleştirilmiş ateşleme seviyesini hesaplamaktadır. Her bir kural düğümünün çıkışı tüm kuralların ateşleme gücünü oluşturular.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i=1,2. \quad (4.16)$$

Bu katmanın çıkışına ateşleme gücünün normalizasyonu adı verilir.



Şekil 4.3 Anfis Bulanık İlişki Mimarisi

- (a) Tip 1 Bulanık İlişki
- (b) Tip 1 ANFIS Mimarisi

4.2.4. Katman 4

Durulaştırma katmanıdır. Durulaştırma katmanındaki her bir düğümde verilen bir kuralın ağırlıklandırılmış sonuç değerleri hesaplanmaktadır. 4.katmandaki her düğüm noktası için;

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad i=1,2. \quad (4.17)$$

\bar{w}_i 3.katmanın çıkışı, $\{p_i, q_i, r_i\}$ parametre kümesidir. Bu katmandaki parametreler sonuç parametreleri (consequent parameters) adı verilir.

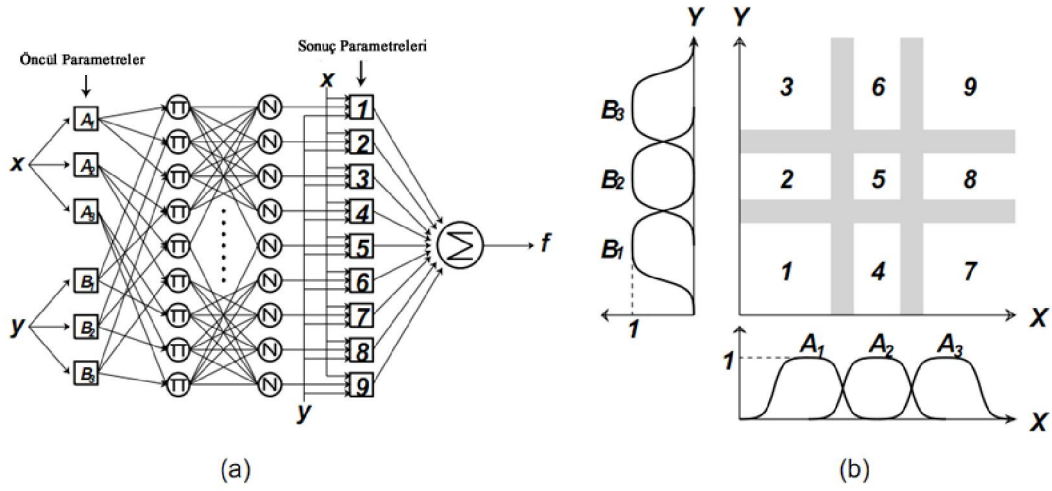
4.2.5. Katman 5

Toplam katmanıdır. Bu katmanda sadece bir düğüm vardır ve \sum ile etiketlenmiştir. Burada, 5. katmandaki her bir düğümün çıkış değeri toplanarak 5. katmandaki tek düğüm noktası gelen sinyallerinden çıkış değeri elde eder.

$$O_i^5 = \text{toplam çıkış} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, i=1,2. \quad (4.18)$$

Böylelikle Tip 3 formundaki bulanık çıkarım sisteminin uyarlamalı ağı yapılandırılır. Şekil 4.3'de görüleceği üzere Tip 1 ANFIS mimarisinde 3. ve 4. Katmanlar hesaba katılmadan direk sonuç hesaplanmaktadır. Tip 2 formundaki Anfis mimarisinde bundan farklı merkez durulaştırma operatörü olarak alanın merkez operatörü kullanılmaktadır.

Şekil 4.4'de 2 giriş, 1 çıkış ve 9 kural içeren Tip 3 formunda Anfis yapısı görülmektedir.



Şekil 4.4 Anfis Bulanık İlişki Kuralları

4.3. Hibrid Öğrenme Algoritması

Şekil 4.2'deki Anfis Mimarisi parametreler ile tanımlanmaktadır ve çıkış fonksiyonu sonuç parametrelerin lineer kombinasyonudur. f fonksiyonu:

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \quad (4.19)$$

$$f = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad (4.20)$$

$$f = (\bar{w}_1 x) p_1 + (\bar{w}_1 x) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_1 \quad (4.21)$$

Lineer sonuç parametreleri $\{p_1, q_1, r_1, p_2, q_2, r_2\}$,

S = tüm parametreler,

S_1 = öncül parametreler,

S_2 = sonuç parametreler olarak tanımlanır.

Hibrid öğrenme algoritması ile 4.katmana kadar iletilen sinyaller işlenir ve en küçük kareler yönetimi kullanılarak sonuç parametreleri hesaplanır. Geri dönüşte hata derecesi geriye doğru yayılır ve “gradient descent” yöntemi ile parametreler tekrar güncellenir. Parametrelerin güncelleme tablosu şu şekildedir;

	İleri geçiş	Geriye geçiş
Öncül parametreler	Sabit	Gradient descent
Sonuç parametreleri	En küçük kareler	Sabit
Sinyaller	Düğüm çıkışları	Hata oranları

Tablo 4.1 Parametre Güncelleme Kuralları

Hibrid öğrenme algoritmasında öncül ve sonuç parametrelerin güncellemesi gradient descent algoritmasına göre oldukça performanslıdır. Bunun yanı sıra performansı arttıran diğer algoritmalar; Conjugate descent, second order back propagation, quick propagation olarak sayabiliriz.

4.4. Anfis Matlab Araçları

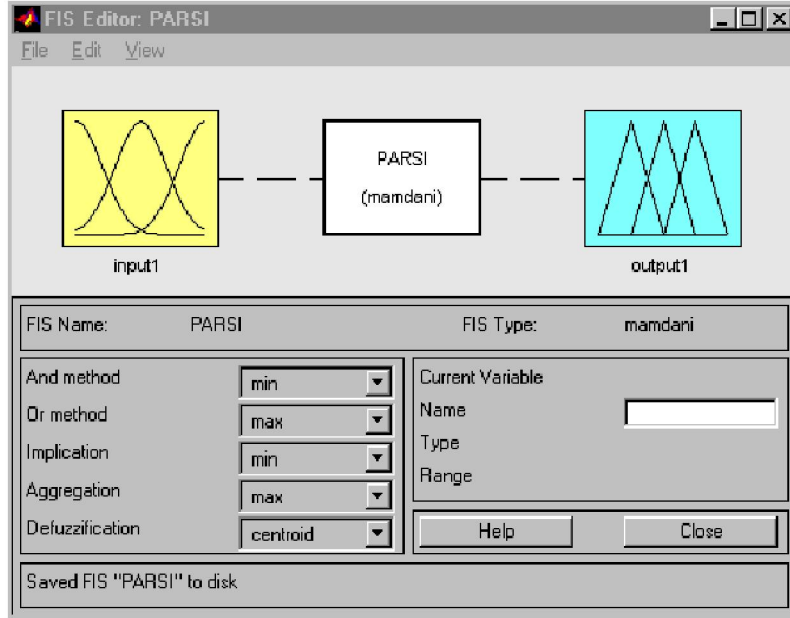
Matlab programında Anfis problemlerini çözüm yaklaşımı ve araçların kullanılmasını görmek üzere örnek üzerinde inceleyelim. x ve y giriş-çıkış değerleri aşağıdaki tabloda verilmiştir.

x	y
-2	2
-1	0.5
0	0
-1	0.5
-2	2

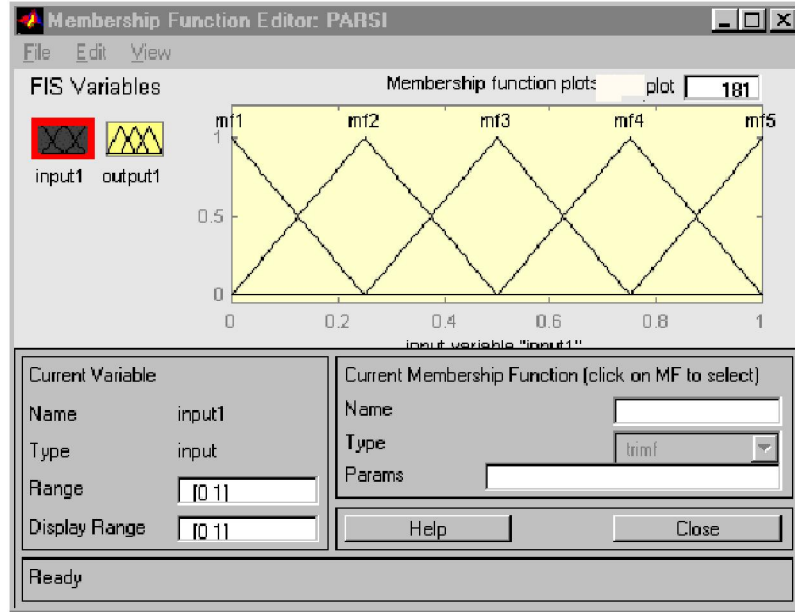
Tablo 4.2 Örnek Matris

Tablodaki verinin fonksiyonu $y = x^2$ olsun. Veri Mamdani ilişki ile giriş değerleri için üçgensel üyelik fonksiyonu, çıkış değeri için ise basit üyelik fonksiyonu olarak tanımlansın.

Kurallar ; Eğer $x = \mu_{f_i}$ ise $y = \mu_{f_i}$ formu ile tanımlayalım.

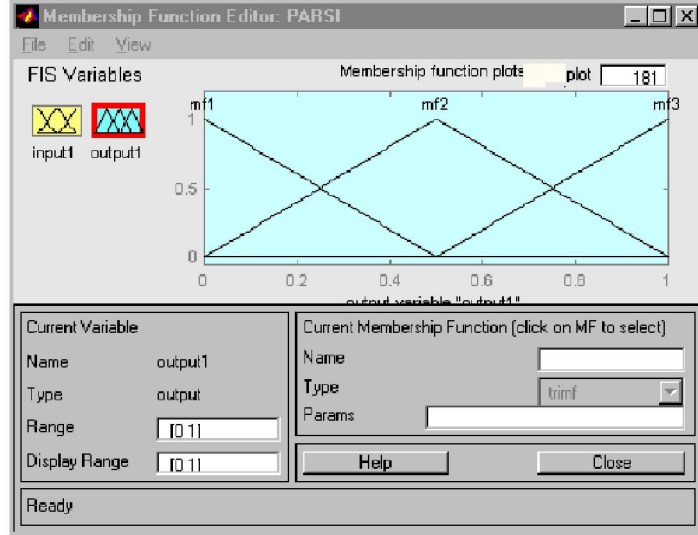


Şekil 4.5 Mamdani ilişkili FIS Editörünün Standart Görüntüsü



Şekil 4.6 Giriş Değerlerinin Üyelik Fonksiyonları

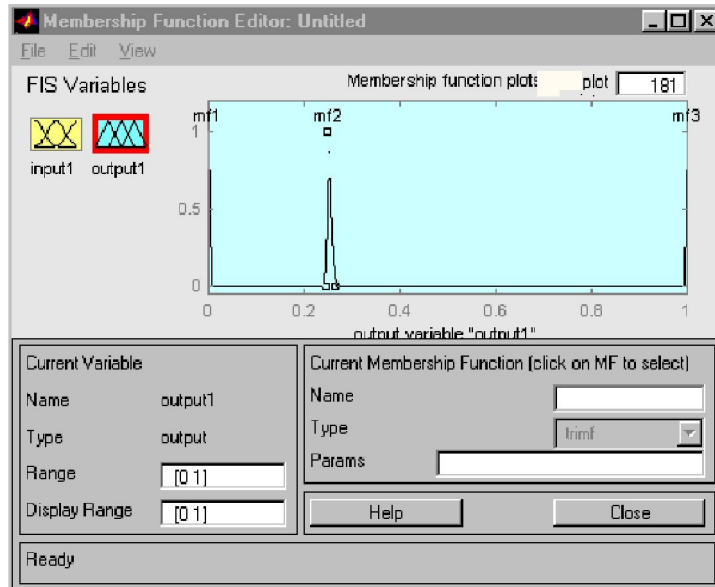
“input1” üzerine basıldığında veriler için oluşturulmuş üyelik fonksiyonları görülebilir. Aralık (Range) alanı [0 1] olarak ayarlanmıştır. Bunun anlamı; x değişkeni maksimum 1 minimum 0 değerini alabilir.



Şekil 4.7 Çıkış Değerlerinin Üyelik Fonksiyonları

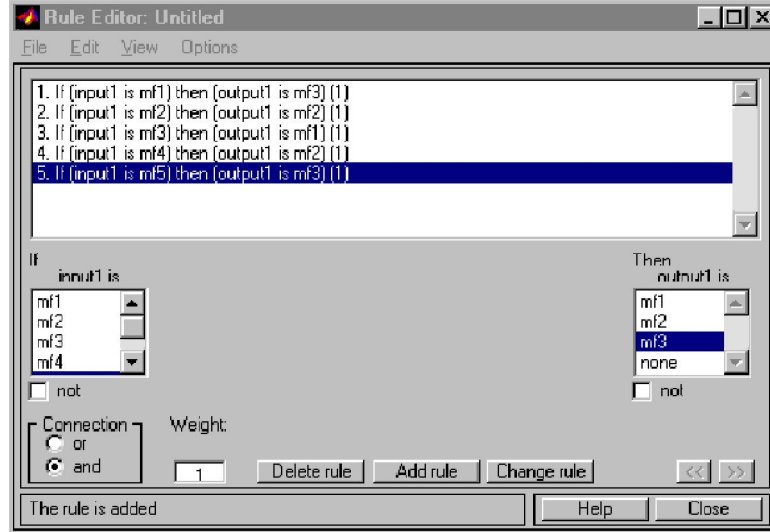
“output1” üzerine basıldığında çıkış verileri için oluşturulmuş üyelik fonksiyonları görülebilir. Range alanı [0 1] olarak ayarlanmıştır. Bunu anlamı; y değişkeni maksimum 1 minimum 0 değerini alabilir.

