

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

KESİKLİ BİR BİYOREAKTÖRDE YAPAY SİNİR AĞLARININ KULLANIMI

Tufan METE

KİMYA MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2008**

Her hakkı saklıdır

TEZ ONAYI

Tufan METE tarafından hazırlanan “**Kesikli Bir Biyoreaktörde Yapay Sinir Ağlarının kullanımı**” adlı tez çalışması 01/07/2008 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Kimya Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Gülay ÖZKAN

Jüri Üyeleri :

Başkan : Prof. Dr. Canan CABBAR

Gazi Üniversitesi Mühendislik Fakültesi Kimya Mühendisliği A.B.D

Üye : Doç. Dr. Gülay ÖZKAN

Ankara Ünivesitesi Mühendislik Fakültesi Kimya Mühendisliği A.B.D

Üye : Doç. Dr. Zehra ZEYBEK

Ankara Ünivesitesi Mühendislik Fakültesi Kimya Mühendisliği A.B.D

Yukarıdaki sonucu onaylarım.

Prof.Dr.Orhan ATAKOL

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

KESİKLİ BİR BİYOREAKTÖRDE YAPAY SİNİR AĞLARININ KULLANIMI

Tufan METE

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Kimya Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Gülay ÖZKAN

Bu çalışmada, Yapay Sinir Ağları (YSA) kullanılarak *Saccharomyces cerevisiae* mikroorganizmasının çoğaltıldığı kesikli olarak işletilen bir biyoreaktörde farklı sıcaklık ve pH değerinde reaksiyon hızı tahmin edilmiştir ve bulunan sonuçlar deneysel verilerle karşılaştırılmıştır. Tahmin edilen reaksiyon hızlarının deneysel reaksiyon hızlarına yakın olduğu görülmüştür. Ayrıca yapay sinir ağları kullanılarak biyoreaktörde kütle aktarım katsayısı tahmin edilmiştir. Biyoreaktörde çözünmüş oksijen derişim kontrolü yine yapay sinir ağı kullanılarak gerçekleştirilmiştir. Sistemin matematiksel modellenmesi için iki ayrı model önerilmiştir. Bu modeller diferansiyel denklem (governing) ve polinom tipi (ARMAX) modeldir. Modellerin geçerliliği deneysel veriler kullanılarak test edilmiştir. Sistemin çözünmüş oksijen kontrolü her bir model için MATLAB ortamında ayrı ayrı sağlanmıştır. Sistemde kontrol edici değişkeni olarak biyoreaktörde çözünmüş oksijen derişim, ayarlanabilen değişken olarak hava akış hızı seçilmiştir. Kontrol algoritması için MATLAB ortamında prosesin YSA modeli tasarlanarak Levenberg Marquardt Algoritması ile eğitilmiştir. Bu model temel alınarak, set noktası değişiminde biyoreaktörde çözünmüş oksijen derişim kontrolü; YSA Temelli Model Öngörmeli Kontrol, NARMA-L2 (Geri beslemeli linerizasyon) Kontrol ve Model Referanslı Kontrol algoritmaları ile gerçekleştirilmiştir.

Temmuz 2008, 129 sayfa

Anahtar Kelimeler: Yapay sinir ağı, YSA temelli model öngörmeli kontrol, NARMA-L2, model referanslı kontrol

ABSTRACT

Master Thesis

USE OF ARTIFICIAL NEURAL NETWORKS (ANN) IN A BATCH BIOREACTOR

Tufan METE

Ankara University
Graduate School of Natural and Applied Sciences
Department of Chemical Engineering

Supervisor: Assoc.Prof.Dr. Gülay ÖZKAN

In this work, increase of *S. Cerevisiae* microorganism in batch bioreactor at different temperature and pH value was estimated reaction rates by using neuralnetwork and finding result was compared of experimental result. It was seen that prediction reaction rate is alike experimantal result. In addition by using neural network in bioreactor, mass transfer coefficient was estimated. In batch bioreactor, dissolved oxygen concentration control was realized by using neural network. System was modeled two model. These are differential equations (governing) and polynomial (ARMAX) model. Model verification was tested by using experimental data. Dissolved oxygen was controled using matlab for every model. At system, output variable is dissolved oxygen concentration; adjust variable is air flow rate was chosen. For control algorithm in matlab, neural network model of process was designed and neural network model was trained with levenberg Marguardt algorithm. This model was base at different set point, dissolved oxygen control was realized using predictive control, NARMA-L2 (feedback linearization), model reference control.

July 2008, 129 pages

Key Words: Artificial neural network, predictive control, NARMA-L2 (Feedback linearization) control, model reference control

TEŞEKKÜR

Çalışmalarım boyunca yardım ve katkıları ile beni yönlendiren hocam Doç. Dr. Gülay ÖZKAN'a yine kıymetli tecrübelerinden yararlandığım Prof. Dr. Mustafa ALPBAZ'a ayrıca laboratuarda görevli tüm arkadaşlarıma ve çalışmalarım sırasında maddi, manevi fedakarlıklarından ötürü aileme teşekkürü bir borç bilirim.

Tufan METE

Ankara, Temmuz 2008

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER DİZİNİ.....	vii
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ.....	xii
1.GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	6
3. MATERYAL VE YÖNTEMLER.....	11
3.1 Materyal.....	11
3.1.1 Mikroorganizma.....	11
3.1.2 Katı besiy ortamı.....	11
3.1.3 Sıvı besiy ortamı.....	12
3.1.4 Ölçek büyütme.....	12
3.1.5 Deney düzeneği.....	12
3.2 Yöntem.....	14
3.2.1 Yapay sinir ağları.....	14
3.2.1.1Yapay sinir ağlarının tanımı.....	14
3.2.1.2 Yapay sinir ağlarının tarihi gelişimi.....	15
3.2.1.3 Yapay sinir ağlarının uygulama alanları.....	16
3.2.1.4 Yapay sinir ağlarının özellikleri.....	18
3.2.1.4.1 Bellek ve genelleme.....	19
3.2.1.4.2 Eksik bilgilerle çalışabilme ve hata toleransı.....	20
3.2.1.4.3 Nonlineerlik ve paralellik.....	21
3.2.1.5 Biyolojik sinir sistemi ve nöron yapısı.....	21
3.2.1.6 Yapay nöron.....	23
3.2.1.6.1 Ağırlıklar.....	24
3.2.1.6.2 Birleştirme fonksiyonu.....	24
3.2.1.6.3 Transfer fonksiyonu.....	25
3.2.1.7 Yapay sinir ağının yapısı.....	27

3.2.1.8 Yapay sinir ağlarında öğrenme ve eğitme.....	28
3.2.1.8.1 Yapay sinir ağlarının eğitilmesi.....	29
3.2.1.8.1.1 Öğretici eğitme.....	29
3.2.1.8.1.2 Skor ile eğitme.....	30
3.2.1.8.1.3 Kendini düzenleme ile eğitme.....	30
3.2.1.9 Geri yayılım algoritması.....	30
3.2.1.9.1 Öğrenme oranı, momenttim katsayısı ve bias.....	35
3.2.1.10 Levenberg-Marguardt metodu (LMM).....	35
3.2.2 Sistem tanımlanması.....	37
3.2.3 Tahmine dayalı kontrol.....	42
3.2.3.1 YSA kullanarak nonlinear sistemin modelinmesi.....	42
3.2.3.2 YSA bazlı tahmin.....	42
3.2.4 YSA geri beslemeli lineerizasyon kontrol.....	44
3.2.5 Model referanslı kontrol.....	47
3.2.6 Karıştırmalı biyoreaktörde oksijen kütle aktarım katsayısının bulunması	48
4. BULGULAR.....	50
4.1 Yapay Sinir Ağı ile Reaksiyon Hızı Tahmini.....	50
4.1.1 Sıcaklığın mikroorganizma çoğalmasına etkisi.....	50
4.1.2 Farklı sıcaklıklarda yapay sinir ağları ile reaksiyon hızı tahmini.....	51
4.1.3 pH'ın mikroorganizma çoğalmasına etkisi.....	56
4.1.4 Farklı pH değerlerinde yapay sinir ağları ile reaksiyon hızı tahmini.....	57
4.2 Saccharomyces Cerevisiae Mikroorganizmasının YSA İle Kontrolü.....	62
4.2.1 Tahmine dayalı kontrol.....	69
4.2.2 NARMA-L2(Geri beslemeli linerizasyon)kontrol.....	77
4.2.3 Model referanslı kontrol.....	89
4.3 Kütle Aktarım Katsayısının Yapay Sinir Ağları Tarafından Tahmini.....	103
5. TARTIŞMA VE SONUÇ.....	106
KAYNAKLAR.....	110
EKLER.....	113
EK 1 Governin Eşitliğinin Similasyonu.....	114
EK 2 Kullanılan Similasyonda ki Bloklarının Görevleri.....	115

EK 3 ARMAX'tan Elde Edilen Modelin Similasyonu.....	117
EK 4 ARMAX Katsayılarının Bulunmasında Kullanılan Matlab Programı	118
EK 5 Kütle Aktarımı Katsayısının Hesaplanması için Oluşturulan Matlab Programı.....	121
EK 6 Farklı Sıcaklıklarda Yapay Sinir Ağları ile Reaksiyon Hızı Tahmininde Kullanılan Matlab Programı.....	122
EK 7 Farklı pH Değerlerinde Yapay Sinir Ağları İle Reaksiyon Hızı Tahmini....	123
EK 8 Farklı sıcaklıklarda yapay sinir ağları ile reaksiyon hızı tahmininde kullanılan matlab programında kullanılan hizgiris.xls, hizcikis.xls verileri.....	124
EK 9 Farklı Ph Değerlerinde Yapay Sinir Ağları İle Reaksiyon Hızı Tahmini Matlab Programında Kullanılan Hiz2giris.Xls Ve Hiz2giris.Xls Verileri.....	126
EK 10 Kütle Aktarımı Katsayısının Hesaplanması İçin Oluşturulan Matlab Programındaki Giris.Xls ve Cikis.Xls Verileri.....	127
EK 11 Kütle Aktarımı, Farklı Sıcaklık ve pH Değerinde Kullanılan Test Verileri.....	128
ÖZGEÇMİŞ.....	129