Çıkış üyelik fonksiyonlarını darlaştırılarak basit haline getirirsek:



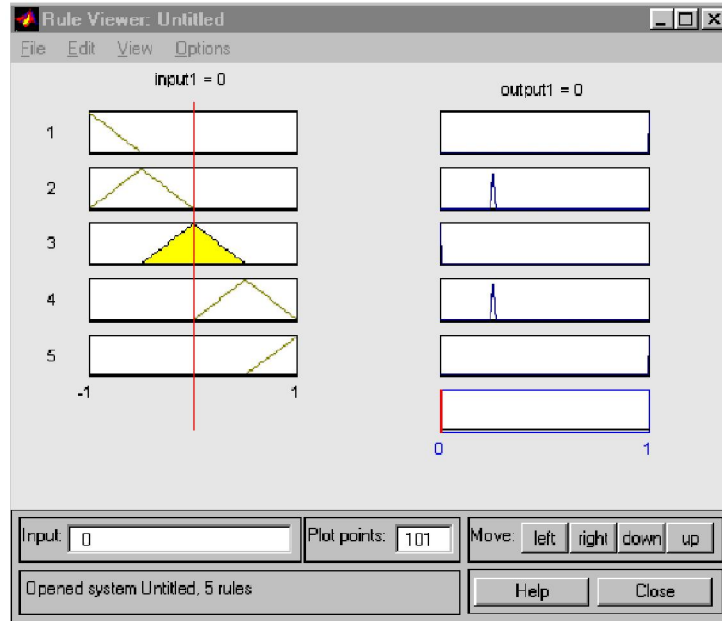
Şekil 4.8 Çıkış Değerlerinin Aralıklarının Darlaştırılması

Anfis kuralları kural editörü kullanılarak tanımlanır. Örnekteki kurallar aşağıdaki gibi tanımlansın:



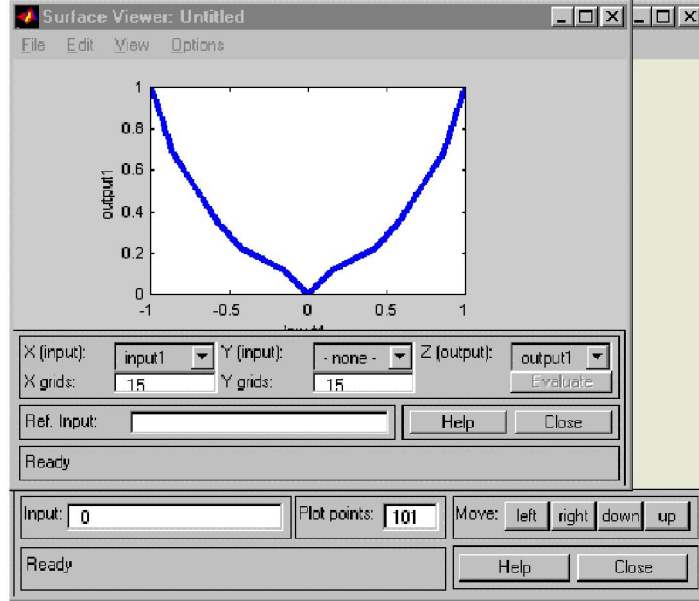
Şekil 4.9 Giriş-Çıkış Kural Tablosu

Kuralların işleyişinin incelenebilmesi için kural izleyici araç kullanılabilir. Kuralların izlenmesi aşağıdaki gibidir:



Şekil 4.10 Giriş-Çıkış Kural İzleme Ekranı

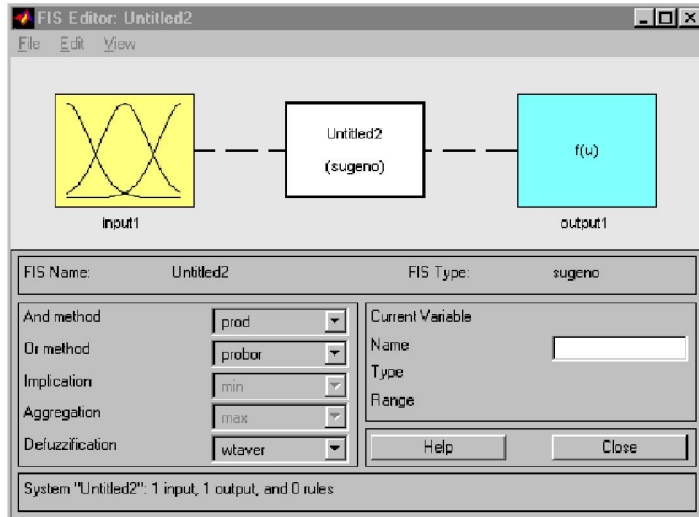
Sonuç verinin kurallar dahilinde oluşturduğu yakınsama Yüzey İzleme Aracından izlenebilir.



Şekil 4.11 Giriş-Çıkış Sonuç Yüzey İzleme Ekranı

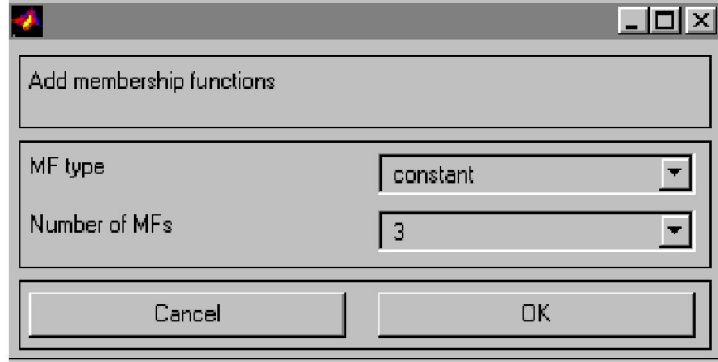
Giriş verilerine göre $y = x^2$ grafiğine yakınsama grafiği çıkış ekranından görülmektedir. Giriş veri sayısı arttıkça $y = x^2$ fonksiyonuna yakınsama artacaktır. Benzer olarak, yukardaki örneği Sugeno tipi bulanık çıkarım sistemi ile ele alırsak;

Kurallar ; Eğer $x = X_i$ ise $y = y_i$ formu ile tanımlayalım.



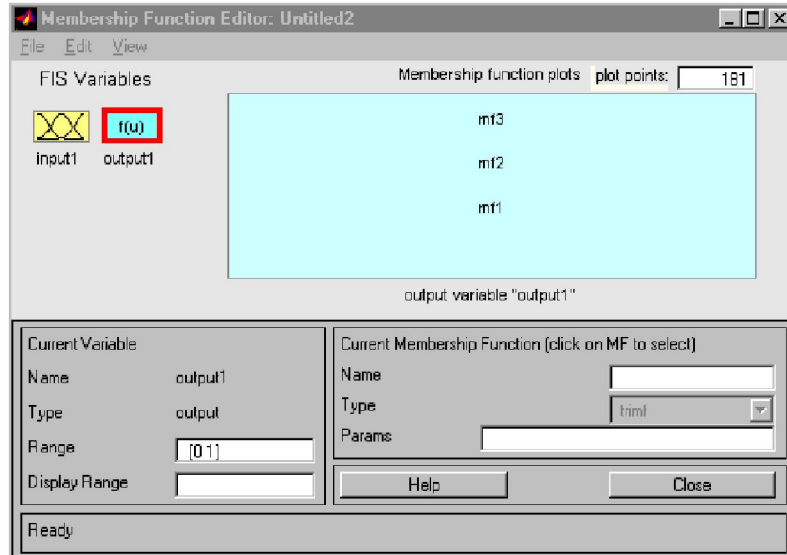
Şekil 4.12 Sugeno ilişkili FIS Editörünün Standart Görüntüsü

“input1” üzerine basılarak veriler için üyelik fonksiyonları oluşturulabilir. Örneğin “output1” çıkış bölümü seçilerek üyelik fonksiyonu ekleme menüsü seçilir. Aşağıdaki gibi çıkan ekran üzerinden kaç tane çıkış üyelik fonksiyonu olduğu ve tipi seçilir. Örnek için 3 adet sabit çıkış seçelim.



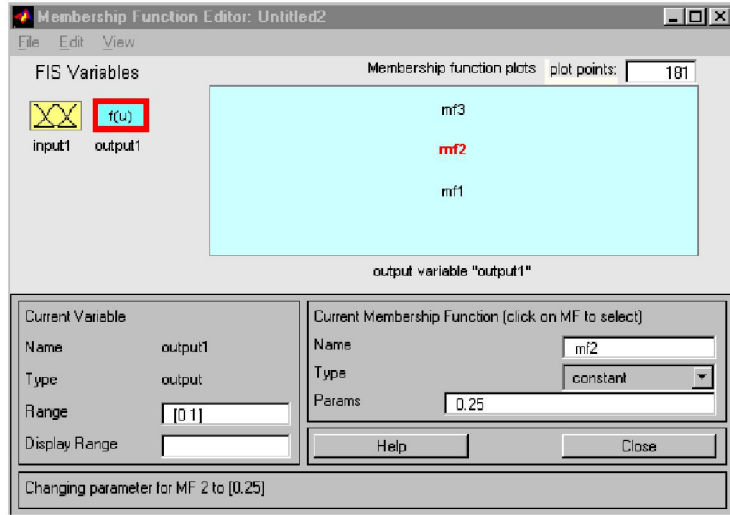
Şekil 4.13 Sugeno ilişkili FIS Editörünün Üyelik Fonksiyon Giriş Ekranı

Üyelik fonksiyonları girildikten sonra aşağıdaki çıktıya benzer ekran görüntüsü alınabilir.



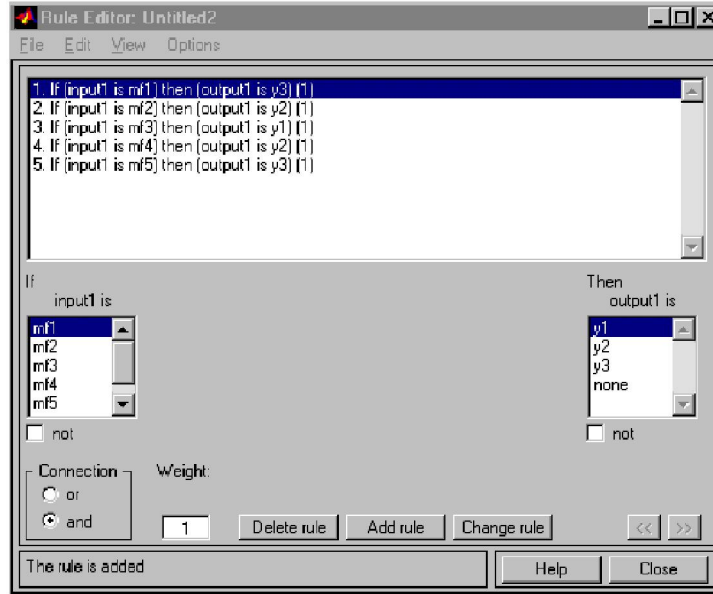
Şekil 4.14 Sugeno ilişkili FIS Editörünün Çıkış Birimi Üyelik Fonksiyonları

Üyelik fonksiyonların üstlerine tıklanıldığında kendisine ait özellikler buradan değiştirilebilir. Örneğin aralık olarak [0 1] verilmiş ve yakınsamasını istenen çıkış değeri 0.25 olarak tanımlanmıştır. Aşağıdaki ekranda bu ayarları görebiliriz:



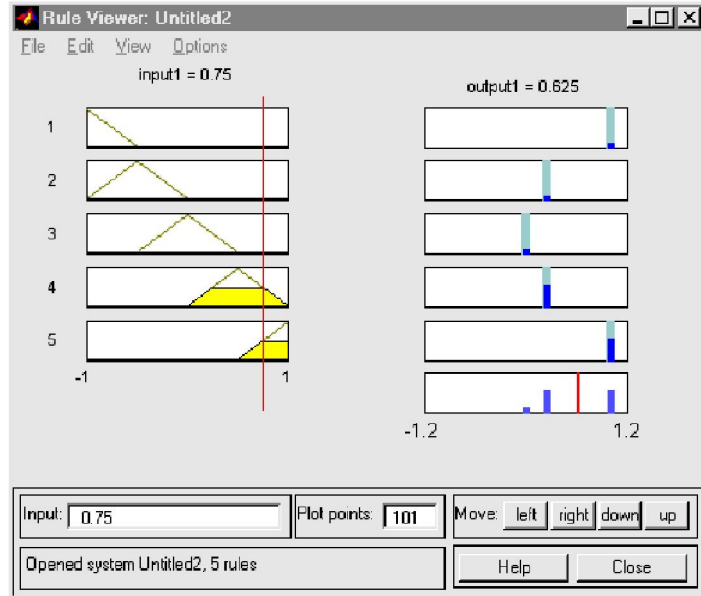
Şekil 4.15 Sugeno ilişkili FIS Editörünün Çıkış Birimi Üyelik Fonksiyonları

“Edit Rules” menüsünden açılan aşağıdaki gibi kural editöründe değişken arası ilişkilerin tanımlamaları yapılmaktadır.



Şekil 4.16 Sugeno ilişkili FIS Editörünün Kural Tanımlamaları

Kuralların tanımlanmasından sonra Sugeno tipi bulanık çıkarım sistemi kurulmuş olur. Oluşturulan sistemin ilişkilerinin incelenmesi için kural izleme “Rule Viewer” düğmesine basılır. Açılan aşağıdaki ekranda olduğu gibi x değişkeni (kırmızı çizgi) değiştirilerek çıkış değerinin hesaplamaları incelenebilir.



Şekil 4.17 Sugeno ilişkili FIS Editörünün Kural İzleme Ekranı

5. OTOMATİK TREN YÖNETİM SİSTEMİ

Otomatik Tren Yönetim Sistemi (Automatic Train Operation - ATO), lokomotifin kontrol birimlerinin hızlandırma ve yavaşlatma fonksiyonlarını kullanarak sistemi etkileyen iç ve dış faktörler altında en iyi kararı yerine getiren sistem olarak tanımlanabilir. Sistem makinist olmadan kendi kendine yönetimi sağlamak üzere tasarlanmıştır. Tamamen otomatik olarak hareket eden tren, bir noktadan bir noktaya hızlı, güvenli ve konfor düzeyini en üstte tutarak seferini tamamlar. Japonya Hitachi Sistem Geliştirme Laboratuvarında üzerinde çalışan araştırmacılar tarafından ilk defa Sendai Municipal Metro Sisteminde uygulanmıştır. 14 km boyunca 16 istasyonda duran tren duruş ve kalkış ivmelerinin en az hissedildiği toplu taşıma aracıdır ve bu yolla trenin istenen konumda durması üç kat daha iyileştirilmiş, kullanılan enerji ise %10 azaltılmıştır [8].

Otomatik Tren Yönetim Sistemi;

- ✓ Tren varış zamanı,
- ✓ Hız ayarlama,
- ✓ İstasyonun hedef noktasında durma kontrollerini gerçekleştirir.

5.1. Otomatik Tren Yönetim Sisteminin Genel İşleyişi

Türkiye Cumhuriyeti Devlet Demiryolları kurumu altında farklı amaçlar için kullanılan birçok tipte lokomotif bulunmaktadır. Elektrikli, Dizel, Dizel-Elektrikli, Hızlı Tren, EMU, DMU, Banliyö, Manevra Lokomotifleri bunlardan bazılarıdır. Bu kadar çeşitli tiplerdeki lokomotiflerin kabin içleri de birbirinden farklıdır. Ancak lokomotif bir noktadan bir noktaya götürülmek isteniyorsa makinist tarafından yapılan işlemler benzerlik gösterir. Lokomotifler üzerinde dinamik fren ve kampana fren olmak üzere iki tip fren bulunmaktadır. Dinamik fren, motorun ters çalışması ile oluşturulan enerji kullanılarak frenleme yapmasıdır. Kampana fren, direk tekerleklerin durdurulması ile frenleme sağlar. Makinist fren kollarını kapattıktan sonra ivmelenme kademesini seçerek lokomotifin hareket etmesini sağlar. Lokomotif hareket halinde iken dinamik

frenin kademeleri ile anlık yavaşlama sağlanmaktadır. Lokomotiflerin otomatik yönetilmesi söz konusu olduğunda yukarıdaki bahsedilen kontroller kullanılarak yönetim sağlanmaktadır.