SİMGELER DİZİNİ

$\pm a$	Genlik seviyesi
C_x	Biyokütle derişimi (g l^{-1} kuru ağırlık)
C_i	Derişim
c_0	Gaz faz çözünmüş oksijen mol kesri
DO	Sıvı faz oksijen konsantrasyonu (g/l)
DO_{set}	Set noktası için çözünmüş oksijen (g/l)
$e(t)$	Beyaz gürültü
e	Hata
$G(q)$	Transfer fonksiyonu
H	Henry sabiti($\text{gO}_2/\text{L H}_2\text{O}$)
$H(q)$	Bozucu etki
J_k	jakobiyen matrisi
I	Birim matris
k_{La}	Oksijen kütle aktarım katsayısı (s^{-1})
m	Ayar değişkeni sayısı
M	Molekül ağırlığı
n	Sistem çıktısı sayısı
n_a, n_b, n_c	Polinom tipi modelin derecesini tanımlayan parametre
n_k	Gecikme
N_u	Kontrol ufku
N_1, N_2	minimum ve maksimum tahmin ufku
OUR	Oksijen tüketim hızı ($\text{g O}_2/(\text{l}^*\text{s})$)
P_a	Güç tüketimi (W)
$\frac{P_q}{V}$	Güç katsayısı (W^*m^{-1})
p	Basınç (atm)
r	Reaksiyon hızı
R	Gaz sabiti
Q	Hacimsel hava akış hızı(l/dk)
s	Substrat derişimi (g^*l^{-1})

t	Şimdiki zaman
T	Sıcaklık
$u(t)$	Model girdisi
X_t	Şimdiki değer
v_s	Hava akış hızı ($m*s^{-1}$)
V	Reaktör hacmi
$\hat{y}$	YSA model çıktısı
$y(t)$	Model çıktısı
y_i	Gerçek değer
y_{id}	Deneysel değer
y_{in}	Giren hava oksijen kesri
Y_{o_2}	Oksijen verimi (g biyokütle/g oksijen)
Y_{sx}	Substrat verimi (g biyokütle/g oksijen)
x	Biyokütle konsantrasyonu (g/l)

Yunan Alfabesi

η	Eğitme katsayısı
α	Momentum katsayısı
λ	Marquardt parametresi
σ	Aktivasyon fonksiyonu
ε	Boşluk kesri
μ	Spesifik büyüme hızı (1/s)
ξ	Ağırlık faktörü

Kısaltmalar

PRBS	Sahte rasgele ardışık ikili sinyal
YSA	Yapay sinir ağı
<i>S.cerevisiae</i>	<i>Saccharomyces cerevisiae</i>
TDL	Sistem çıktısının ve hava akış hızının geçmiş ve şu anki değeri

ŞEKİLLER DİZİNİ

Şekil 3.1 Deney düzeneği.....	13
Şekil 3.2 Tipik olarak sinir ağının bağlantı şekli.....	19
Şekil 3.3 Üç katmanlı yapay sinir ağı.....	20
Şekil 3.4 Biyolojik sinir ağının blok gösterimi.....	21
Şekil 3.5 Bir nöronun şematik yapısı.....	22
Şekil 3.6 Bir nöron yapısı	23
Şekil 3.7 Eşik fonksiyonları.....	26
Şekil 3.8 Çok katmanlı nöral ağ.....	27
Şekil 3.9 Çok katmanlı bir sinir ağı ve katmanları.....	28
Şekil 3.10 Geriye yayılım ağ mimarisi.....	31
Şekil 3.11 Geriye yayılım algoritması.....	34
Şekil 3.12 PRBS sinyali.....	37
Şekil 3.13 Model referanslı kontrol şeması.....	47
Şekil 4.1 Farklı sıcaklıklarda mikroorganizma değişiminin değişimi.....	51
Şekil 4.2 YSA şematik gösterimi.....	53
Şekil 4.3 10-12-1 nörona sahip YSA için eğitimdeki hata.....	53
Şekil 4.4 Deneysel hıza karşı YSA tarafından hesaplanan hız verileri.....	54
Şekil 4.5 32 °C’de deneysel değişimle YSA tarafından hesaplanan hızdan elde edilen Değişim.....	56
Şekil 4.6 Çoğalma ortamı pH değerinin mikroorganizmanın çoğalma hızına etkisi.....	57
Şekil 4.7 YSA şematik gösterimi.....	58
Şekil 4.8 Eğitilen 12-13-1 nörona sahip YSA için eğitimdeki hata.....	59
Şekil 4.9 Deneysel hıza karşı YSA tarafından tahmin edilen hız.....	60
Şekil 4.10 pH=5 için deneysel değişim ile YSA tarafından bulunan hızdan elde edilen Değişim.....	61
Şekil 4.11 Deneysel ve teorik oksijen değişiminin karşılaştırılması.....	64
Şekil 4.12 Deneysel oksijen değişimine karşı teorik oksijen değişimi	64
Şekil 4.13 Sistem çıktısının PRBS etkisiyle değişimi.....	65
Şekil 4.14 PRBS etki.....	65
Şekil 4.15 Sistem ile ARMAX çıktısının karşılaştırılması.....	65

Şekil 4.16 ARMAX sıfırları ve kutupları.....	67
Şekil 4.17 Oto korelasyon ve kros korelasyon grafikleri.....	68
Şekil 4.18 ARMAX model artıkları.....	68
Şekil 4.19 Tahmine dayalı kontrol blok diyagramı.....	70
Şekil 4.20 Sistem girdisi ve sistemin bu girdilere olan tepkisi.....	70
Şekil 4.21 Yüz deneme sonucunda YSA hata değeri.....	71
Şekil 4.22 Oksijen akış hızının, Taşınım olaylarına bağlı diferansiyel (governing) eşitliklerinden elde edilen çıktının, YSA çıktısının ve hata değeri.....	72
Şekil 4.23 Sistemin oksijen kontrolü.....	73
Şekil 4.24 ARMAX modelinin giriş ve çıkış değerleri.....	74
Şekil 4.25 Oksijen akış hızının, governing eşitliklerinden elde edilen çıktının, YSA çıktısının ve hata değeri.....	75
Şekil 4.26 Yüz Epochs için hata değeri	76
Şekil 4.27 ARMAX modelinin oksijen kontrolü.....	77
Şekil 4.28 Tahmin için kullanılan modelin yapısı.....	78
Şekil 4.29 Kontrol edicinin ağ yapısı.....	79
Şekil 4.30 Tüm sistemin blok diyagramı.....	80
Şekil 4.31 Sistemin blok diyagramının matlab toolbox kullanımı.....	80
Şekil 4.32 Governing eşitliklerinin giriş ve çıkış verileri.....	81
Şekil 4.33 Yüz Epochs için hata değeri.....	82
Şekil 4.34 Sistem girdisi,çıkışı,YSA çıktısı ve hata değeri.....	83
Şekil 4.35 NARMA-L2 yöntemiyle oksijen kontrolü.....	84
Şekil 4.36 Model girdisi ve çıktısı.....	85
Şekil 4.37 Eğitimde ki hata değeri.....	86
Şekil 4.38 Model çıktısı, YSA çıktısı ve bunlar arasındaki hata değerleri.....	87
Şekil 4.39 NARMA kontrol edici ile ARMAX'tan elde edilen modelin kontrolü.....	88
Şekil 4.40 Model referanslı kontrol edici yapısı.....	89
Şekil 4.41 Matlab toolbox model referanslı kontrol edici.....	89
Şekil 4.42 Model çıktısı ve girdisi.....	90
Şekil 4.43 Sistem çıktısı ile model çıktısı arasındaki eğitimde hata değeri.....	91
Şekil 4.44 Model girdi ve çıktısı, YSA çıktısı ve YSA çıktısı arasındaki hata.....	92
Şekil 4.45 Kontrol edici girdi ve çıktısı.....	93

Şekil 4.46 Sistem çıktısı ile referans model arasındaki hata.....	94
Şekil 4.47 Referans model ve model çıktısı.....	95
Şekil 4.48 Governing eşitliğinden elde edilen modelin oksijen kontrolü.....	96
Şekil 4.49 Model girdisi ve çıktısı.....	97
Şekil 4.50 Eğitimdeki hata değeri.....	98
Şekil 4.51 Model çıktısı YSA çıktısı ve model çıktısı ile arasındaki hata.....	99
Şekil 4.52 Referans model girdi ve çıktısı.....	100
Şekil 4.53 Referans model ile sistem çıktısı arasındaki YSA hata değerleri.....	100
Şekil 4.54 Referans model girdi değeri, YSA ve referans model çıktısı.....	101
Şekil 4.55 Çalışmada kullanılan ARMAX modelinin oksijen kontrolü.....	102
Şekil 4.56 Kütle aktarımın katsayısının tahmininde kullanılan YSA şematik gösterimi.....	104
Şekil 4.57 Lewenberg Marquart algoritması ile eğitilen ağın Hatası.....	105
Şekil 4.58 Geri besleme algoritması ile eğitilen ağın hatası.....	105

ÇİZELGELER DİZİNİ

Çizelge 3.1 Katı besi ortamı bileşimi.....	12
Çizelge 3.2 En çok kullanılan birleşme fonksiyonları.....	24
Çizelge 3.3 En çok kullanılan transfer fonksiyonları.....	25
Çizelge 4.1 Deneysel hız ile YSA tarafından tahmin edilen hızın karşılaştırılması.....	54
Çizelge 4.2 Deneysel hız ile YSA tarafından tahmin edilen hızın karşılaştırılması.....	59
Çizelge 4.3 Denklemden elde edilen $k_L a$ ile YSA tarafından elde edilen $k_L a$ 'nın karşılaştırılması.....	104

1.GİRİŞ

Yapay sinir ağı (YSA) , insanın beynindeki birçok nöronun, ya da basit işlemcilerin birbirlerine değişik etki seviyeleriyle paralel olarak bağlanması ile oluşan bir bilgi işleme sistemi olarak tanımlanır. Beynin fizyolojik yapısı olan düşünme, hatırlama ve problem çözme yeteneklerini bilgisayara aktarma mantığına dayalı teknolojidir. İnsan öğrenme metotlarından birisi olan örneklerle öğrenme metodunun makinelere aktarılması esasına dayanmaktadır. Yapay sinir ağı kullanılarak beynin fonksiyonlarını yerine getiren model oluşturulmaya çalışılmıştır. Yapılan bütün çalışmalara rağmen yapay sinir ağı modelleri, biyolojik ağların karmaşıklığını yansıtmaktan henüz çok uzaktır. Ancak tam olarak beynin çalışmasını yansıtan bir çalışma bulunmasada, günlük hayatı kolaylaştıracak birçok YSA uygulaması geliştirilmiştir.

Önceleri temel tıp bilimlerinde insan beynindeki nöronların matematiksel modellenmesi çabalarıyla başlayan çalışmalar son yıllarda oldukça disipline bir hal almıştır. YSA, bugün fizik, matematik, elektrik ve bilgisayar mühendisliği vs. gibi çok farklı bilim dallarında araştırma konusu halin gelmiş ve çeşitli meslek gruplarında çalışanlar, YSA'nı kendi uzmanlık alanlarına tanıtmışlardır. Böylece insan zekâsına has görünen bazı kavramların sayısal olarak ifade edilebileceği ve böylece makinelerin insan zekâsına şaşırtıcı derecede benzer yollarda öğrenme ve hatırlama işlemlerini yapabileceği görülmüştür.

Herhangi bir problemin çözümünü sağlayan bir algoritma geliştirilmiş ise geleneksel bilgisayar sistemleri, problemi çözmek için yeterli olmaktadır. Önemli olan problemin çözümünü veren bir formül olmadığı durumda bilgisayara problemi çözdürmektir. Bu zekâ gerektiren bir fonksiyondur. Bu gibi durumlarda YSA örneklerle eğitilerek model önermeden problemi çözmektedir. Yapay sinir ağının geniş kapsamlı olarak kullanılmasının en önemli nedeni, klasik tekniklerle çözümü zor olan problemlerin çözümünde etkin rol oynamasıdır. Yapay sinir ağlarının, karmaşık eşleştirmeleri hassas bir biçimde gerçekleştirmesinden ve yapısal dinamiğe sahip olmasından dolayı gün geçtikçe uygulama alanı genişlemektedir. Eğitilmiş bir YSA analiz edilecek bilgiler için bir uzman olarak düşünebilir. Bu uzman, yeni durumlar için tahminler yapabilir.

Ağın gizli katman sayısı ve bir katmanda bulunan nöron sayısı ne kadar çok ise ağ o kadar iyi öğrenir gibi bir kural yoktur. Gizli katman sayısı ve gizli nöron sayısı arttıkça işlem sayısı artmakta bu şekilde ağın öğrenme süresi uzamaktadır yani ağ tasarlanırken mümkün olan en küçük ağ yapısı seçilmelidir. Gizli katman sayısı fazla seçilir ise ağda ezberleme olasılığı artmaktadır. Ağ yapısı deneme yanılma ile en ideal yapısı bulunur bazı çalışmalarda önerilen ağ yapıları olmasına rağmen henüz kesin bir yöntem bulunamamıştır. Ağda kullanılan en ideal birleşme fonksiyonu da deneme yanılma ile bulunmaktadır ama genelde toplam birleşme fonksiyonu kullanılmaktadır. Aynı şekilde kullanılan transfer fonksiyonu da deneme yanılma ile bulunur henüz hangi durumda hangi transfer fonksiyonun kullanacağını belirleyen kesin bir yöntem yoktur. Transfer fonksiyonu ağın fonksiyon haline dönüştürülmesi için kullanılır. Yapay sinir ağlarının avantajlarını özetleyecek olursak: Matematiksel bir model eşitliğine gerek duyulmamaktadır, kompleks mühendislik problemleri için sayısal çözümlerin bazı aşamalarında kullanılabilir, YSA kendine ait avantajları ile çok geniş kullanım alanı bulmuştur, klasik yöntemlerde eşitlik çıkarılması için belirli varsayım yapılmıyorken YSA örneklerle öğrendiği için varsayımlardan doğan hataları içermez, yüksek güvenilirlikte sonuçlar üretmesiyle geleneksel yöntemlere bir alternatif olarak kullanılabilir.

Biyokimyasal tepkimeleri katalizleyen mikroorganizmalar, biyokimyasal dönüşümleri gerçekleştirmek üzere hem araştırma hem de ticari uygulamalarda biyokatalizör kaynağı olarak görev alırlar. Biyoteknolojik proseslerde mikroorganizmanın kendisi ya da yapısından izole edilen bir enzim, biyokimyasal dönüşümü gerçekleştirmek üzere kullanılır. Biyokimyasal ürünler; bira ve şarap gibi içecekler, antibiyotikler, çözücüler ve organik asitler gibi endüstriyel kimyasallar olabilmektedir. Bu çalışmada *Saccharomyces cerevisiae* mikroorganizması çoğaltılmıştır. Bu mikroorganizma bir mayadır. Biyoteknolojik proseslerde biyokatalizör kaynağı olarak mayalarının kullanılmasının avantajı, ağırlıklarının iki katına ulaşabilmesi için geçen sürenin 10-120 dakika gibi bir kısa zamanda gerçekleşmesidir. Diğer mikroorganizmalar için bu süre günler veya haftalar olabilmektedir.

Mayalar doğada yaygın olarak bulunurlar. Tek hücrelidirler. Genellikleri çapları 4-10 μ arasında değişen yuvarlak, oval, silindirik şekillerde bulunurlar. Mayalar genellikle

tomurcuklanma, sporlanma ve bölünme ile çoğalır. *Saccharomyces cerevisiae* mayası ise sporlanarak çoğalmaktadır. Mayanın gelişebilmesi ve çoğalabilmesi için suya, karbonhidratlı, azotlu ve madensel maddelere gereksinimi vardır. Birleşiminde ise genellikle su, azotlu maddeler, karbonhidratlar, anorganik tuzlar, yağ enzimler ve vitaminler bulunmaktadır.

Her canlı gibi mayanın büyük bir kısmını su teşkil eder. Bu da hücre zarının esnekliğini ve seçici geçirgenliğini sağlarken, serbest halde bulunan su, maddenin çözünür halde kalmasını sağlamaktadır. Mayalar bilindiği gibi aerobik ve anaerobik koşullarda yaşayabilmektedir. Solunum yani oksidatif metabolizma sırasında maya kendi yapı maddelerini oluşturarak gelişir ve ürer. Anaerobik koşullarda ise etil alkol oluşumu gerçekleşir. Mayanın gelişme koşulları içerisinde suyun, oksijenin, ışığın, besi ortamı ozmatik basıncının ve pH'ın etkisi büyüktür. *Saccharomyces cerevisiae* mikroorganizmasının çözünmüş oksijen derişimi için herhangi bir kontrol işlemi uygulanmadığında ürün seçimliliğini etkilemektedir. Çözünmüş oksijen derişimi belirli bir derişimin altına düştüğünde maya etil alkol üretmektedir. Sisteme fazla hava göndermek çözülen oksijen miktarının fazla olmasını sağlar ama sisteme fazla hava göndermenin bazı sakıncaları vardır bu sakıncalar ise pompalama maliyetini artırması ve sistemde kabarcıklaşmaya neden olmaktadır. Sistemde ki kabarcıklaşma ölçüm elemanların hassas ölçüm almasını engellemektedir. Bu yüzden sistemde oksijen kontrolüne ihtiyaç vardır. Prosesin O_2 ihtiyacı önemli bir etken olarak prosesin gidişini etkilemektedir. O_2 derişimin zamanla değişimi, hava ile *Saccharomyces cerevisiae* mayası arasında kütle aktarım katsayısına bağlıdır. *Saccharomyces cerevisiae* mayası üretmek için yapılması gereken reaktörün boyutlarının belirlenmesi için teorik tasarım hesaplamalarına gerek vardır. Bu hesaplamalar için O_2 tüketim hızının, kütle aktarım katsayısının ve maya üreme hızının bilinmesi gerekmektedir. Bu çalışma üç aşamadan oluşmaktadır. Bunlar: Mikroorganizma reaksiyon hızlarının tahmini, kütle aktarım katsayılarının YSA ile tahmini, çözünmüş oksijen derişimini YSA ile kontrolüdür. Deneysel veriler ve bağıntılar literatürden alınmıştır.

Model denklemleri kullanılmadan reaksiyon hızlarının tahmini YSA ile yapılmıştır. Kimya mühendisliğinde prosesler birçok parametrelere ve değişkenlere bağlı olup

aralarında karmaşık bir ilişki vardır. Bunlardan herhangi birinde belirli bir oranda değişim diğerine orantılı olmayan bir değişime yol açabilir. Bu tip prosesler doğrusal olmayan prosesler olarak değerlendirilip benzetiminde ve kontrolünde doğrusal olmayan modellerini kullanılması gerekmektedir.