Otomatik Tren Yönetim Sistemi lokomotif ve yol üzeri alt sistemlerden oluşmaktadır [15, 16, 17, 18].

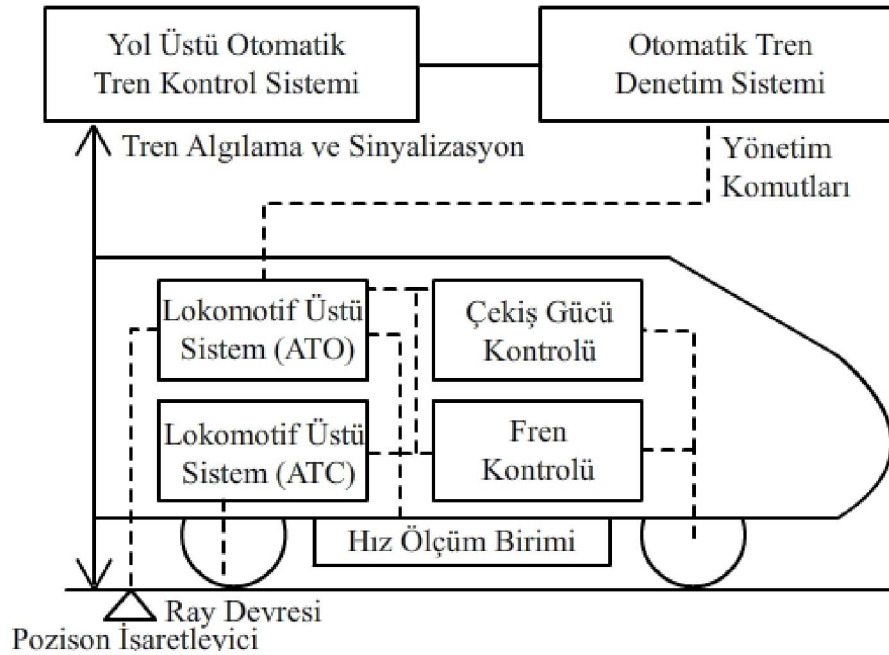
Yol üzeri sistemler;

- ✓ Yol Üstü Otomatik Tren Kontrol Sistemi (ATC Wayside System)
- ✓ Otomatik Tren Denetim Sistemi (ATS System)
- ✓ Pozisyon İşaretleyici (Position Marker)
- ✓ Ray Devresi (Track Circuit)

Lokomotif üzeri sistemler;

- ✓ ATO Lokomotif Üstü Sistem (ATO Onboard System)
- ✓ Otomatik Tren Kontrol Lokomotif Ekipmanı (ATC Onboard System)
- ✓ Çekiş gücü Kontrolü (Traction Controller)
- ✓ Fren Kontrolü (Brake Controller)
- ✓ Hız Ölçüm Birimi (Tacho Generator)

Bu sistemlerin birbirleri ile ilişkileri aşağıdaki diyagramda gösterilmiştir.



Şekil 5.1 Otomatik Tren Yönetim Sistemi

Tren operasyonları aşağıdaki gibi iki kontrol modunda incelenebilir.

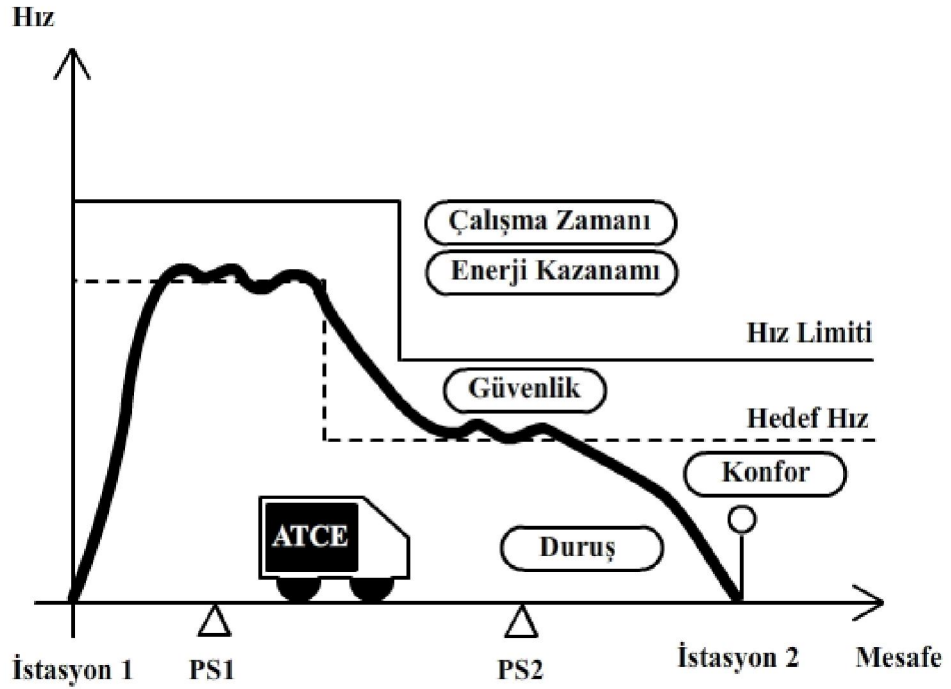
(1) Tren Hız Ayarlama ve Sabit Hız Kontrolü (CSC) :

Bu kontrol modunda trenler harekete başlar ve hesaplanan hedef hız değerine göre devam ederler.

(2) Tren Otomatik Durma Kontrolü (TASC):

Bu kontrol altında en yüksek titizlikle hedeflenen bölgede durma işlevini yerine getirirler.

Otomatik Tren Yönetim Sisteminde makinistin teknik becerisi bir örnek oluşturmakta ve bu yönetim sisteminde güvenlik, konfor gibi parametreler ile kontrol edilmektedir. Aşağıdaki diyagramda kontrol indekslerini görmekteyiz.



Şekil 5.2 Otomatik Tren Yönetim Sistemi Diyagramı

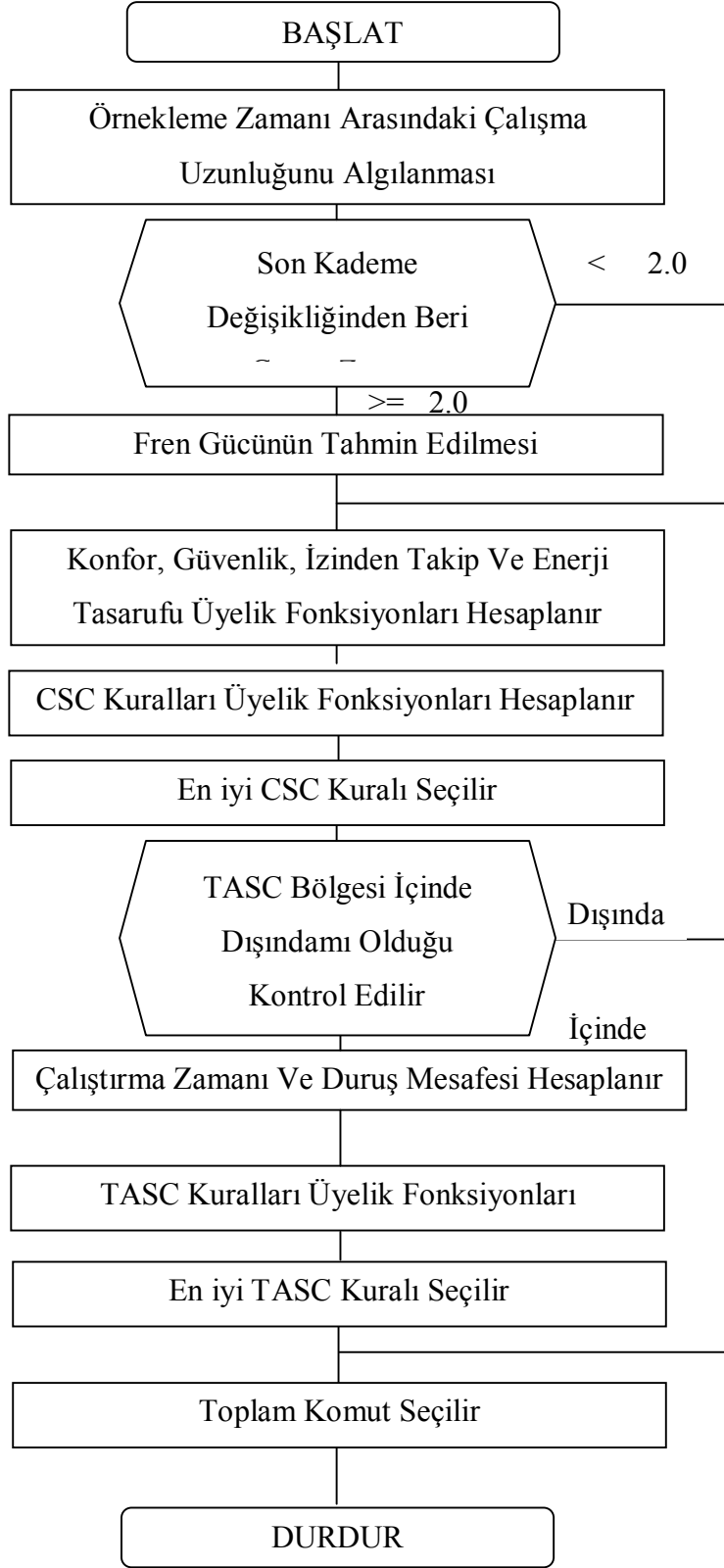
ATCE : Otomatik Tren Kontrol Birimleri

PS1: CSC Pozisyon İşaretleyicisi

PS2: TASC Pozisyon İşaretleyicisi

5.2. Bulanık Çıkarım Sistemi ile Otomatik Tren Yönetim Sistemi

Otomatik Tren Yönetim Sisteminin giriş değişkenlerini hız, kontrol birim kademesi, odometre mesafe bilgisi, anlık hız bilgisi, hız sınırlama bilgisi oluşturmaktadır. Otomatik Tren Yönetim Sisteminin kontrol komutları ise güç için 7 kademe, frenleme için 9 kademe ile tanımlanmıştır. Sabit Hız Kontrolü (CSC) ve Otomatik Duruş Kontrolü (TASC) için bulanık çıkarım sistemi ile kontrol kuralları tanımlanır. Sistemin çalışma algoritması Şekil 5.3’de verilmiştir.



Şekil 5.3 Otomatik Tren Yönetim Sistemi Akış Şeması

5.3. Otomatik Tren Yönetim Sistemi için Bulanık Kurallar

Otomatik Tren Yönetim Sisteminin kontrolü için oluşturulmuş Sabit Hız Kontrolü ve Otomatik Duruş Kontrolü kuralları aşağıdaki gibi tanımlansın;

5.3.1. Sabit Hız Kontrolü (CSC)

- (1) Eğer kontrol komutu değişmemiş ise ve takip edilen hedef hızının durumu “iyi” ise, sonra kontrol komut değişmeyebilir.

$$(K-1) \text{ Eğer } \left(N = \frac{N(t)+B_{max}}{2} \rightarrow S = SD \right) \text{ ise } N = \left(\frac{N(t)+B_{max}}{2} \right)$$

- (2) Eğer kontrol komutu güç için 0. Kademesine getirilmiş ise, konfor sürekliliği “İyi” ve takip edilen hedef hız “Çok iyi” ise , sonra komut 0 kademeye alınabilir.

$$(K-2) \text{ Eğer } (N = 0 \rightarrow S = SS, C = CG \text{ ve } E = ES) \text{ ise } N = 0$$

- (3) Eğer kontrol komutu güç için 7. Kademeye getirildi ise ve konfor sürekliliği “iyi” ise ve hedef hız takibi “çok iyi” ise, sonra komut 7. Kademeye getirelebilir.

$$(K-3) \text{ Eğer } (N = 7 \rightarrow S = SS, C = CG \text{ ve } T = TL) \text{ ise } N = P7$$

- (4) Eğer kontrol komutu +/- n . kademelerde ve arasındaki fark 0 ise ve konfor sürekliliği “iyi” ise ve hedef hız takibi “çok iyi” ise, sonra komut n. Kademeye getirelebilir. (n=1,2 ve 3)

$$(K-4) \text{ Eğer } (DN = 0 \rightarrow S = SS \text{ ve } T = TG) \text{ ise } DN = 0$$

- (5) Eğer kontrol komut farkı frenleme +/- n. Kademe de ise ve konfor sürekliliği “iyi” ise ve hedef hız takibi “çok iyi” ise, sonra komut 4. Kademeye getirelebilir.

$$(K-5) \text{ Eğer } (DN = n \rightarrow S = SS, C = CG \text{ ve } T = TA) \text{ ise } DN = n$$

$$(n = +/-1, +/-2, +/-3)$$

5.3.2. Otomatik Duruş Kontrolü (TASC)

- (1) Eğer kontrol komutu frenleme 7. Kademeye değişmiş ise ve güvenlik durumu “tehlikeli” ise, sonra komut acil duruş kademesine değiştirilebilir.

$$(K-6) \text{ Eğer } (BN = 7 \rightarrow S = SD) \text{ ise } BN = 9$$

- (2) Eğer kontrol komutu güç 0. Kademeye değişmiş ise ve konfor sürekliliği “iyi” ise ve yavaşlama başlangıç noktasına ulaşmadan önce zaman sınırı “Büyük” ise, sonra komut 0. Kademeye değiştirilebilir.

$$(K-7) \text{ Eğer } (PN = 0 \rightarrow C = CG \text{ ve } R = RF) \text{ ise } PN = 0$$

- (3) Eğer kontrol komutu frenleme 1. Kademeye değişmiş ise ve konfor sürekliliği “iyi” ise ve TASC bölgesinin içinde ise, sonra komut 1. Kademeye değiştirilebilir.

$$(K-8) \text{ Eğer } (BN = 1 \rightarrow C = CG \text{ ve } R = RP) \text{ ise } PN = B1$$

- (4) Eğer kontrol komutu frenleme +/-n Kademeye değişmiş ise ve konfor sürekliliği “iyi” ise ve yavaşlama başlangıç noktasına ulaşmadan önce zaman sınırı “Yok” ise ve duruş doğruluğu “çok iyi” ise, sonra komut n. Kademeye değiştirilebilir. (n=1,2 ve 3)

$$(K-9) \text{ Eğer } (DN = n \rightarrow C = CG, G = GA \text{ ve } R = RP) \text{ ise } DN = n \\ (n = +/-1, +/-2, +/-3)$$