Bu çalışmada YSA kullanılarak hız verileri modellemeye başvurulmadan, deneyselle yakın olarak başarı bir şekilde tahmin edilmiştir. Yapılan araştırmada mikroorganizmanın üretiminde sıcaklık, pH, başlangıç derişiminin etkili olduğu gözlenmiştir. Bu yüzden ağın giriş değişkenleri olarak sıcaklığın mikroorganizma çoğalmasındaki etkisinde ki hız hesaplanmasında: mikroorganizma başlangıç derişimi, mikroorganizma derişimi ve sıcaklık seçilmiştir. Değişik pH'lardaki ağın giriş değişkenleri ise pH, mikroorganizma derişimi ve mikroorganizma başlangıç derişimi seçilmiştir. Mayanın çoğalması beş ayrı sıcaklıkta gerçekleştirilmiştir (27, 30, 32, 35, 38 °C) bu sıcaklık değerleri kullanılarak reaksiyon hızları değişik sıcaklarda tahmin edilmeye çalışılmıştır, kullanılan bu sıcaklıklarda pH değeri sabit olup pH değeri 5'tir. Farklı pH değerlerinde YSA ile reaksiyon hızları tahmininde ise pH değerleri 3, 4, 5, 6, 7 değerleri seçilmiş ve farklı pH değerleri için reaksiyon hızı YSA kullanılarak hesaplanmıştır. Bu pH değerlerinde alınan deneysel veriler sabit sıcaklıkta (32°C) alınmıştır. Hız verilerinin tahmininde kullanılan ağ yapısı ise farklı sıcaklıkta YSA ile hız tahmininde 10-12-1 ağ yapısı kullanılmış olup farklı pH değerinde YSA ile reaksiyon hızları tahmininde ise ağ yapısı 12-13-1'dir.

Sistemin matematiksel modeli iki şekilde yapılmıştır, bunlar: polinom tipi (ARMAX) model ve taşınım olayları ile ilgili diferansiyel eşitliklerdir. Taşınım olayları ile ilgili diferansiyel denklemin doğruluğunu kontrol etmek için biyoreaktörün dinamik özellikleri deneysel ve teorik olarak incelenmiş teorik çalışma ile deneysel çalışmanın birbirine uygunluğu araştırılmıştır. Verilen işletim şartlarında teorik ve deneysel olarak bulunan oksijen derişimi karşılaştırılmıştır. Teorik ve deneysel oksijen derişimi profilinin zamana göre değişimi yaklaşık 4 saat boyunca yakın olduğu görülmüştür. Gecikme fazı ve Üstsel üreme faz için denklemlerin kullanılabileceği sonucuna varılmıştır. Teorik ve deneysel oksijen derişiminin azda olsa farklı olmasının sebebi

modelin geliştirilmesindeki varsayımlar ve deneysel hatalardan ileri geldiđi kanısına varılmıřtır.

Sistemin oksijen deriřimi kontrolü iin  ayrı kontrol yntemi kullanılmıř olup, her kontrol yntemi hem ARMAX ile modellenen; hem tařınım olayları ile ilgili diferansiyel eřitliklerle modellenen denklemlere uygulanmıřtır. Uygulanan kontrol yntemleri řunlardır: Tahmine dayalı kontrol (model ngrmeli kontrol), NARMA-L2 (geri beslemeli linerizasyon) kontrol, model referanslı kontrol. Tahmine dayalı kontrol ve NARMA-L2 kontrol ynteminde hem ARMAX hem tařınım olaylarına dayalı olarak ıkarılan diferansiyel modelle kontrol bařarı ile sađlanmıřtır.

Ktle aktarım katsayısının hesaplanmasında kullanılan ađ yapısı 6-7-7-1'dir. Ađ girdisi olarak hava akıř hızı, biyoktle deriřimi, g katsayısı; ıkıř deđiřkeni ise ktle aktarım katsayısıdır (k_{La}). Yapay sinir ađı iki ayrı algoritma ile eđitilerek (Lewenberg Marquard algoritması, geri beslemeli algoritmasıyla) her bir algoritmadan elde edilen tahmin verileri ve deneysel veriler karřılařtırılmıřtır. Levenberg Marquardt algoritması geri beslemeye gre daha hızlı eđitilmiř ve eđitilen ađın hata deđeri daha kk bulunmuřtur.

2. KAYNAK ÖZETLERİ

Son yıllarda yapay sinir ağları, çözümü güç ve karmaşık olan yada ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmaktadır ve genellikle başarılı sonuçlar alınmaktadır. Bu kısımda yapay sinir ağlarının genel uygulama alanlarından bahsedilecektir.

Van can *et al.* (1998), doğrusal olmayan kimyasal prosesi, geri beslemeli lineerizasyon yöntemi (NARMA-L2) kullanarak YSA ile basıncı kontrol etmişlerdir. Deneysel verilerden faydalanarak sistem modellenmiş ve bu modeli kullanarak basınç kontrolü sağlamışlardır. Ayar değişkeni olarak sisteme giren hava akış hızı seçilmiştir.

Kasparian *et al* (1998), Yapay sinir ağları kullanılarak model referans bazlı kontrol ile doğrusal ve doğrusal olmayan bir sistemi modelleyerek bu sistemi kontrol etmeye çalışmışlardır. Ağ eğitiminde iki ayrı algoritma kullanılmış ve bu algoritmaların bir birlerine karşı olan üstünlükleri tartışılmıştır.

Hang *et al.* (1998), Yapay sinir ağını kullanarak GKT reaktörünün geri beslemeli lineerizasyon yöntemiyle derişim kontrolünü sağlamaya çalışmışlardır. Öncelikle sistem modellenmiş (diferansiyel denklem olarak) elde edilen model kullanılarak kontrolü başarılı şekilde sağlanmıştır. Ayar değişkeni olarak soğutucu akış hızı seçilmiştir.

Molga and Cherbanski (1999), sıvı-sıvı reaksiyon sisteminde YSA kullanarak reaksiyon hızını modellemeye çalışmıştır. Reaksiyon olarak propionik anhidrit katalizi seçilmiştir. YSA değişik koşullarda kesikli ve yarı kesikli reaktöre uygulanmıştır. Eğitim için hız verileri deneysel olarak saptanmıştır. Giriş değişkeni olarak reaktör sıcaklığı, propionic anhidritin mol kesri, propionik asidin mol kesri, sülfürik asidin mol kesri ve karıştırma hızı alınmıştır. Çıkış değişkeni olarak hız alınmıştır. Tahmin edilen hız, kütle ve enerji denkleminde kullanılıp elde edilen kütle ve sıcaklık bilgileri deneysel sonuçlar ile karşılaştırılmıştır. Bu verilerden YSA kabul edilebilir deneysel sonuç ürettiği görülmüştür.

Horn (2001), yaptığı çalışmada polimerizasyon reaktöründe YSA kullanılarak geri beslemeli lineerizasyon kontrol yöntemi ile sıcaklık kontrolü sağlamaya çalışmıştır. Sistem modellenmiş elde edilen modelden yararlanılarak sıcaklık kontrolü başarı ile sağlanmıştır. Sisteme beslenen monomer derişim ayar değışkeni olarak seçilmiştir.

Garcia and Gomez (2001), sürekli karıştırılmalı bir polisakkarit reaktörde oksijen kütle aktarım katsayısını YSA kullanarak tahmin etmeye çalışmışlardır. Kütle aktarım verileri deneysel olarak elde edilmiştir. Ağın giriş verileri olarak karıştırma hızı, gaz akış hızı, sıcaklık, akışkan vizkozitesi, güç katsayısı, reaktör çapı önerilmiş bunlardan hangisinin kütle aktarım katsayısında daha etkili olduğu tartışılmış ve doğrusal olmayan denklem önererek kütle aktarım katsayısını hesaplamışlar ve YSA'dan bulunan kütle aktarım katsayısı ile karşılaştırma yapmıştır. Yaya sinir ağından bulunan kütle aktarım katsayısının deneyselle daha yakın olduğu görülmüştür, ağının aktivasyon fonksiyonu olarak sigmoid seçilmiştir.

Yu and Gomm (2003), çok değışkenli bir kimyasal reaktörün kontrolüne yapay sinir ağını uygulamışlardır. Kontrol edilen değışkenler sıcaklık, pH, çözünmüş oksijen derişimidir. Ayar değışkenleri ısı, asit baz akış hızı, oksijen akış hızı dır. Kontrolün sağlanması için bir amaç fonksiyonu önerilmiş ve bu amaç fonksiyonunu sıfır yapan ayar değışkenleri amaç fonksiyonunun ayar değışkenlerine göre türevi alınıp sıfıra eşitlenmesi ile bulunmuşlardır.

Terzi (2006), eğridir gölünün su sıcaklığını YSA ile tahmin etmeye çalışmıştır. Ağın giriş değışkenleri olarak günlük hava sıcaklığı, güneş ışıınımı, nisbi nem parametreleri kullanılarak bir(hava sıc.,güneş ışıınımı, nisbi nem), iki(hava sıc.,nisbi nem; hava sıc.,güneş ışıınımı; güneş ışıınımı, nisbi nem) ve üç girdili (hava sıc., güneş ışıınımı, nisbi nem) yedi farklı YSA modeli geliştirilmiştir. Geri yayınımlı YSA öğrenme metodu kullanılmış. Eğitim verileri 2000-2001-2002 yıllarına ait günlük değerler test için 2003 yılına ait değerler alınmıştır. Eğitim İçin 613 veri, test için 261 veri kullanılmış. Uygun ağ yapısı deneme yanılmayla bulunmuştur. Bu çalışmanın amacı eğridir gölünün su sıcaklığının YSA ile belirlenmesi için model türetmek ve bu modelin doğrulunun test edilmesidir.

Fırat vd. (2006), Yapay sinir ağıları kullanılarak tuğla duvardaki ve tesisat borusundaki ısı kaybı tespit edilmeye çalışılmıştır. Ağ eğitim algoritması olarak geri yayınlımlı YSA tercih edilmiştir ve veriler normalize edilerek ağa sunulmuştur. Ağdan elde edilen çıkışlar deneysel sonuçlarla karşılaştırılmıştır ve yeterli hassasiyette olduğu görülmüştür. Yalıtımlı duvar için ağın giriş verileri iç sıva kalınlığı, dış sıva kalınlığı, tuğla kalınlığı yalıtım malzemesi kalınlığı, iç ortam sıcaklığı, dış ortam sıcaklığı. Ağın çıkış verisi olarak duvar iç yüzey sıcaklığı alınmıştır. Yalıtımsız boru için ise boru çapı, boru iç sıcaklığı, dış hava sıcaklığı ağa giriş olarak uygulanmış çıkış parametresi ise boru ısı kaybıdır.

Özkan vd. (2006), Yaptığı çalışmada, kızıllırmak nehrinin çözünmüş oksijen konsantrasyonunun değişimi modellenmiştir. dokuz ay süre ile on ayrı noktadan su kalite değerleri alınmıştır. Giriş değişkeni olarak sıcaklık, pH, toplam çözünmüş katı madde miktarı, ay(süre), ölçüm noktası alınmış. Çıkış değişkeni olarak ise çözünmüş oksijen konsantrasyonu alınmıştır. 90 adet verinin 80 tanesi eğitim 10 tanesi test için kullanılmıştır. Sigmoid aktivasyon fonksiyonu kullanılmış, en uygun ağ yapısı deneme yanılma ile bulunmuştur.

Bu çalışmada YSA'nın klasik modellemeye göre üstünlükleri şu şekilde sıralanmıştır; Hızlı olması, Basit yapıda olmaları, Modellenmesi zor problemlere kabul edilebilir sonuçlar sunmasıdır.

Mjalli *et al.* (2006), Etanol'ün fermantasyon prosesin kontrolünü, hem model bazlı tahmine dayalı kontrol algoritmasının YSA ile birleştirerek sağlamış hem de NARMA-L2 yöntemi ile sağlamıştır. Eğitim ve test için veriler kütle ve enerji denkliklerinden elde edilmiştir. Ayar değişkeni seyrelme hızı kontrol değişkeni etanol derişimidir. NARMA-L2 yönteminin daha hızlı kontrolü sağladığını görmüşlerdir.

Boyacı vd. (2006), Yapay sinir ağlarını kullanarak reaksiyon hızları tahmin etmeye çalışmışlardır. Reaksiyon hızları tahmini için ağa giriş değişkeni olarak maltoz, glukoz derişimi ve glukoz başlangıç derişimi uygulanmış; ağ çıktısı ise reaksiyon hızıdır. Ayrıca sistem için modeller önerilmiş, deneysel reaksiyon hızlarını kullanarak önerilen

modellerin reaksiyon sabitlerini bulmaya çalışmışlardır. Bulunan bu reaksiyon sabitleri kullanılarak önerilen modellerden reaksiyon hızları hesaplanmış ve hesaplanan reaksiyon hızı ile YSA tarafından tahmin edilen reaksiyon hızı karşılaştırılmış ve YSA deneysel hıza daha yakın değerde olduğu görülmüştür.

Boyacının (2006) yaptığı çalışmasından da anlaşılacağı gibi yapay sinir ağları klasik yöntemlerle bulunan verilere göre daha yüksek doğrulukta sonuçlar bulmuştur yani deneyselle daha yakın sonuçlar bulmuştur. Sonuçların yüksek güvenilirlikte olması sebebiyle geleneksel yöntemlere bir alternatif olarak kullanılabilir. Geleneksel tahmin yöntemlerde (örneğin model önererek) de deneysel verilerle çalışma imkânı mümkündür ama yapay sinir ağı, ağı tasarlayan kişinin bilgi ve tecrübesine göre deneyselle daha yakın sonuçlar üretmektedir.

Yapılan çalışmalarda Ağın eğitimi sırasında kullanılan örnek sayısı ağı tasarlayan kişi tarafından seçilmiş olup örnek sayısı ne kadar fazla ise öğrenme o derece gerçekleşmektedir. Ağın ilk ağırlıkları kullanıcı tarafından rasgele verilmekte bu ağırlık güncellenerek ağ eğitilmektedir Ağın eğitilmesi için kullanılan algoritma kullanıcı tarafından seçilmektedir, genellikle geri yayılım algoritması ve Levenberg Marquardt algoritması kullanılmaktadır. Levenberg Marquardt algortmasının geri yayılım algoritmasına göre üstünlükleri vardır bunlar: ileri beslemeli ağlarda en hızlı öğrenme algoritması olması ve yavaş yakınsama problemlerinden etkilenmemesidir. Bir katmandaki nöronlar genellikle aynı birleşme ve aktivasyon fonksiyonuna sahiptir. Bir işlem elemanı (nöron) hiçbir zaman tek başına çalışmaz nöronlar çok sayıda bağlantı içeren ağ içerisinde bulunur. Tasarlanan ağın gizli katman sayısı da kullanıcı tarafından tasarlanmaktadır. Çalışmalarda da uygulanacak gizli katman sayısı, bir katmandaki nöron sayısı ve kullanılan aktivasyon sayısı da deneme yanılma ile bulunmaktadır.

Az sayıda gizli katman seçildiğinde ağın genelleme yeteneğinin arttığı görülmüştür. Çok sayısı gizli nöron ayın genelleme yeteneğinden çok ağın ezberleme yeteneği kazanmasına neden olmuştur ve çok sayıda gizli katman veya nöron hesap sayısının artmasına neden olmaktadır.

Yapay sinir ağıları hızlı olmaları, basit bir yapıya sahip olmaları modellenmesi zor olan problemlere kabul edilebilir sonuçlar üretmesi sebebiyle, birçok dalda rahatlıkla kullanılabilecek bir yöntemdir. Doğruluğu kontrol edilen bir ağ defalarca kullanılabilir. Bütün bunlar göz önüne alındığında yapay sinir ağıları düşük maliyetle uygulanabilecek bir yöntemdir.

3. MATERYAL VE YÖNTEMLER

3.1 Materyal

Yapılan çalışmada tüm deneysel veriler literatürden alınmıştır (Akay 1998, Boyacıoğlu 2002). Bu deneysel verilerin elde edilmesi için gerekli materyal aşağıda verilmiştir.

3.1.1 Mikroorganizma

Çalışmada *Saccharomyces cerevisiae* NRRL Y-567 mayası (Northern Research Center, Peoria, IL, USA) kullanılmıştır. Mikroorganizmanın çoğaltıldığı katı ve sıvı besi ortamları literatürde verilen bileşimlere göre hazırlanarak 121°C ve 1.2 atm’de 20 dakika boyunca doymuş buharla sterilizasyon gerçekleştirilmiştir.

3.1.2 Katı besi ortamı

Çizelge 3.1’de bileşimi verilen katı besi ortamı karbon, azot, enerji kaynağı ile agardan oluşmaktadır. Besi ortamını oluşturan bileşenlerin, çalışılacak hacme göre tartımları alınarak saf su içinde homojen karışımı hazırlanmıştır. Otoklavda glukoz sterilizasyon sonunda karıştırılmış ve tuz çözeltileri ayrı ayrı sterilizasyon geçirilerek ağızları mikrobiyolojik filtre ile kapatılmış, tüplere belli bir seviyede doldurularak tüpler eğik şekilde soğumaya bırakılarak; eğik agarlar hazırlanmıştır. Mikroorganizma eğik agar üzerine steril kabinde alev yanında öze ile aktararak 32°C’de 20 saat süre ile inkübatörde çoğaltılmıştır.

Mikroorganizmalar on günde bir yeni agar ortamına aşılacak ve belirtilen koşullarda inkübatörde çoğaltıldıktan sonra 14°C’de saklanmıştır.

Katı besi ortamında çoğaltılan mikroorganizma ölçek büyütme ortamına uygun hacme göre hazırlanan sıvı besi ortamına aktarılmıştır.

3.1.3 Sıvı besi ortamı

Ölçek büyütme oranına uygun hacimlerde çizelge 3.1’de verilen ancak agar içermeyen sıvı besi ortamları için katı besi ortamı hazırlamada izlenen yol takip edilmiştir.

3.1.4 Ölçek büyütme

Çalışmada 1:10 oranında ölçek büyütme kullanılmıştır.

3.1.5 Deney düzeneği

Deney düzeneği 2 Lpyrex camdan üretilmiş ceketli bir biyoreaktör ile çeşitli ekipman ve sensörlerden oluşmuştur. Cekete soğutma suyunun belli sıcaklık ve akış hızında gönderen sirkülatör ile pompa, biyoteknik ortamının sıcaklık, pH ve çözünmüş oksijen derişimini ölçen sırasıyla termofit, pH metre ve DO metre yer almaktadır. Sisteme on-line bağı bilgisayar, her örnekleme zamanında ölçüm sinyalleri (sıcaklık, pH, DO) iletilerek kaydedilmektedir. Deney düzeneği ayrıca hava akış hızı kontrolü yapmak üzere gerekli donanımına sahiptir. Kompresörden sağlanan hava sırasıyla regülatör, rotametre ve mikrobiyolojik filtreden geçtikten sonra bir dağıtıcı ile karıştırıcının altından biyoreaktöre gönderilmektedir. Deney düzeneği şekil 3.1’de görüldüğü gibidir.