5.4. Otomatik Tren Yönetim Sistemi Bulanık Çıkarım Fonksiyonları

Üyelik fonksiyonlarında kullanılmak üzere A, B ve C fonksiyonları aşağıdaki gibi tanımlansın;

$$\begin{aligned} A(x, a, b) &= \frac{b}{a+b-x} && : x < a, b > 0 \\ &= 1.0 && : a \leq x, b > 0 \\ &= 0 && : x < a + b, b < 0 \end{aligned} \quad (5.1)$$

$$= 1.0 + \frac{(a - x)}{2b} \quad : a + 2b \leq x < a, b < 0$$

$$B(x, a, b) = A(2a - x, a, b) \quad (5.2)$$

$$C(x, a_1, b_1, a_2, b_2) = \min(A(x, a_1, b_1), B(x, a_2, b_2)) \quad (5.3)$$

5.5. Bulanık Çıkarım Sistemi Fonksiyonlarının Matlab Kodları

(1) $A(x, a, b)$ fonksiyonu FN_A fonksiyonu ile adlandırılınsın. x değişkeni ve parametre değerleri ile tanımlanan FN_A fonksiyonun kaynak kodu:

```
function out = FN_A(x, params)
    a = params(1);
    b = params(2);
    y = 0;
    if ((x < a) && (b > 0))
        y = b / (a + b - x);
    elseif ((a <= x) && (b > 0))
        y = 1;
    elseif ((x < (a + 2 * b)) && (b < 0))
        y = 0;
    elseif (((a + 2 * b) <= x < a) && (b > 0))
        y = 1 + ((a - x) / 2 * b);
    elseif ((a <= x) && (b < 0))
        y = 1;
    end
    out = y;
end
```

(2) $B(x, a, b)$ fonksiyonu FN_B fonksiyonu ile adlandırılınsın. x değişkeni ve parametre değerleri ile tanımlanan MF_B fonksiyonun kaynak kodu:

```
function out = FN_B(x, params)
    a = params(1);
    b = params(2);
```

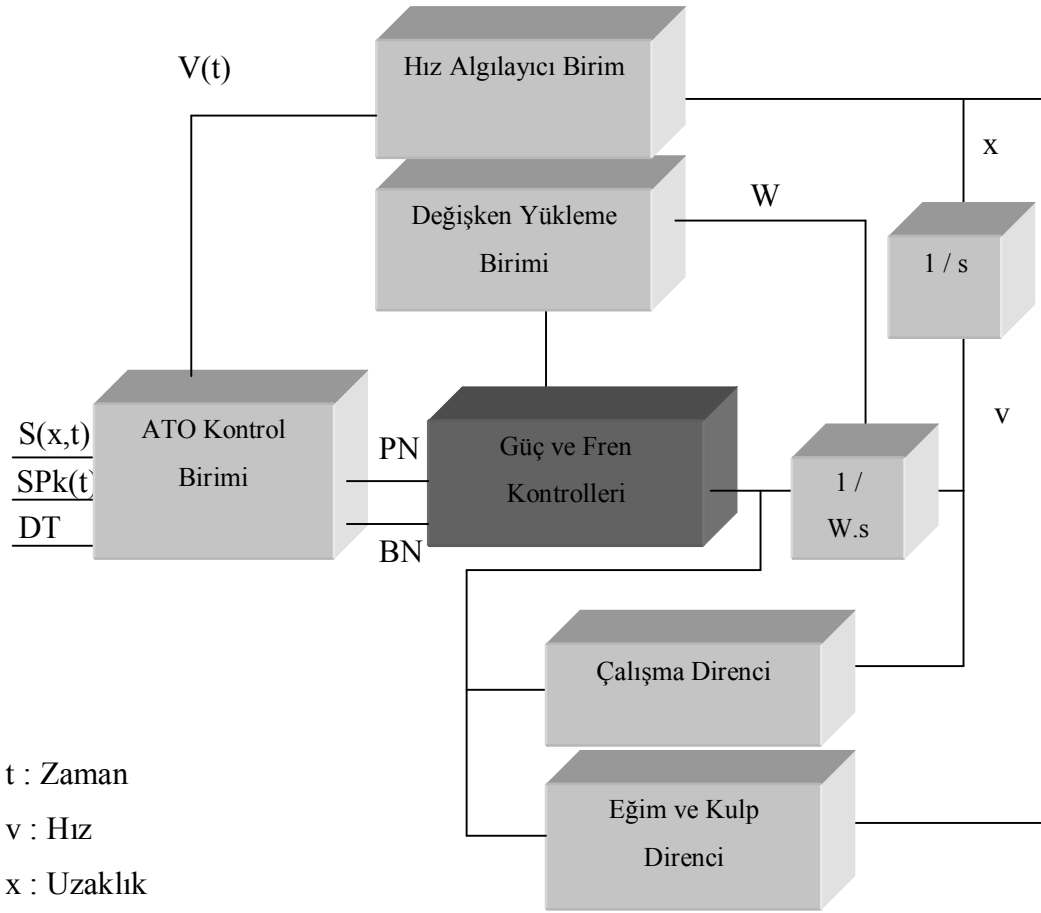
```
out = FN_A((2*a)-x,[a b]);  
end
```

(3) $C(x, a_1, b_1, a_2, b_2)$ fonksiyonu FN_C fonksiyonu ile adlandırılımsın. x değişkeni ve parametre değerleri ile tanımlanan MF_C fonksiyonun kaynak kodu:

```
function out = FN_C(x, params)  
a1 = params(1);  
b1 = params(2);  
a2 = params(3);  
b2 = params(4);  
out = min(FN_A(x,[a1 b1]),FN_A(x,[a2 b2]));  
end
```

5.6. Lokomotif Güç ve Fren Kontrol Bağlantı Şeması

Lokomotif üzerinde Otomatik Tren Yönetim Sistemi Kontrol Birimi, Güç ve Fren Kontrol Birimleri, Hız Algılama Birimi bulunmaktadır. Sistemi etkileyen çalışma direnci, eğim ve kulp direnci sistem birimleri arasındaki ilişkileri Şekil 5.4'de gösterilmiştir;



t : Zaman

v : Hız

x : Uzaklık

S : ATC Sinyali

PN : Güç kademesi

BN : Fren Kademesi

SPk : Nokta Sinyali

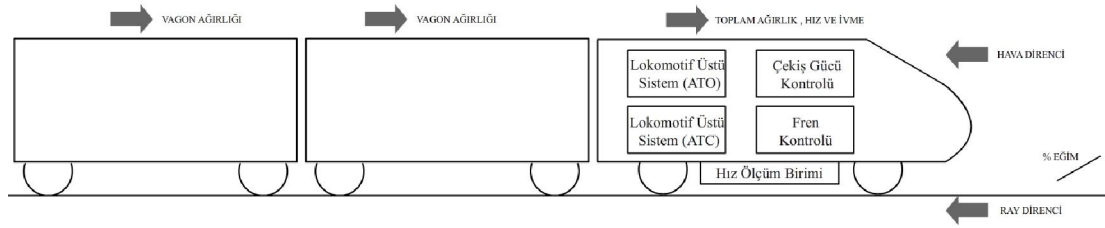
DT : Varış Zamanı

Şekil 5.4 Birim Bağlantı Diyagramı

5.7. Lokomotif Simülatörü

Otomatik Tren Yönetim Sistemini bulanık mantık ve Anfis teknikleri ile testleri yapılırken lokomotifin gerçek koşullardaki davranışını simüle etmek için lokomotif simülatörü yapılmıştır. Bu simülatör temel olarak yolun eğimi, ray kulp açısı, hava direnci, anlık hız ve ivme, trenin vagonları ile birlikte toplam ağırlığı

ve lokomotifin cer motorlarının çalışma direnci parametrelerini kullanarak lokomotifin gerçek davranışını benzetmeye çalışmaktadır. Örneğin lokomotif üzerinde hızlanma için gerekli olan ivmelenme kademesi herhangi bir seviyeye getirildiğinde toplam ağırlık, yolun eğimi, lokomotifin tipi v.b değişkenler hızın azalmasına ve gerekli gücün artırılmasına ihtiyaç duyarken, lokomotifin önceden hareket halinde olması, taşıdığı yükün hafif olması gibi etkenler hızlı ivmelenmesini ve az güç ihtiyacını gerektirmektedir. Simülatör bu etmenler ile bir sonraki t anında olacak hız ve konum bilgilerini türetilmiştir. Aşağıdaki şekilde dış etmenlerin lokomotive etkileri gösterilmektedir.



Şekil 5.5 Lokomotif Simülatörü Dış Etmenler Diyagramı

5.7.1. Lokomotif simülatörü matlab kodu

Simülasyon fonksiyonu t anındaki hız, t anındaki konum, seçilen kontrol kademesi, son kademe değişikliğinden beri geçen zaman değişkenleri ile t+1 zamanındaki hız ve konum bilgilerini oluşturulması aşağıdaki kodda verilmiştir.

```
function [CurrentSpeed, X_t] = Speed_Sim(CurrentSpeed, X_t, Nc, tc)
```

```
%Vagon Sayısı
```

```
VagonCount=5;
```

```
%Vagon Ağırlığı
```

```
VagonWeight=5000;
```

```
%Lokomotif Ağırlığı
```

```
LocomotiveWeight=100000;
```

```
%Toplam Ağırlık
```

```
TotalWeight=(VagonWeight.*VagonCount)+LocomotiveWeight;
```

```
%Yerçekim Ağırlığı
```

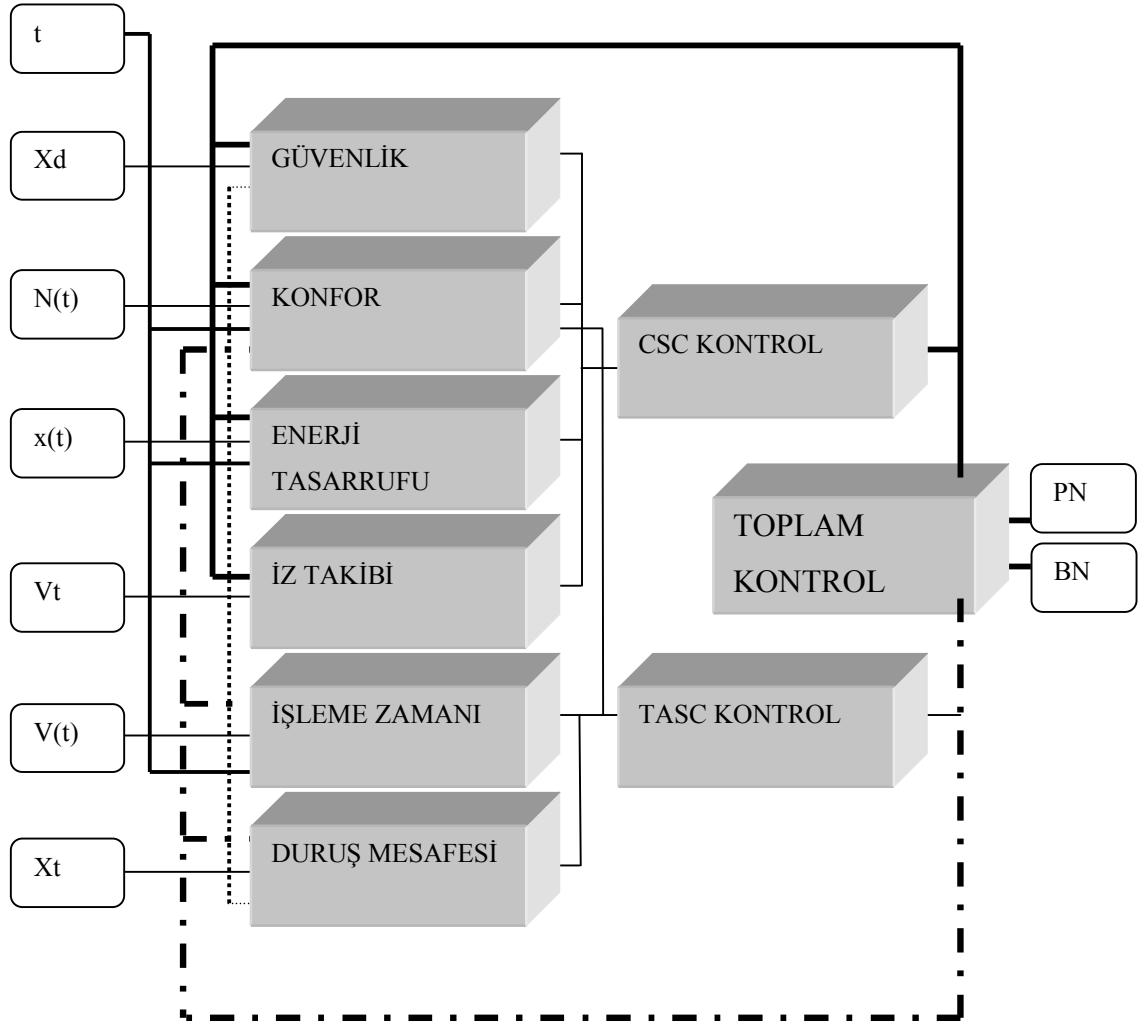
```

YerCekim=9.81;
%Yolun Eğimi
Egim=1;
%Simülatör Fonksiyonları
FKuvvet_RunningResistance=(1.97+(0.016.*CurrentSpeed)+(0.00084.*
CurrentSpeed.*CurrentSpeed)).*sin(degtorad(Egim));
FKuvvet_Egim=-
1*(LocomotiveWeight.*YerCekim.*sin(degtorad(Egim)));
FKuvvet_Ivme=Nc.*TotalWeight;
FKuvvet =
FKuvvet_RunningResistance+FKuvvet_Egim+FKuvvet_Ivme;
%İvme Değişikliği
IvmeDegisimi=FKuvvet./TotalWeight;
%Hız Değişikliği
HizDegisimi=tc.*IvmeDegisimi;
if (HizDegisimi<-6.68)
    HizDegisimi=-6.68;
end
%Tahmin Edilen Hız
PredictedSpeed=CurrentSpeed+HizDegisimi;
if(PredictedSpeed<0)
    PredictedSpeed=0;
    PredictedLocation=X_t;
else
    PredictedLocation=X_t + (CurrentSpeed.*tc) +
(0.5.*IvmeDegisimi.*tc.*tc);
end
CurrentSpeed=PredictedSpeed;
if(CurrentSpeed>0)
    X_t=PredictedLocation;
end
end

```

5.8. Performans İndeksleri - ATO Sistem İlişkisi

Otomatik Tren Yönetim Sisteminde; zaman (t), hız düşüm noktasına uzaklık (X_d), t anındaki seçili kademe ($N(t)$), t zamanında alınan yol ($x(t)$), t zamanında hız ($v(t)$), hedef hız (V_t), duruş mesafesi (X_t) girdileri ile güvenlik, konfor, enerji tasarrufu, iz takibi, işleme zamanı, duruş mesafesi performans indeksleri hesaplanır. Hesaplamalar sonucunda Sabit Hız Kontrolü ve Otomatik Duruş Kontrolü kademe dereceleri bulunur ve bu çıkışların sonucunda toplam kontrol hesaplanır.



Şekil 5.6 Performans İndeksleri ve İlişkileri

5.9. Performans İndeksleri Üyelik Fonksiyonları

Otomatik Tren Yönetim Sisteminde kullanılan İz Takip İndeksi, Konfor İndeksi, Güvenlik İndeksi, İşleme Zamanı İndeksi, Duruş Mesafesi İndeksi, Enerji Tasarrufu İndeksleri ile trenin hızlanması ya da yavaşlaması için gerekli kademe seçimine karar verilir. Bu performans indeksleri t anındaki hız ve konum, mevcut hız ve seçilen kademe, hız düşüm noktası gibi değişkenler ile sürekli sistem içerisinde hesaplanmakta ve simülatör sonuçları da sisteme girdi sağlanılarak döngü oluşturulmaktadır.

5.9.1. İz takip indeksi (Traceability)

Hesaplanan hız ile hedef hızın arasındaki fark iz takip indeksi ile hesaplanır. Lokomotif hızının limit etrafında kalması için hesaplanır.

(a) Oldukça izinden takip (TG) : μ_{TG}

$$\mu_{TG}(V_p(N)) = C(V_p(N), V_t - V_e, V_e, V_t(t) + V_e, V_e)$$

(b) Kesin izinden takip (SS) : μ_{TA}

$$\mu_{TA}(V_p(N)) = C(V_p(N), V_t, V_e, V_t, V_e)$$

(c) Düşük Hız (SS) : μ_{TL}

$$\mu_{TL}(V_p(N)) = B(V_p(N), V_t/2, V_t/4)$$

İz takip indeksi matlab kodu:

$$(a) \mu_{TG}(V_p(N)) = C(V_p(N), V_t - V_e, V_e, V_t(t) + V_e, V_e):$$

Denkleminde tanımlandığı gibi μ_{TG} üyelik fonksiyonu, $V_p(N)$ değişkeni ve ilgili parametre değerlerin bulunduğu C fonksiyonu ile tanımlansın. μ_{TG} üyelik fonksiyonun kaynak kodu:

```
function out = MF_TG(x,params)
    a = params(1);
    b = params(2);
    c = params(3);
```

```

y = zeros(length(x),1);
for i=1:length(x)
    y(i) = FN_C(x(i),[a-b b c+b b]);
end
out = y;
end

```

$$(b) \mu_{TA}(V_p(N)) = C(V_p(N), V_t, V_e, V_t, V_e):$$

Denkleminde tanımlandığı gibi μ_{TA} üyelik fonksiyonu, $V_p(N)$ değişkeni ve ilgili parametre değerlerin bulunduğu C fonksiyonu ile tanımlansın. μ_{TA} üyelik fonksiyonunun kaynak kodu:

```

function out = MF_TA(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_C(x(i),[a b a b]);
    end
    out = y;
end

```

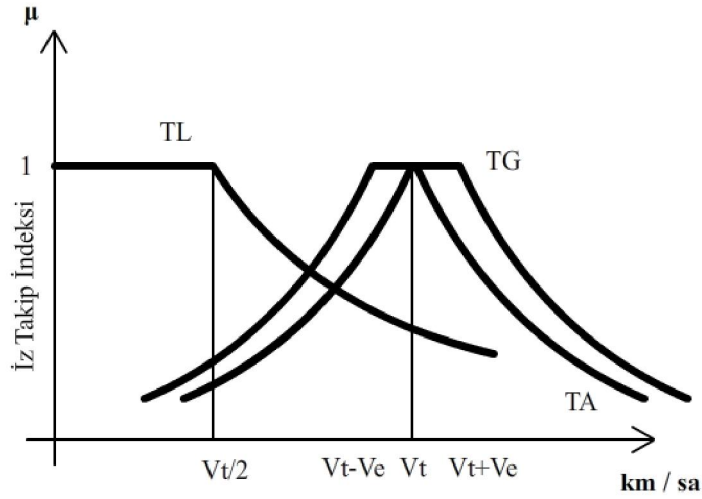
$$(c) \mu_{TL}(V_p(N)) = B(V_p(N), V_t/2, V_t/4):$$

Denkleminde tanımlandığı gibi μ_{TL} üyelik fonksiyonu, $V_p(N)$ değişkeni ve ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{TL} üyelik fonksiyonunun kaynak kodu:

```

function out = MF_TL(x,params)
    a = params(1);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_B(x(i),[a/2 a/4]);
    end
    out = y;
end

```



Şekil 5.7 İz Takip İndeksi Üyelik Fonksiyonu

5.9.2. Konfor İndeksi (Comfort)

Kademede son değişiklik derecesi ve o zamandan beri geçen süre ile hesaplanır. Konfor indeksi yolculuğun konforlu geçip geçmediğini takip eder ve özellikle anlık ivmelenme ve yavaşlamalarda müdahale ederek konforun en üst seviyede tutulmasını sağlar.

(a) Oldukça Konforlu (CG) : μ_{CG}

$$\mu_{CG}(t_c, N_c) = A(t_c, N_c, -N_c + C_b)$$

(b) Vasat Konforlu (CB) : μ_{CB}

$$\mu_{CB}(t_c, N_c) = B(t_c, C_b, -N_c + C_b)$$

Konfor indeksi matlab kodu:

$$(a) \mu_{CG}(t_c, N_c) = A(t_c, N_c, -N_c + C_b):$$

Denkleminde tanımlandığı gibi μ_{CG} üyelik fonksiyonu, t_c, N_c değişkeni ve ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{CG} üyelik fonksiyonunun kaynak kodu:

```
function out = MF_CG(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
```

```

for i=1:length(x)
    y(i) = FN_A(x(i),[a (-1.*a)+b]);
end
out = y;
end

```

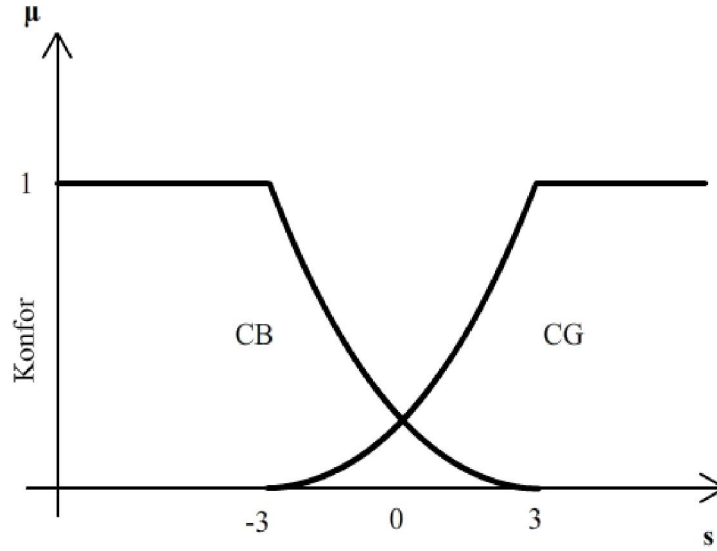
$$(b) \mu_{CB}(t_c, N_c) = B(t_c, C_b, -N_c + C_b):$$

Denkleminde tanımlandığı gibi μ_{CB} üyelik fonksiyonu, t_c, N_c değişkeni ve ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{CB} üyelik fonksiyonunun kaynak kodu:

```

function out = MF_CB(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_B(x(i),[b (-1.*a)+b]);
    end
    out = y;
end

```



Şekil 5.8 Konfor İndeksi Üyelik Fonksiyonu

5.9.3. Güvenlik indeksi (Safety)

Düşük hız limiti (Tehlikeli) olan bölgeye yaklaşıldığında tren için zamana bağlı güvenlik indeksi hesaplanır. Güvenlik indeksi duruş noktasını sürekli takip eder ve durmak için güvenli ve tehlikeli mesafeler için zamanlama hesaplar.

(a) Tehlikeli (SD) : μ_{SD}

$$\mu_{SD}(t_s) = B(t_s, 0.0, -T_s)$$

(b) Güvenli (SS) : μ_{SS}

$$\mu_{SS}(t_s) = A(t_s, -T_s, -T_s)$$

Güvenlik indeksi matlab kodu:

(a) $\mu_{SD}(t_s) = B(t_s, 0.0, -T_s)$:

Denkleminde tanımlandığı gibi μ_{SD} üyelik fonksiyonu, t_s değişkeni ve ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{SD} üyelik fonksiyonun kaynak kodu:

```
function out = MF_SD(x,params)
    a = params(1);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_B(x(i),[0 (-1.*a)]);
    end
    out = y;
end
```

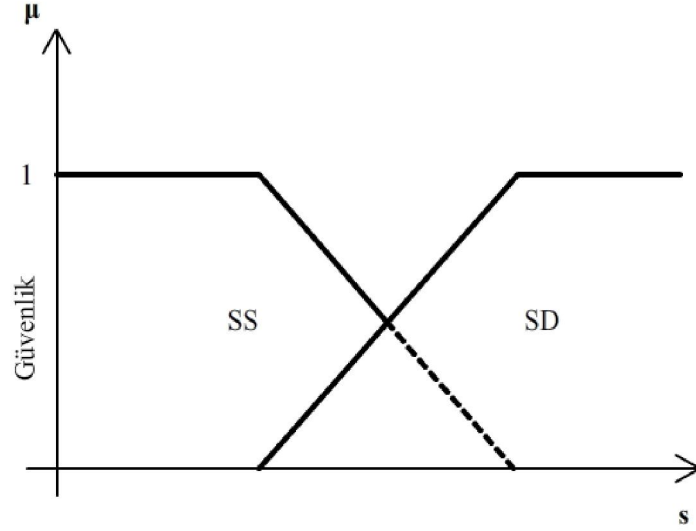
(b) $\mu_{SS}(t_s) = A(t_s, -T_s, -T_s)$:

Denkleminde tanımlandığı gibi μ_{SS} üyelik fonksiyonu, t_s değişkeni ve ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{SS} üyelik fonksiyonun kaynak kodu:


```

function out = MF_SS(x,params)
    a = params(1);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_A(x(i),[(-1.*a) (-1.*a)]);
    end
    out = y;
end

```



Şekil 5.9 Güvenlik İndeksi Üyelik Fonksiyonu

5.9.4. İşleme zamanı performansı indeksi (Running Time Performance)

TASC'e göre oluşturulmuş tren hız başlama noktasının zamanı hesaplanır.

(a) TASC bölgesi içinde (RT) : μ_{RT}

$$\mu_{RT}(t_z) = B(t_z, 0.0, -R_t)$$

(b) Güvenli (SS) : μ_{RF}

$$\mu_{RF}(t_z) = A(t_z, 2R_t, -R_t)$$

İşleme zamanı performansı matlab kodu:

(a) $\mu_{RT}(t_z) = B(t_z, 0.0, -R_t)$

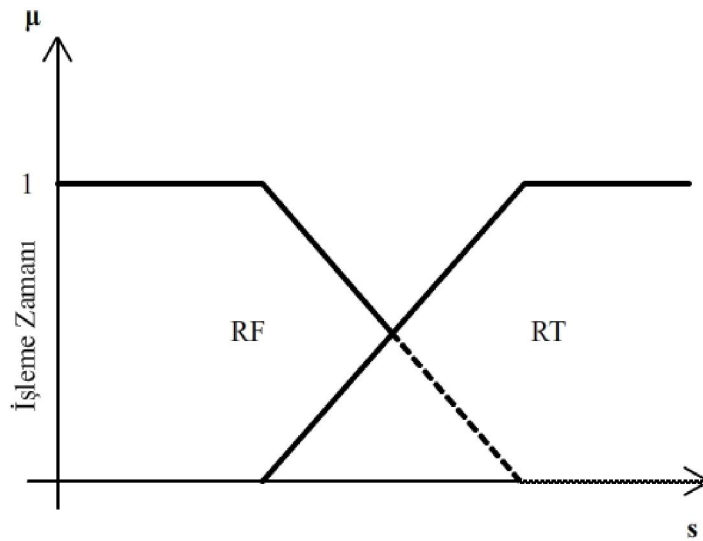
Denkleminde tanımlandığı gibi μ_{RT} üyelik fonksiyonu, t_z değişkeni ve ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{RT} üyelik fonksiyonunun kaynak kodu:

```
function out = MF_RT(x,params)
    a = params(1);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_B(x(i),[0 (-1.*a)]);
    end
    out = y;
end
```

$$(b) \quad \mu_{RF}(t_z) = A(t_z, 2R_t, -R_t)$$

Denkleminde tanımlandığı gibi μ_{RF} üyelik fonksiyonu, t_z değişkeni ve ilgili parametre değerlerin bulunduğu A fonksiyonu ile tanımlansın. μ_{RF} üyelik fonksiyonunun kaynak kodu:

```
function out = MF_RF(x,params)
    a = params(1);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_A(x(i),[2.*a (-1.*a)]);
    end
    out = y;
end
```



Şekil 5.10 İşleme Zamanı İndeksi Üyelik Fonksiyonu

5.9.5. Duruş Mesafesi Performans İndeksi (Stop Gap Performance)

Hedef konum ile hesaplanan konum arasındaki farklılık duruş mesafesi performans indeksi ile hesaplanır.

(a) Oldukça iyi duruş (GG) : μ_{GG}

$$\mu_{GG}(X_p(N)) = C(X_p(N), X_t - X_e, X_e, X_t + X_e, X_e)$$

(b) Güvenli duruş (GA) : μ_{GA}

$$\mu_{GA}(X_p(N)) = C(X_p(N), X_t, X_e, X_t, X_e)$$

Duruş mesafesi performansı matlab kodu:

$$(a) \mu_{GG}(X_p(N)) = C(X_p(N), X_t - X_e, X_e, X_t + X_e, X_e)$$

Denkleminde tanımlandığı gibi μ_{GG} üyelik fonksiyonu, $X_p(N)$ değişkeni ve ilgili parametre değerlerin bulunduğu C fonksiyonu ile tanımlansın. μ_{GG} üyelik fonksiyonunun kaynak kodu:

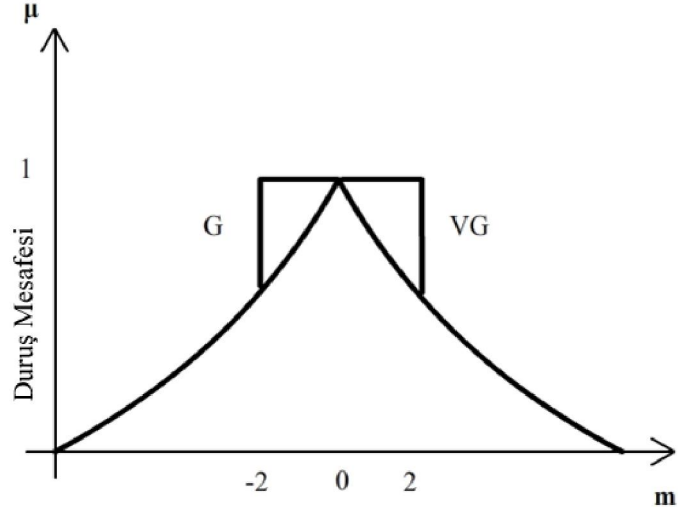
```
function out = MF_GG(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_C(x(i),[a-b b a+b b]);
    end
    out = y;
end
```

$$(b) \mu_{GA}(X_p(N)) = C(X_p(N), X_t, X_e, X_t, X_e)$$

Denkleminde tanımlandığı gibi μ_{GA} üyelik fonksiyonu, $X_p(N)$ değişkeni ve ilgili parametre değerlerin bulunduğu C fonksiyonu ile tanımlansın. μ_{GA} üyelik fonksiyonunun kaynak kodu:

```
function out = MF_GA(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_C(x(i),[a b a b]);
    end
```

end
out = y;
end



Şekil 5.11 Duruş Mesafesi İndeksi Üyelik Fonksiyonu

5.9.6. Enerji tasarruf performans indeksi (Energy Saving Performance)

Lokomotif çalışması için harcanan enerji zaman sınırlarında enerji performans indeksi hesaplanarak bulunur. Lokomotifin harcadığı enerji miktarını en alt seviyede tutmak için kullanılır.