Çizelge 3.1 Katı besi ortamı bileşimi

Madde	Bileşim (% ağı./hacim)
Glikoz	2
Maya özütü	0.6
K ₂ HPO ₄	0.3
(NH ₄)SO ₄	0.335
NaH ₂ PO ₄	0.376
MgSO ₄ .7H ₂ O	0.052
CaCl ₂ .4H ₂ O	0.0017
Agar	2

3.2 Yöntem

3.2.1 Yapay sinir ağları

3.2.1.1 Yapay sinir ağlarının tanımı

Yapay sinir ağları yada kısaca YSA; insan sinir sisteminin biyolojik yapısının ve insan beynindeki çalışma sisteminin yapay olarak benzetimi çalışmalarının bir sonucu olarak ortaya çıkmıştır. Genel anlamada bir yapay sinir ağı insanın beynindeki birçok nöronun, ya da yapay olarak basit işlemcilerin bir birine değişik etki seviyeleriyle paralel olarak bağlanmasıyla oluşan bir bilgi işleme sistemi olarak tanımlanabilir. Kısaca, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistemdir.

Önceleri temel tıp bilimlerinde insan beynindeki nöronların matematiksel modellenmesi çabalarıyla başlayan çalışmalar son yıllarda oldukça disipline bir hal almıştır. YSA, bugün fizik, matematik, elektrik ve bilgisayar mühendisliği vs. gibi çok farklı bilim dallarında araştırma konusu haline gelmiş ve çeşitli meslek gruplarında çalışanlar, YSA'yı kendi uzmanlık alanlarına taşımışlardır. Böylece insan zekâsına has görünen bazı kavramların sayısal olarak ifade edilebileceği ve böylece marinaların insan zekâsına şaşırtıcı derecede benzer yollarla öğrenme ve hatırlama işlemlerini yapabileceği görülmüştür.

YSA'nın pratik kullanımı, genelde çok farklı yapıda ve formlarda bulunabilen verileri hızlı bir şekilde tanımlama ve algılama üzerinedir, YSA'nın mühendislik uygulamalarındaki geniş çaplı kullanımının en önemli nedeni, klasik tekniklerle çözümü zor olan problemlerin çözümünde etkin bir yol olmasıdır.

3.2.1.2 Yapay sinir ağlarının tarihi gelişimi

İnsan daima kendi beyinin yapısı ve çalışması hakkında merak sahibi olmuştur. Beynin üstün özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve insan sinir sisteminin yapısı ve fonksiyonları hakkındaki bilgi zamanla artmıştır. Birbiri ile bağlantılı olan ve sayılan milyarları bulan nöronlar, en gelişmiş bilgisayarla kıyaslandığında bile, dev bir sistemi oluşturmaktadır. Araştırmacılar, nöronların fonksiyonları ve birbiri ile olan bağlantıları daha iyi anladıkça teorilerini test etmek amacı ile matematiksel modeller geliştirme imkânı bulmuştur. Böylece sinir modellemenin, birincisi insan sinir sisteminin fizyolojik ve psikolojik işleyişini anlamak, ikincisi de beynine benzeyen fonksiyonlar gerçekleştiren yapay sinir ağları üretmek olmak üzere birbirini karşılıklı olarak destekleyen iki ana hedefi belirlenmiştir.

Bu gelişmelerle birlikte yapay sinir ağları ile ilgili çalışmalar 1943 yılında başlamıştır. Mc Culloch ve Pitts tarafından yapılan çalışmayla yapay nöronlardan oluşan bir ağın matematiksel ve mantıksal işlevleri yapabileceği gösterilmiştir. 1949 yılında Hebb, ağı oluşturan bağlantıları ayarlamak için ilk öğrenme kuralını önermiştir.

1950 ve 1960'lı yıllarda, bir grup araştırmacı bu biyolojik gelişmeleri birleştirerek ilk yapay sinir ağını üretmişlerdir. İlk olarak elektrik devreleri şeklinde üretilen ağ, daha sonra bilgisayar simülasyonu haline dönüştürülmüştür. Marvin Minsky, Frank Rosenblatt, Bernard Winrow ve daha birçok bilim adamları tek yapay nöron katmanından oluşan ağlar geliştirdiler. Perceptron adı verilen bu ağlar, hava tahmini, elektrokardiogramlar analizleri ve yapay görme gibi çeşitli alanlarda kullanıldı. Bu dönemde bilim adamları zekanın anlaşıldığı, çözüldüğü düşüncesine beynin benzerini üretmenin yeterince büyük bir ağın oluşturulmasından ibaret olduğu iyimserliğine kapıldılar .

1969'larda yapılan çalışmalar, ağların o güne kadar çözebildiği problemlere genel hatları ile benzeyen farklı problemleri çözümleyememiştir. Bu sebeple bilinmeyen başarısızlıklar sonunda Marvin Minsky, matematiksel teknikleri kullanarak, ağ işlemleriyle ilgili sağlam teoriler geliştirmiş ve Perceptrons adlı kitabıyla , Minsky ve

Papert o zaman kullanımda olan tek katmanlı ağların teorik olarak oldukça basit problemleri çözemeyeceğini ispatlamışlardır. Sonuçta Minsky yapay sinir ağlarının başarılı olacağı fikrine karşıydı. Bu sonuçlar karşısında yapay sinir ağları ile ilgili olan çalışmalara teşvik destekleri çekilmiş ve bu alandaki araştırmalar durma noktasına gelmiş, 20 yıllık karanlık bir döneme girmiştir. Ancak bazı bilim adamları çalışmalarına devam etmiştir. 1976'da Grossberg, Adaptif Rezonans Teori (ART)'yi, 1982'de Hopfield dinamik Hopfield ağını (recurrent network) geliştirmiştir.

1986 yılında, Rumelhart ve McClelland çok katmanlı ileri beslemeli ağlarda geriye yayılma algoritmasını ortaya koyarak Minsky ve Papert tarafından ortaya konan problemi çözmüş ve yapay sinir ağlarına olan ilgiyi bir anda artırmışlardır.

Son yıllarda, bu alanda yapılan araştırmalar büyük bir ivme kazanmış ve yapay sinir ağları günümüzde de devamlı gelişme göstermektedir. Hatta bu teknolojinin ticari kısmıyla ilgilenen şirketler ortaya çıkmış ve birçok alana uygulanmıştır.

3.2.1.3 Yapay sinir ağlarının uygulama alanları

Son yıllarda yapay sinir ağları, çözümü güç ve karmaşık olan yada ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmakta ve genellikle başarılı sonuçlar alınmaktadır. Yapay sinir ağları çok farklı alanlara uygulanabildiğinden burada genel bir sınıflandırma ile yapay sinir ağlarının uygulama alanları aşağıdaki gibi özetlenebilir.

- Arıza analizi ve tespiti

Bir sistemin, cihazın veya elemanın (doğru) çalışma şeklini öğrenen bir YSA yardımıyla bu sistemlerde meydana gelebilecek arızaların tanımlanma olanağı vardır. Bu amaçla YSA; elektrik marinalarının, uçakların yada bileşenlerinin, entegre devrelerin v.s. arıza analizinde kullanılmıştır.

- Tıp alanında

EEG ve ECG gibi tıbbi sinyallerin analizi, kanserli hücrelerin analizi, protez tasarımı, transplantasyon zamanlarının optimizasyonu ve hastanelerde giderlerin optimizasyonu v.s. gibi uygulama yeri bulmuştur.

- Savunma sanayi

Silahların otomasyonu ve hedef izleme, nesneleri/görüntüleri ayırma ve tanıma, yani algılayıcı tasarımı ve gürültü önleme vs. gibi alanlara uygulanmıştır.

- Haberleşme

Gürültü ve veri sıkıştırma, otomatik bilgi sunma servisleri, konuşmaların gerçek zamanda çevirisi vs. gibi alanlarda uygulama örnekleri vardır.

- Üretim

Üretim sistemlerinin optimizasyonu, kimyasal proseslerin dinamik modellenmesi ürün analizi ve tasarımı, ürünlerin (entegre, kağıt, kaynak vs.) kalite analizi ve kontrolü, planlama ve yönetim analizi vs. alanlarına uygulanmıştır.

- Otomasyon ve kontrol

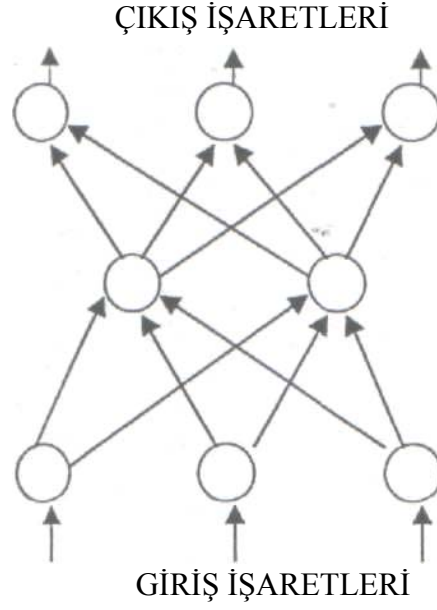
Uçaklarda otomatik pilot sistemi otomasyonu, ulaşım araçlarında otomatik yer bulma/gösterme, robot sistemlerin kontrolü, doğrusal olmayan sistem modelleme ve kontrolü, elektrikli sürücü sistemlerin kontrolü vs. gibi yaygın bir uygulama yeri bulmuştur.

Ayrıca yapay sinir ağılar bankacılık ve finans, maliyet analizi, performans değerlendirmesi gibi daha birçok uygulama alanına sahiptir.