(a) Enerji Tasarflu Çalışma (ES) : μ_{ES}

$$\mu_{ES} = B(x(t), x_k, -E_x)$$

(b) Enerji Tasarufsuz Çalışma (EN) : μ_{EN}

$$\mu_{EN} = A(x(t), x_k + E_x, -E_x)$$

Enerji tasarruf performansı matlab kodu:

(a) $\mu_{ES} = B(x(t), x_k, -E_x)$

Denkleminde tanımlandığı gibi μ_{ES} üyelik fonksiyonu, ilgili parametre değerlerin bulunduğu B fonksiyonu ile tanımlansın. μ_{ES} üyelik fonksiyonun kaynak kodu:

```

function out = MF_ES(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_B(x(i),[a (-1.*b)]);
    end
    out = y;
end

```

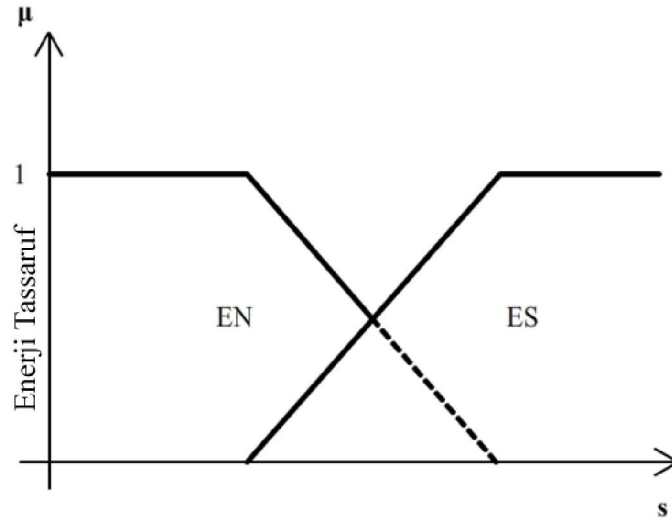
$$(b) \mu_{EN} = A(x(t), x_k + E_x, -E_x)$$

Denkleminde tanımlandığı gibi μ_{EN} üyelik fonksiyonu, ilgili parametre değerlerin bulunduğu A fonksiyonu ile tanımlansın. μ_{EN} üyelik fonksiyonunun kaynak kodu:

```

function out = MF_EN(x,params)
    a = params(1);
    b = params(2);
    y = zeros(length(x),1);
    for i=1:length(x)
        y(i) = FN_A(x(i),[a+b (-1.*b)]);
    end
    out = y;
end

```



Şekil 5.12 Enerji Tassaruf İndeksi Üyelik Fonksiyonu

5.10. Matlab’de Otomatik Tren Yönetim Sisteminin Tanımlanması

Matlab’de bulanık çıkarım sistemi FIS=NEWFIS(FISNAME, FISTYPE, andMethod, orMethod, impMethod, aggMethod, defuzzMethod) komutu ile oluşturulur. Sabit Hız Kontrol Bölgesi ve Duruş Mesafesi Kontrol Bölgesi olmak üzere iki bölümden oluşan uygulamadaki kod aşağıdaki gibidir:

5.10.1. Sabit hız kontrol bölgesi

```
ATOFIS_CSC=newfis('ATO_FuzzyLogicCSC', 'sugeno', 'min', 'max', 'prod', 'sum',  
'wtaver');
```

5.10.2. Güvenlik üyelik fonksiyonu

```
ATOFIS_CSC=addvar(ATOFIS_CSC, 'input', 'Safety', [-1000 6000]);  
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 1, 'SAFE(SS)', 'MF_SS', 3);  
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 1, 'DANGER(SD)', 'MF_SD', 3);
```

5.10.3. Konfor üyelik fonksiyonu

```
ATOFIS_CSC=addvar(ATOFIS_CSC, 'input', 'Comfort', [-1000 1000]);  
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 2, 'POORCOMFORT(CB)',  
'MF_CB', [Nc 50]);  
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 2, 'GOODCOMFORT(CG)',  
'MF_CG', [Nc 50]);
```

5.10.4. Enerji tasarufu üyelik fonksiyonu

```
ATOFIS_CSC=addvar(ATOFIS_CSC, 'input', 'EnergySave', [-1000 6000]);  
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 3, 'NOENERGYSAFE(EN)',  
'MF_EN', [Xk 5]);
```

ATOFIS_CSC=addmf(ATOFIS_CSC,'input', 3, 'ENERGYSAFE(ES)', 'MF_ES', [Xk 5]);

5.10.5. İzlenebilirlik üyelik fonksiyonu

ATOFIS_CSC=addvar(ATOFIS_CSC, 'input', 'Traceability', [0 3000]);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 4, 'ACCURATETRACE(TA)', 'MF_TA', [Vt Ve]);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 4, 'GOODTRACE(TG)', 'MF_TG', [Vt Ve V_t]);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'input', 4, 'LOWSPEED(TL)', 'MF_TL', Vt);

5.10.6. Çıkış değişkeni parametreleri

ATOFIS_CSC=addvar(ATOFIS_CSC, 'output', 'POWER NOTCH', [0 8]);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_0', 'constant', 0);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_1', 'constant', 1);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_2', 'constant', 2);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_3', 'constant', 3);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_4', 'constant', 4);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_5', 'constant', 5);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_6', 'constant', 6);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 1, 'PN_7', 'constant', 7);
ATOFIS_CSC=addvar(ATOFIS_CSC, 'output', 'BREAK NOTCH', [-10 0]);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_0', 'constant', 0);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_1', 'constant', -1);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_2', 'constant', -2);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_3', 'constant', -3);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_4', 'constant', -4);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_5', 'constant', -5);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_6', 'constant', -6);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_7', 'constant', -7);

ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_8', 'constant', -8);
ATOFIS_CSC=addmf(ATOFIS_CSC, 'output', 2, 'BN_9', 'constant', -9);

5.10.7. Kural listesi

ruleListtemp=[
2 0 0 0 1 abs(floor((Npp-9)/2)) 1 1;
1 2 2 0 0 1 1 1 1;
1 2 0 3 0 8 1 1 1;
1 2 0 1 0 1 2 1 1];

5.10.8. Kuralların eklenmesi

ATOFIS_CSC=addrule(ATOFIS_CSC,ruleListtemp);
ruleview(ATOFIS_CSC);

5.10.9. Giriş değişkenleri atamaları

Safety=ts;
Comfort=tc;
EnergySave=X_t;
Traceability=Vp;

5.10.10. Bulanık çıkarımının hesaplanması

fuz=evalfis([Safety Comfort EnergySave Traceability StopGap], ATOFIS_CSC);

5.10.11. Güç kademesi

Npp=floor(fuz(1));

5.10.12. Fren kademesi

NBp=floor(fuz(2));

5.10.13. Duruş mesafesi kontrol bölgesi

if(tz<2)

ATOFIS_TCSC=newfis('ATO_FuzzyLogicTCSC','sugeno','min','max','prod','sum','wtaver');

5.10.14. Çalışma zamanı üyelik fonksiyonu

ATOFIS_TCSC=addvar(ATOFIS_TCSC, 'input', 'RunningTime', [0 10000]);

ATOFIS_TCSC=addmf(ATOFIS_TCSC, 'input', 1, 'INTASCZONE(RT)', 'MF_RT', 2);

ATOFIS_TCSC=addmf(ATOFIS_TCSC, 'input', 1, 'NOTINTASCZONE(RF)', 'MF_RF', 2);

5.10.15. Duruş mesafesi üyelik fonksiyonu

ATOFIS_TCSC=addvar(ATOFIS_TCSC, 'input', 'StopGap', [-100 10000]);

ATOFIS_TCSC=addmf(ATOFIS_TCSC, 'input', 2, 'ACCURATESTOP(GA)', 'MF_GA', [Xt Xe]);

ATOFIS_TCSC=addmf(ATOFIS_TCSC, 'input', 2, 'GOODSTOP(GG)', 'MF_GG', [Xt Xe]);

5.10.16. Konfor üyelik fonksiyonu

ATOFIS_TCSC=addvar(ATOFIS_TCSC, 'input', 'Comfort', [-1000 10000]);

ATOFIS_TCSC=addmf(ATOFIS_TCSC, 'input', 3, 'POORCOMFORT(CB)', 'MF_C B', [Nc 50]);

*ATOFIS_TCSC=addmf(ATOFIS_TCSC,'input',3,'GOODCOMFORT(CG)','MF_C
G',[Nc 50]);*

5.10.17. Çıkış değişkeni parametreleri

*ATOFIS_TCSC=addvar(ATOFIS_TCSC,'output','POWER NOTCH',[0 8]);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_0','constant',0);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_1','constant',1);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_2','constant',2);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_3','constant',3);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_4','constant',4);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_5','constant',5);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_6','constant',6);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',1,'PN_7','constant',7);*

*ATOFIS_TCSC=addvar(ATOFIS_TCSC,'output','BREAK NOTCH',[-10 0]);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_0','constant',0);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_1','constant',-1);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_2','constant',-2);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_3','constant',-3);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_4','constant',-4);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_5','constant',-5);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_6','constant',-6);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_7','constant',-7);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_8','constant',-8);
ATOFIS_TCSC=addmf(ATOFIS_TCSC,'output',2,'BN_9','constant',-9);*

5.10.18. Kural listesi

*ruleListtemp=[
1 2 0 1 1 1 1;
2 0 2 1 1 1 1;*

```
1 0 2 1 2 1 1;  
1 1 2 1 2 1 1];
```

5.10.19. Kuralların eklenmesi

```
ATOFIS_TCSC=addrule(ATOFIS_TCSC,ruleListtemp);  
% ruleview(ATOFIS_TCSC);
```

5.10.20. Giriş değişkenleri atamaları

```
Comfort=tc;  
RunningTime=tz;  
StopGap=Xp;
```

5.10.21. Bulanık çıkarımının hesaplanması

```
fuz=evalfis([RunningTime StopGap Comfort], ATOFIS_TCSC);
```

5.10.22. Güç kademesi

```
NPp=floor(fuz(1));
```

5.10.23. Fren kademesi

```
NBp=floor(fuz(2));  
end
```

5.10.24. Ana döngü

```
function th=FuzzyCode()  
clc;  
clear;
```

FuzzyConst.X_t=0; % *t* anındaki aldığı mesafe # *x(t)* metre
FuzzyConst.V_t=0; % *t* anındaki hızı # *V(t)* km/sa
FuzzyConst.Xt=50000; % Sonraki istasyonun hedef konumu # *X(t)* metre
FuzzyConst.Vt=60; % Hedef Hız # *Vt* km/sa
FuzzyConst.Tt=2; % Hesaplanan çalışma zamanı # *Tt* saniye
FuzzyConst.Xd=2500; % Maksimum hız limitinin azaldığı konum # *xd* metre
FuzzyConst.ts=1000; % *xd* noktasına erişme zamanı # *ts* saniye
FuzzyConst.Xk=0; % Yokuş aşağı ilerlemenin son konumu # *xk* metre
FuzzyConst.Xz=5000; % TASC bölgesinin başlama noktası # *xz(v)* metre
FuzzyConst.tc=1000; % Son kademe değişikliğinden beri geçen zamanı # *tc*
saniye
FuzzyConst.Nc=0; % Son kademe değişikliğinin derecesi # *Nc*
FuzzyConst.N_t=7; % *t* anındaki kademe kontrolü # *N(t)*
FuzzyConst.NPp=0; % Hesaplanan güç kademe derecesi # *NPp*
FuzzyConst.NBp=0; % Hesaplanan fren kademe derecesi # *NBp*
FuzzyConst.Vp=0; % *Np* kademesi seçildiğinde hesaplanan hız # *Vp(Np)*
km/sa
FuzzyConst.Ve=5; % İzin verilen hızlanma aralığı # *Ve* km/sa
FuzzyConst.Xp=0; % *Np* kademesi seçildiğinde hesaplanan duruş konumu #
Xp(Np) metre
FuzzyConst.Xe=20; % İzin verilen duruş aralığı
FuzzyConst.Sumt=0; % Toplam süre
FuzzyConst.tz=1000;
FuzzyConst.SpeedArray=[]; % Hız dizisi
FuzzyConst.XArray=[]; % Konum dizisi
FuzzyConst.VtArray=[]; % Hedef hız dizisi
FuzzyConst.VsArray=[]; % Hız sınırı dizisi
FuzzyConst.timecount=0; % sayaç

5.10.25. Zamanlayıcı

th=timer('startdelay',1,...

```

    'period',1,...
    'taskstoexecute',150,...
    'executionmode',...
    'fixedrate');
th userdata=FuzzyConst;
th.TimerFcn=@test;
start(th);
function test(obj, ~)

```

5.10.26. Grafik çizimi

```

v.timecount = v.timecount+1;
v.SpeedArray(v.timecount) = v.V_t;
v.VsArray(v.timecount) = v.Vt + 10;
v.VtArray(v.timecount) = v.Vt;
v.XArray(v.timecount) = v.X_t;
if(get(obj,'TasksExecuted')==150)
    plot(v.XArray,v.SpeedArray,v.XArray,v.VtArray,v.XArray+750,v.VsArray);
end

```

5.10.27. ts ve tz zamanlamalarının hesaplanması

```

if(v.V_t~=0)
    v.tz = (v.Xz-v.X_t)./v.V_t;
    if(v.tz<0)
        v.tz = 0;
    end
    v.ts = ((v.Xd-v.X_t)).*(3.6./v.V_t);
    if(v.ts<0)
        v.ts = 0;
    end
end
end

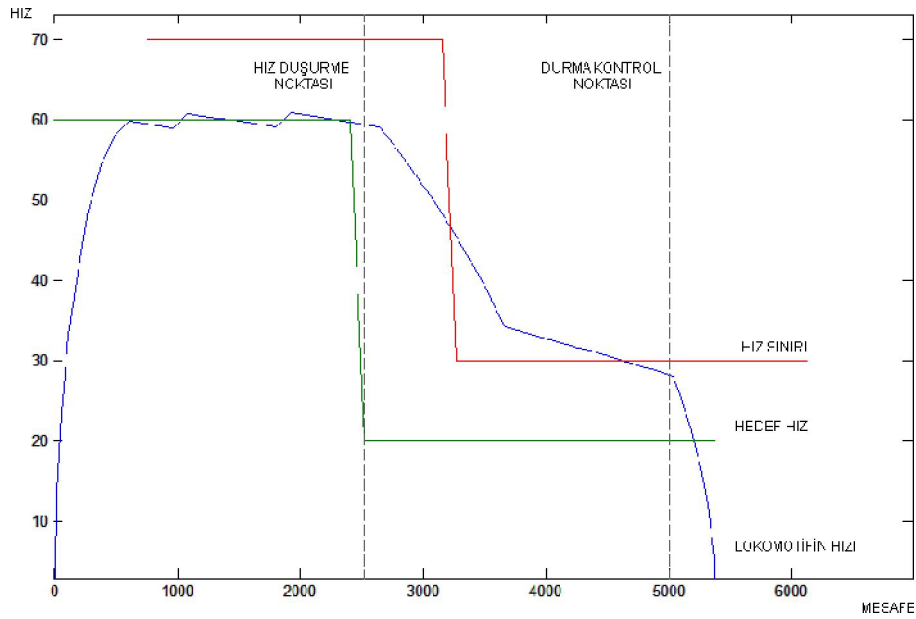
```

5.10.28. Simülasyonun çalıştırılması

```
[v.V_t v.X_t] = Speed_Sim(v.V_t,v.X_t,v.N_t,v.Tt);  
v.Vp=v.V_t;  
[v.NPp v.NBp v.Nc] = callback_fuzzy(v.N_t, v.V_t, v.X_t, v.Xt, v.Vt, v.Xd, v.ts,  
v.Xk, v.tz, v.tc, v.Nc, v.Vp, v.Ve, v.Xp, v.Xe, v.NPp, v.NBp, v.Xz);  
v.N_t = v.NPp+v.NBp;  
obj.userdata = v;
```

5.11. Bulanık Mantık ile Kontrol ve Simülasyon Sonuçları

Bulanık mantık kontrollü Otomatik Tren Yönetim Sistemi simülasyonu 7000 metrelik bir mesafe için tasarlanmıştır. Birinci İstasyon 0. metrede, İkinci İstasyon 5500. metrede bulunmaktadır. İlk hız düşürme noktası 2500 ve durma kontrol noktası 5000 metre olarak belirlenmiştir. Yol eğimi ve kulp, 1 değeri olarak ayarlanmıştır. Lokomotif 5 vagondan oluşmakta ve toplam ağırlığı 125000 kg olarak belirlenmiştir. Bu parametreler ile yapılan simülasyonun Hız – Konum Grafiği aşağıda verilmiştir.



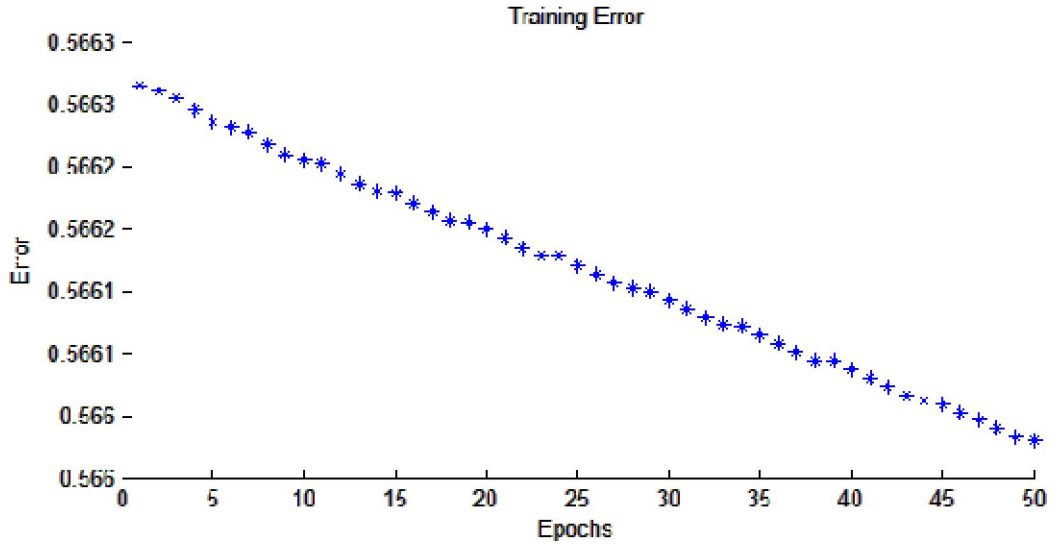
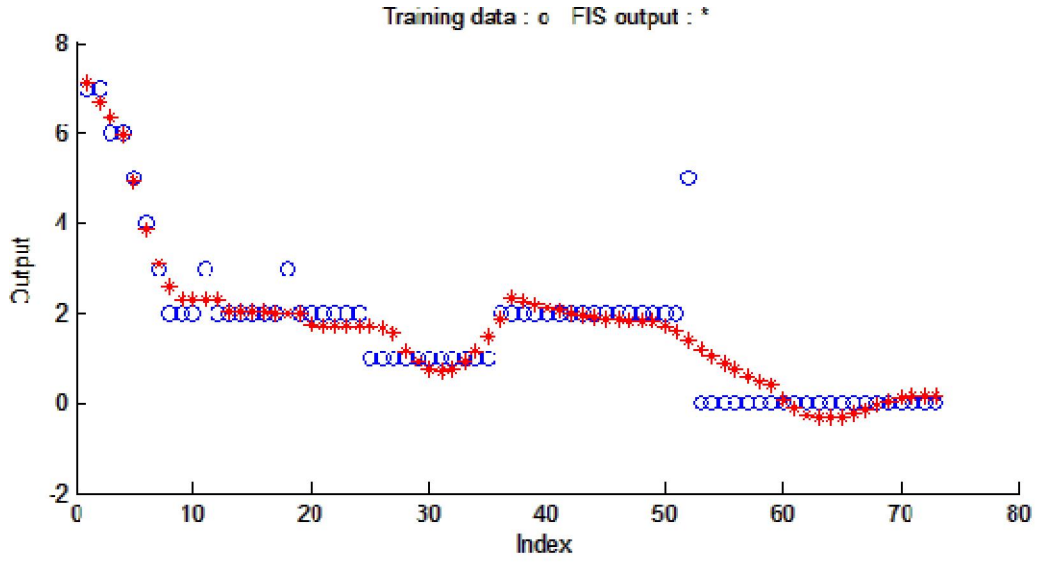
Şekil 5.13 Hız - Konum Grafiği

Hız – Konum grafiğinde görüleceği gibi lokomotifin hız eğrisi hedef hızın etrafında seyretmektedir. Burada hız aşımı olduğunda kademe değişimi arttırılarak, hız düşümü olduğunda ise kademe değişimi azaltılarak gerçek zamanlı kontrol sağlanmaktadır. İlk hız düşüm noktasından sonra konfor ve güvenlik indekslerinin etkinliğinde hedef hıza yakınsama yapılmaktadır. Durma kontrol noktasından sonra ise güvenlik indeksi ile en iyi duruş gerçekleştirilmektedir.

Şu ana kadar bulanık mantık ile kurallar tanımlanmış ve bu kurallar çerçevesinde oluşturulan algoritma ile trenin hareketi sağlanmıştır. Ancak kurulan üyelik fonksiyonları ve oluşturulan kural tabanı her yerde ve koşulda sistemi tam olarak tanımlayamayabilir. Özellikle ülkemizdeki demiryolu ağının bölgelere göre farklılık göstermesi ve çeşitlilikten dolayı bu kuralları eksiksiz tanımlamak oldukça zordur. Bu gibi durumlarda belli girişlere verilmesi gereken tepki biliniyorsa, bir başka deyişle sistem hakkında elde yeterli ve sağlıklı veriler varsa, Anfis sistemi kullanılarak denetçinin iyileştirilmesi sağlanabilir.

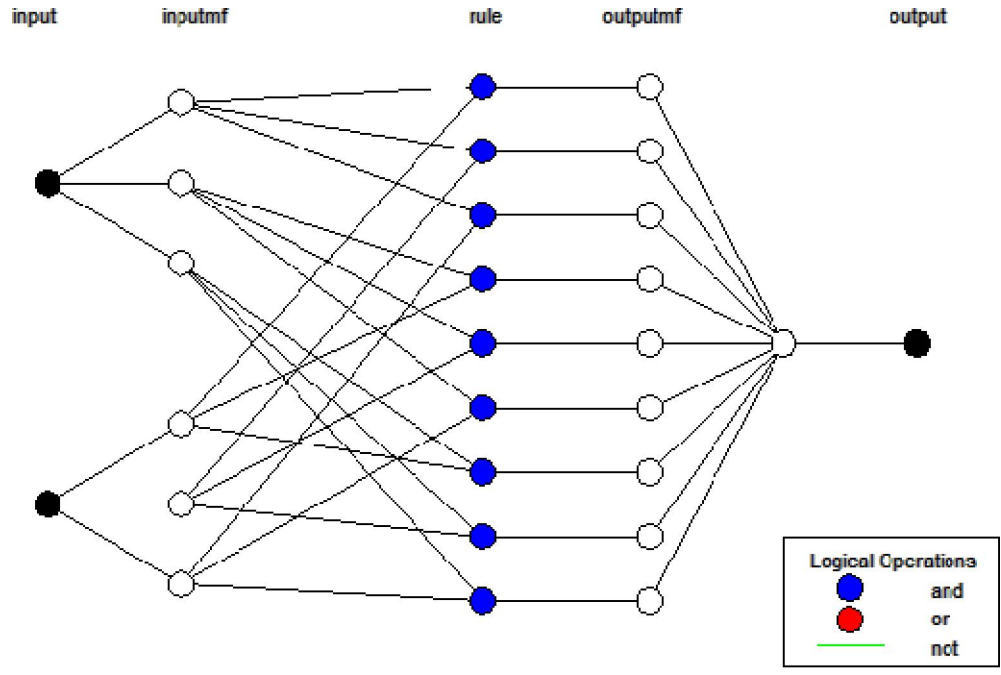
5.12. Anfis ile Kontrol ve Simülasyon Sonuçları

Bulanık Mantık kontrollü simülasyonundan elde edilen t anındaki hız, konum, seçilen güç kademesi ve fren kademesi MATLAB ortamında Anfis aracı ile modellenmiştir. Eğitim verisi ile oluşturulan bulanık çıkarım sistemi 3 yamuksal üyelik fonksiyonu ve sabit çıkış değeri ile 50 adımda eğitilmiş ve test sonuçları aşağıdaki grafiklerde sunulmuştur.

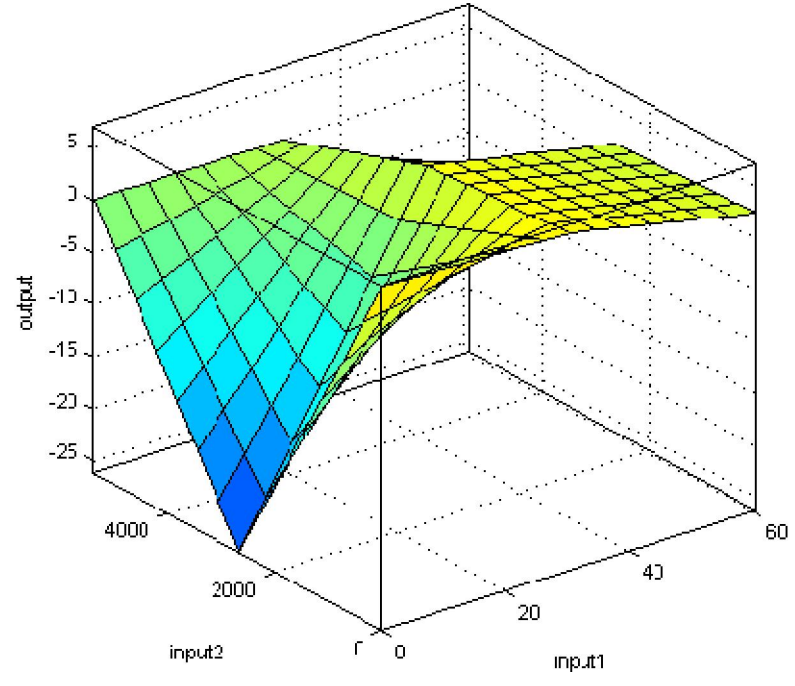


Şekil 5.15 Bulanık Mantık Kontrollünün Hata Grafiği

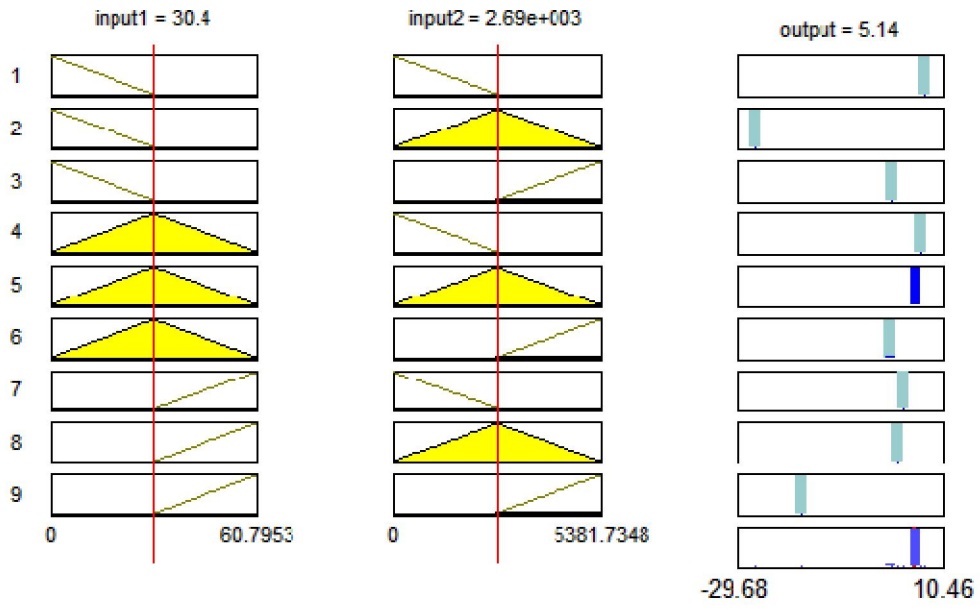
Hata Grafiğinde görüleceği gibi 50 adım sonunda 0 yakınsama sağlanmıştır. Anfis modeli 2 girişli ve tek çıkışlı olarak belirlenmiştir. Giriş değişkenleri 3 yamuksal üyelik fonksiyonu ile tanımlanmıştır. Mimari yapı aşağıdaki grafikte verilmiştir.