3.2.1.4 Yapay sinir ağlarının özellikleri

Yapay sinir ağı, insan sinir sistemindeki nöronların (sinir hücresi) fonksiyonunu gören yapı taşlarından oluşmaktadır.

Her nöron tipik olarak kendi bağlantılarından çeşitli işaretler alır. Bu işaretlerin bazıları diğer nöronlardan olduğu gibi bazıları da dış dünyadan olabilir. Nöronlar birden fazla çıkış işareti üretirler. Bu işaretler, çok küçük bağlantı parçalarına ayrılan farklı uzaklıklarda son bulan kollardan oluşan çıkış bağlantısı üzerinden gönderilirler. Çıkış bağlantısının her kolu aynı işareti gönderir işaret kollar arasında bölünmez. Dışarı çıkan kolların büyük çoğunluğu ağıdaki diğer nöronlardan gelen bazı kollarda son bulmaktadır. Diğerleri ise ağı dışında son bulmakta ve kontrol işareti veya cevap işareti üretmektedir. Şekil 3.2 tipik olarak sinir ağının fiziksel bağlantı şeklini göstermektedir.



Şekil 3.2 Tipik olarak sinir ağının bağlantı şekli

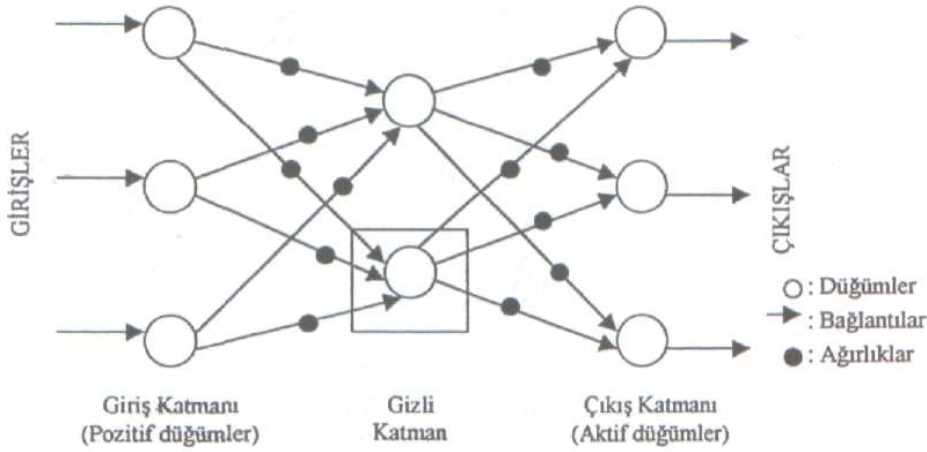
Yapay sinir ağı bağlantıları insan beyninin mimarisinden esinlenmiştir. Ancak yapay sinir ağı nöronları biyolojik nöronların kabataslak benzerleri olduğundan biyolojik nöronların işlevini tam olarak görmezler. YSA'nın temel özellikleri şu şekilde sıralanabilir.

3.2.1.4.1 Bellek ve genelleme

Yapay sinir ağları, verilen giriş bilgisini ağ içerisinde yerel bellekler oluşturarak diğer nöronlara dağıtırlar. Şekil 3.3'de görüldüğü gibi bir yapay nöral ağda her nöronun bir yerel belleği ve sinir ağının o andaki bilgi durumunu belirten nöronlar arası bağlantı ağırlıkları bulunmaktadır. Ağda çıkışı oluşturan giriş bilgisi, ağın içinde bir çok yerel bellek şeklinde dağıtılmış olup nöron ağın bilgiyi bu şekilde saklaması en önemli özelliklerindendir.

Öğrenme işlemi tamamlanmış yani ilgilendiği problemi öğrenmiş bir yapay sinir ağı eğitim sırasında karşılaşmadığı örnekler içinde istenen tepkiyi (çıkışı) üretebilir. Bu özellik, gerçek dünyadaki, çevreden gelen faktörlerle ufak bozulmalara uğramış girdileri fark edebilmek açısından da Önemlidir. Kısaca, yapay sinir ağı belleğinin yapısı; eksik,

gürültülü ve tam seçilemeyen bir giriş uygulandığı zaman da anlamlı çıkışlar elde etmeye uygundur ki buna genelleme denir (Eğrisöğüt 2002).



Şekil 3.3 Üç katmanlı bir yapay sinir ağı

3.2.1.4.2 Eksik bilgilerle çalışabilme ve hata toleransı

Yapay sinir ağı, eğitilme esnasında problemdeki değişikliklere ağırlıklarını ayarlar. Yani ağ, ağırlıkların değiştirilmesiyle eğitilir. Sistem, eğitim kümesinin özünü çıkartır ve saklar. Böylece eksik girişlere uygun şekilde cevap verilebilir. Nöral ağlar, ağdaki her işlem elemanı (nöron) kendi kendine yeterli olduğu ve diğer nöronlarda yapılan işlemlere kayıtsız kendi işini sürdürdüğü için aynı zamanda hata toleranslıdır.

YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağıtılmış bir yapıya sahiptir ve bilgi bütün ağ boyunca yayılı durumdadır. Yani bilgi tek bir yerde saklanmayıp, yerel belleklere dağıtılmıştır. Bu nedenle ağda bazı bağlantıların hatta bazı işlem elemanlarının (nöronların) zarar görmesi veya etkisiz hale gelmesi sistem performansının kötü sonuçlar doğurmayacak şekilde azalmasına yol açmasına rağmen, tamamen başarısız olmasına sebep olmaz. Bunun aksine klasik yöntemlerde, az bir zarar bile sistemi durma noktasına getirir. Bu yüzden klasik yöntemlere göre yapay sinir ağlarının hatayı tolere etme yetenekleri son derece yüksektir.

3.2.1.4.3 Nonlineerlik ve paralellik

Yapay sinir ağlarının temel işlem elemanı olan hücre (nöron) lineer değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da lineer değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özellikleri nedeniyle daha karmaşık problemleri lineer tekniklerden daha doğru çözerler. Matematiksel olarak çözümün zor olduğu nonlineer davranışlar, yapay nöral ağ ile hissedilebilir, algılanabilir ve bilinebilir. Yapay nöral ağlar aynı zamanda son derece paralellğe sahiptir. Bağımsız işlemleri aynı anda çok hızlı yürütebilirler. Paralel donanımlar yapılan gereği yapay nöral ağlara uygun olduğundan alternatiflerinden daha elverişlidirler.

3.2.1.5 Biyolojik sinir sistemi ve nöron yapısı

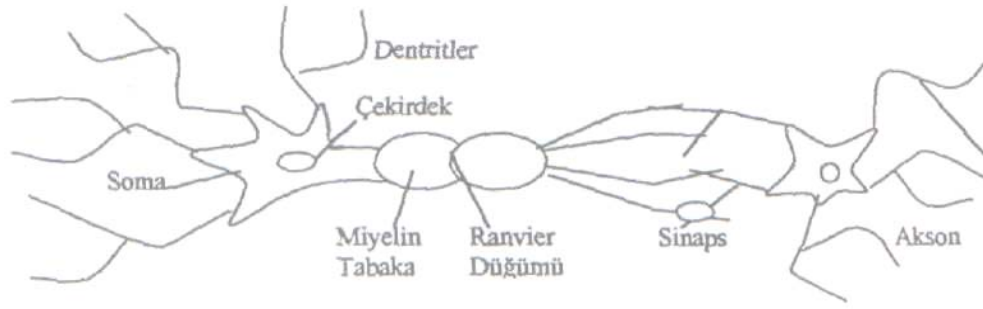
Biyolojik sinir sistemi, merkezinde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin (merkez sinir ağı) bulunduğu üç katmanlı bir sistemdir. Alıcı sinirler organizma içerisinde veya dış dünyadan algıladıkları uyarılan, beyne bilgi üreten elektriksel sinyallere dönüştürür. Tepki sinirleri ise beyin tarafından üretilen elektriksel darbeleri uygun tepkilere dönüştürür. Şekil 3.4'de sinir sisteminin blok gösterimi verilmiştir.



Şekil 3.4 Biyolojik sinir sisteminin blok gösterimi

insan beyninde yaklaşık olarak 10^{10} sinir hücresi (nöron) vardır. Yine hücre başına bağlantı sayısı 10^2 mertebesinde. Beynin temel yapısı sinir hücrelerinden oluşur. Nöronlar beynin haberleşme sistemini oluşturur ve görevi sinyal alma, işlem yapma ve elektro kimyasal sinyallerin sinir ağları içerisinde iletilmesini sağlamaktır.

Bir biyolojik nöron, hücre gövdesi, dendritler, aksonlar olmak üzere üç bileşenden oluşmaktadır. Şekil 3.5'de bir biyolojik nöronun şematik yapısı görülmektedir.



Şekil 3.5 Bir nöronun şematik yapısı

Dendritler diğer hücrelerden elektriksel bilgi alan pasif nöron girişleridir. Aksonlar ise diğer hücrelere karşı darbelerin yani çıkış darbelerinin üretildiği elektriksel olarak aktif nöron çıkışlarıdır. Bir hücrenin dendrit ile diğer hücrenin aksonunun bağlantı noktasına da sinaps denir. Miyelin tabaka işaretin yayılma hızını artırmak için çekirdek ile hücreler arası sıvı kapasitesini düşüren bir yalıtım maddesidir. Ranvier düğümü ise işareti periyodik olarak yeniden üretmeye yarayan boğumlardır.