gibi

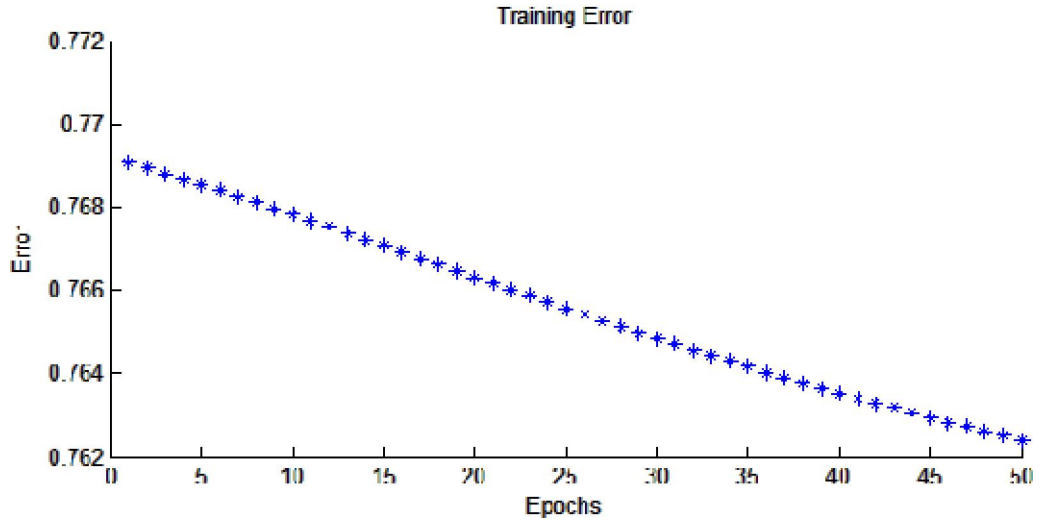


Şekil 5.17 Bulanık Mantık Kontrollü Yüzey Grafiği



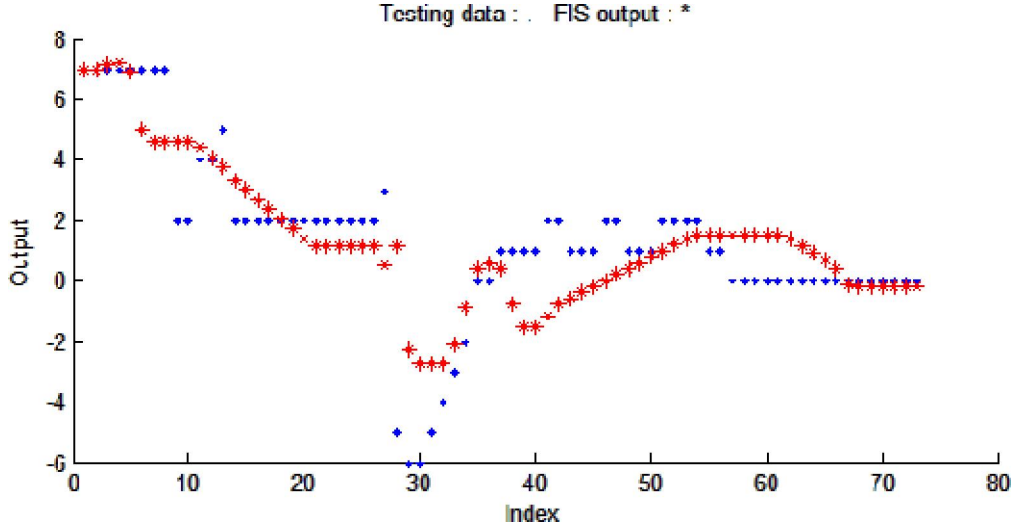
m
eğ
aş

a
1
i



Şekil 5.19 Anfis Kontrollü Hata Grafiği

Makinistlerin yaptığı gerçek hayattaki kontrol, bulanık mantık kontrolü ile farklılık göstermektedir. Aşağıdaki grafikte görülebileceği gibi 30 değerinde, makinistlerin seçtikleri kademeler daha düşüktür.



Şekil 5.20 Anfis Kontrollü Test Grafiği

5.13. Otomatik Tren Yönetim Sistemi Uygulaması Sonuçları

Otomatik Tren Yönetimi ilk olarak Bulanık Mantık Çıkarım Sistemi ile modellenmiş ve simülasyonu gerçekleştirilmiş; aynı uygulama Anfis yöntemi ile de yapılmıştır. Yöntemlerin simülasyon sonuçları incelendiğinde, bazı kritik noktalarda seçilen kademe bilgilerinin güvenliğini tehlikeye sokacak farklılıklara sahip olduğu söylenebilir. Özellikle Şekil 5.14 ve 5.20 karşılaştırıldığında, seçilen kademelerden anlaşılabilen gibi trenlerin aslında o bölgede yavaşladıkları görülmektedir. Bulanık mantık kontrolü ise aynı bölgede daha yüksek hızda seyretmektedir. Bu durumda güvenlik açısından çok önemli sorunlar oluşmaktadır. Bu sorunun çözümü, bulanık mantık kontrolünün makinistlerin deneyimlerini yansıtan Anfis kontrolü ile birlikte kullanılmasıdır. Böylece Anfis bulanık mantık kontrolünün güvenlik açısından denetleyicisi olacaktır. Bunun yanı sıra konfor ve seyir süresinin iyileştirici yanları ile Otomatik Tren Yönetim Sistemi ülkemiz koşullarında güvenle kullanılacaktır.

6. SONUÇ VE ÖNERİLER

Bu tezde, Otomatik Tren Yönetim Sistemi hem bulanık mantık hem de adaptif ağ yapısına dayalı bulanık çıkarım sistemi ile modellenmiş ve simülasyonu yapılmıştır. Modelleme ve simülasyon MATLAB ortamında programlanmış ve elde edilen sonuçlar sunulmuştur. Bulanık Mantık, Otomatik Tren Yönetim Sistemi simülasyonlarında lokomotif üzeri ekipmanlar ile kontrol sağlayarak, uygun koşullarda doğru karar verme becerisini göstermiştir. Bulanık Mantık kontrolü ile trenlerin konforlu, ekonomik ve hızlı seferlerin sağlanmasında kullanılabilecek önemli bir sistem oluşturulmuştur. Ancak Bulanık Mantık Kontrollü Otomatik Tren Yönetim Sistemi, çevresel koşulların kontrol altında tutulabildiği ortamlarda başarılı olabilecektir. Örneğin yeraltı raylı sistemlerde, metrolarda ve Bulanık Mantık kontrolünün uygulandığı Sendai metrosu gibi. Buna karşın şehirler arası raylı ulaşım sistemlerinde koşullar oldukça değişkendir. Bundan dolayı kurulan üyelik fonksiyonları ve oluşturulan kural tabanı her yerde ve koşulda sistemi tam olarak tanımlayamayabilir. Bu gibi durumlarda belli girişlere verilmesi gereken tepki biliniyorsa, bir başka deyişle sistem hakkında elde yeterli ve sağlıklı veriler varsa, Anfis sistemi kullanılarak tüm denetçinin kontrolü otomatikleştirilebilir. Bu mantıktan yola çıkılarak mevcut bulanık mantık ile oluşturulmuş kurallara ek olarak aynı güzergah üzerinde yolculuk yapan makinistlerin deneyimleri ve alışkanlıkları da dahil edilirse bulanık mantığın ürettiği kontrol parametrelerini denetleyen ve güvenliği arttıran bir sistem geliştirilmiş ve lokomotifin kendi kendine karar vermesi, insan ile kullanımına olabildiğince benzetilmesi gerçekleştirilmiştir.

Sonuçta Otomatik Tren Yönetim Sistemi Bulanık Mantık Kontrolü tek başına yetersiz bir sistem oluşturmaktadır. Her koşul ve ortamın özellikleri, makinistin tecrübe ve deneyimleri Anfis modellemesi yardımı ile sisteme dahil edilerek daha güvenli, konforlu, ekonomik ve hızlı işleyen bir sistem yapılabileceği söylenebilir.

KAYNAKLAR

- [1] Zadeh, L. A., "Fuzzy sets", *Information and Control*, **8**, 338-353, 1965.
- [2] Klir, G. J., Yuan, B., *Fuzzy Sets and Fuzzy Logic Theory and Applications*, Prentice Hall, USA, 1995.
- [3] Çağman N., *Olasılık ile bulanık kümelerin karşılaştırılması*, *Bilim ve Ütopya*, Sayı:149, Yıl:13, Sayfa: 51-54, Kasım 2006.
- [4] Babuska, R., *Fuzzy Modeling for Control*, Kluwer Academic Publisher, 1998.
- [5] Zimmermann, H.J., *Fuzzy Set Theory and Its Applications*, Kluwer, 1991.
- [6] Semed, M.; *Dünya Dahilersiz Yaşabilmir*, Azerbaycan Bilimler Akademisi Yay., Bakü, 2000.
- [7] Öztemel, E.; *Yapay Sinir Ağları*, Papatya Yay. I.Baskı, İstanbul, 2003.
- [8] Alan, A. Y., *Nispi Mantık (Fuzzy Logic)*, International Seminar Group, Germany, 2003.
- [9] Zadeh, L. A., Yager R. R., *An Introduction to fuzzy Logic Applications in Intelligent systems*, Kluwer Academic Publishery, Boston/London, 78-84, 1992.
- [10] Fausett L., *Fundamentals of Neural Networks*, Prentice Hall, 328-329, 1994.
- [11] Haykin S., *Neural Networks*, Macmillan Collage Publishing Company, 181-182, 1994.
- [12] Lin C., Lee G., *Neural Fuzzy Systems*, Prentice Hall, 236-240, 242, 445-448, 1996.
- [13] Jang J. S. R., *ANFIS: Adaptive-Network-Based Fuzzy Inference System*, University of California, Berkeley, 1993.
- [14] Jang, J. S. R. ve Sun, C. T., *Neuro-Fuzzy Modeling and Control*, Proc. of the IEEE Special Issue on Fuzzy Logic in Engineering Applications, 83 (3): 378-406, 1995.
- [15] Takagiand T. and Sugeno M., *Derivation of Fuzzy Control Rules From Human Operator's Control Actions*. Proc. of the IFAC Symp.on Fuzzy Information, Knowledge Representation and Decision Analysis, 55-60, 1983.

- [16] S. Yasunobu, S.Miyamoto and H.Ihara Fuzzy Control For Automatic Train Operation System, Systems Development Laboratory, Hitachi, Japan, 1983.
- [17] S. Yasunobu, S.Miyamoto Automatic Train Operation System By Predictive Fuzzy Control, Systems Development Laboratory, Hitachi, Japan, 1985.
- [18] H. Oshima, S.Yasunobu, S. Sekino Automatic Train Operation System Based on Predictive Fuzzy Control, International Workshop on Artificial Intelligent for Industrial Application, 1988.

Ek-1 Eğitim Verisi*Bulanık Mantık Algoritması**ANFIS Algoritması*

No	Hız	Konum	Seçilen Kademe
1	0	0	7
2	13,72606758	13,72606758	7
3	23,45213526	50,90427042	6
4	33,17820308	107,5346088	6
5	40,90427106	181,6170829	5
6	48,63033921	271,1516932	4
7	54,35640756	374,1384399	3
8	58,08247608	486,5773236	2
9	59,80854471	604,4683444	2
10	59,53461339	723,8115025	2
11	59,26068207	842,6067979	3
12	58,98675074	960,8542308	2
13	60,7128194	1080,553801	2
14	60,43888811	1201,705508	2
15	60,16495682	1322,309353	2
16	59,89102551	1442,365336	2
17	59,6170942	1561,873455	2
18	59,34316288	1680,833712	3
19	59,06923155	1799,246107	2
20	60,79530021	1919,110639	2
21	60,52136893	2040,427308	2
22	60,24743763	2161,196114	2
23	59,97350633	2281,417058	2
24	59,69957502	2401,09014	2
25	59,42564371	2520,215358	1
26	59,15171238	2638,792714	1
27	56,87778104	2754,822208	1
28	54,60384963	2866,303839	1
29	52,32991816	2973,237606	1
30	50,05598661	3075,623511	1
31	47,782055	3173,461553	1
32	45,50812333	3266,751731	1
33	43,2341916	3355,494046	1
34	40,96025981	3439,688497	1
35	38,68632796	3519,335085	1
36	36,41239606	3594,433809	2
37	34,13846412	3664,984669	2

Hız	Konum	Seçilen Kademe
0	0	7
10	50	7
20	100	7
30	150	7
40	200	7
50	300	7
60	400	7
60	500	7
60	600	2
60	700	2
60	850	4
60	960	4
60	1050	5
60	1200	2
60	1300	2
60	1400	2
60	1500	2
60	1600	2
60	1700	2
60	1800	2
60	1900	2
60	2000	2
60	2100	2
60	2200	2
60	2300	2
60	2400	2
58	2500	1
56	2600	1
55	2700	1
50	2800	1
50	2900	1
50	3000	1
45	3100	1
45	3200	1
40	3300	1
40	3400	1
35	3500	1

38	33,86453212	3732,987666	2
39	33,59060012	3800,442798	2
40	33,31666811	3867,350066	2
41	33,0427361	3933,70947	2
42	32,76880408	3999,52101	2
43	32,49487206	4064,784687	2
44	32,22094003	4129,500499	2
45	31,947008	4193,668447	2
46	31,67307596	4257,288531	2
47	31,39914391	4320,360751	2
48	31,12521186	4382,885106	2
49	30,85127981	4444,861598	2
50	30,57734775	4506,290226	2
51	30,30341568	4567,170989	2
52	30,02948361	4627,503888	5
53	29,75555154	4687,288923	0
54	29,48161946	4746,526094	0
55	29,20768737	4805,215401	0
56	28,93375528	4863,356844	0
57	28,65982318	4920,950422	0
58	28,38589108	4977,996137	0
59	28,11195898	5034,493987	0
60	25,83802687	5088,443973	0
61	23,56409472	5137,846094	0
62	21,29016253	5182,700351	0
63	19,01623031	5223,006744	0
64	16,74229806	5258,765273	0
65	14,46836578	5289,975936	0
66	12,19443347	5316,638736	0
67	9,920501141	5338,75367	0
68	7,646568787	5356,32074	0
69	5,372636413	5369,339945	0
70	3,098704023	5377,811286	0
71	0,824771617	5381,734761	0
72	0	5381,734761	0
73	0	5381,734761	0

35	3600	1
35	3700	1
30	3800	1
30	3900	2
30	4000	2
30	4050	1
30	4100	1
30	4150	1
30	4200	2
30	4250	2
28	4300	1
28	4350	1
30	4400	1
30	4450	2
30	4500	2
30	4550	2
30	4600	2
30	4650	1
30	4700	1
30	4750	0
30	4800	0
28	4850	0
26	4900	0
24	4950	0
22	5000	0
20	5050	0
18	5100	0
16	5150	0
14	5200	0
10	5250	0
5	5300	0
0	5350	0
0	5370	0
0	5380	0
0	5380	0
0	5380	0