Sinapsa gelen ve dendritlerin aldığı bilgiler genellikle elektriksel darbelerdir, fakat sinapsdaki kimyasal ileticilerden etkilenir. Belirli bir sürede bir hücreye gelen girişlerin değeri, belirli bir eşik seviyesine ulaştığında hücre bir tepki üretir. Hücrenin tepkisini artırıcı yöndeki girişlere uyarıcı, azaltıcı yöndeki girişlere ise önleyici girişler adı verilir ve bu etkiyi sinaps belirler.

Mühendislik dilinde nöron giriş lojik bir olduğu durumlarında elektriksel darbeler üreten ve lojik sıfır durumunda ise çıkış vermeyen bir elemandır.

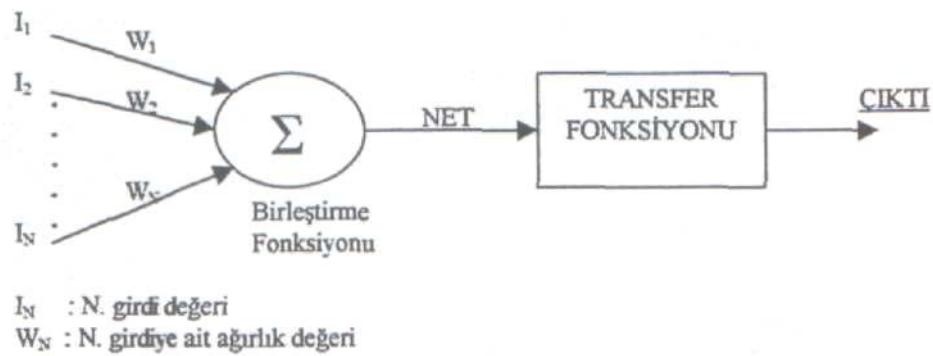
İnsan beyni; öğrenme, birleştirme, uyarılma ve genelleştirme yeteneği nedeniyle son derece karmaşık, doğrusal olmayan ve paralel dağıtılmış bir bilgi işleme sistemi olarak tanımlanabilir (Eğrisöğüt 2002).

3.2.1.6 Yapay nöron

Yapay nöron, biyolojik nöronun girdi, işlem ve çıktı karakteristiğini taklit ederek tasarlanmıştır. Tıpkı biyolojik nöron gibi, işlem elemanına (nöron) birden fazla giriş gelmekte ve sadece bir çıkış gitmektedir. Bu çıkış işareti diğer işlem elemanlarına giriş oluşturmak üzere birçok yola dağılır veya dış dünyaya aktarılır.

Her işlem elemanının yerel belleği bulunur. Bu bellekte uyarlanan katsayıların önceki hesaplamalar ile belirlenen değerleri saklanır.

Yapay sinir ağının en temel birimi olan işlem elemanı olarak da adlandırılan bir yapay nöronun yapısı şekil 3.6'de gösterilmiştir.



Şekil 3.6 Bir yapay nöronun yapısı

Girdiler ağ dışı veya diğer işlem elemanlarından bir işlem elemanına gelen bilgilerdir. Bir nörona uygulanan girişler, $I = [I_1 \ I_2 \ \dots \ I_N]^T$ vektörü şeklinde gösterilebilir.

Yapay nöron ağın giriş veri tipleri ikili (binary) 0-1 veya sürekli değerlerdir. Sürekli değer girişlerinde giriş bilgi değerleri 0-1 arasında olacak şekilde normalize edilmelidir.

3.2.1.6.1 Ağırlıklar

İşlem elemanları arasında bağlantılar bulunur ve her bağlantının bir ağırlığı vardır. Ağırlıklar girdilerin işlem elemanı üzerinde etkisini kontrol ederler. $w_1, w_2, \dots, w_N$ ağırlıkları vektörel olarak w ile gösterilebilir ve sabit veya değişken değerler alabilir. Değişken değer alan ağırlıklar, öğrenme sırasında sürekli olarak değişerek ağırlık girdi ve çıktı arasındaki ilişkiyi yakalayabilmesine çalışırlar. Yani bu ağırlıklar öğrenme sırasında optimize edilir. Ağırlık büyüdükçe bir işlem elemanının diğeri üzerindeki etkisi de artar. Ağırlığın sıfır olması hiçbir etkinin olmaması demektir.

3.2.1.6.2 Birleştirme fonksiyonu

Bir İşlem elemanından gelen bilgileri birleştirme görevi yapar. Yani birleştirme fonksiyonu, bir nörona gelen net girdiyi hesaplayan bir fonksiyondur. En çok kullanılan birleştirme fonksiyonu tipleri çizelge 3.2'de gösterilmiştir.

Bunlardan en yaygın kullanılanı, gelen bilgilerin (girdilerin) ilgili bağlantıların ağırlıkları ile çarpılıp toplanması ile net girdiyi belirleyen "toplam fonksiyon" dur. En uygun birleşme fonksiyonu deneme yanılma ile bulunur.

Çizelge 3.2 En çok kullanılan birleştirme fonksiyonları

Toplam $Net_i = \sum_j w_{ij} I_j$	Çarpım $Net_i = \prod_j w_{ij} I_j$
Maksimum $Net_i = \text{Max}(w_{ij} I_j)$	Minimum $Net_i = \text{Min}(w_{ij} I_j)$
Çoğunluk $Net_i = \sum_j \text{Sgn}(w_{ij} I_j)$	Kümülatif toplam $Net_i = Net_{\text{eski}} + \sum_j w_{ij} I_j$

Çizelge 3.2 En çok kullanılan birleştirme fonksiyonları (devam)

Net_i = İşlem elemanına giren net girdi

I_j = İşlem elemanının çıktısı

W_{ij} = i ve j işlem elemanları arasındaki bağlantının ağırlığı

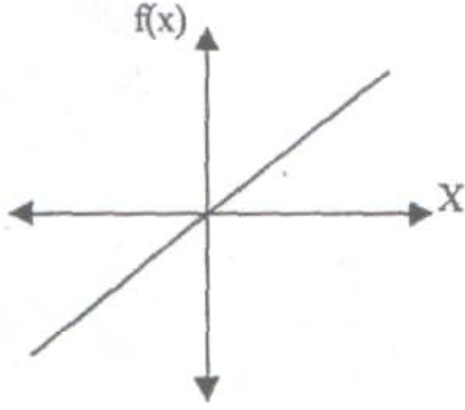
3.2.1.6.3 Transfer fonksiyonu

Eşik veya işaret fonksiyonları diye de adlandırılan transfer fonksiyonu, birleştirme fonksiyonunun sonucunu değerlendirir. Yani birleştirme fonksiyonu tarafından belirlenen net girdiyi alarak işlem elemanının çıkışı belirleyen fonksiyondur. Genel olarak türevi alınabilen bir fonksiyon olması tercih edilir. Değişik transfer fonksiyonları çizelge 3.3'de verilmiştir. Birleştirme ve transfer fonksiyonları problemin yapısına göre seçilir.

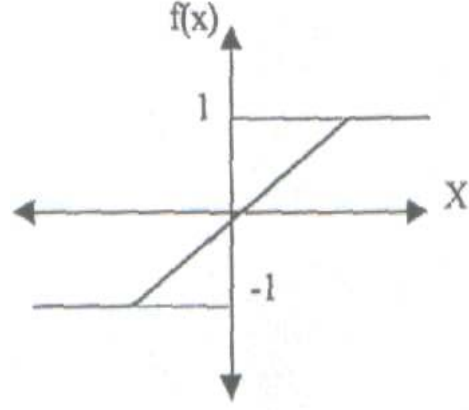
Çizelge 3.3 En çok kullanılan transfer fonksiyonları

Lineer Fonksiyon $f(x) = x$	Sinusoidal $f(x) = \sin(x)$
Sigmoid Fonksiyon $f(x) = \frac{1}{1 + e^{-x}}$	Hiperbolik Tanjant Fonksiyonu $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Eşik Mantıksal Fonksiyon $f(x) = \begin{cases} 0 & x \leq 0 \\ x & 0 < x < 1 \\ 1 & x \geq 1 \end{cases}$	Adım fonksiyonu $f(x) = 1$ eğer $x >$ eşik değeri $f(x) = 0$ diğer durumlar

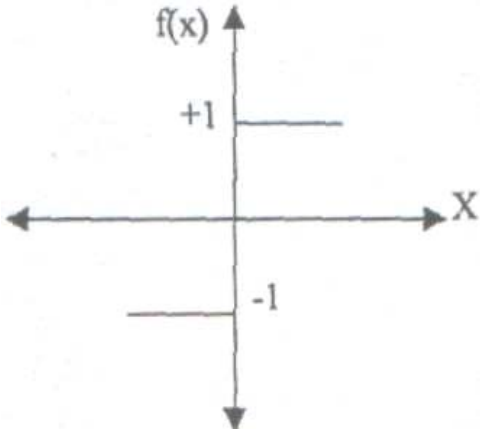
En yaygın kullanılan eşik fonksiyonları şekil 3.7'de görülmektedir.



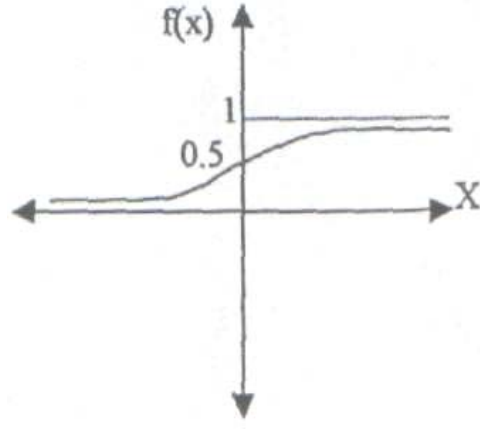
a) Lineer



b) Rampa



c) Basamak



d) Sigmoid

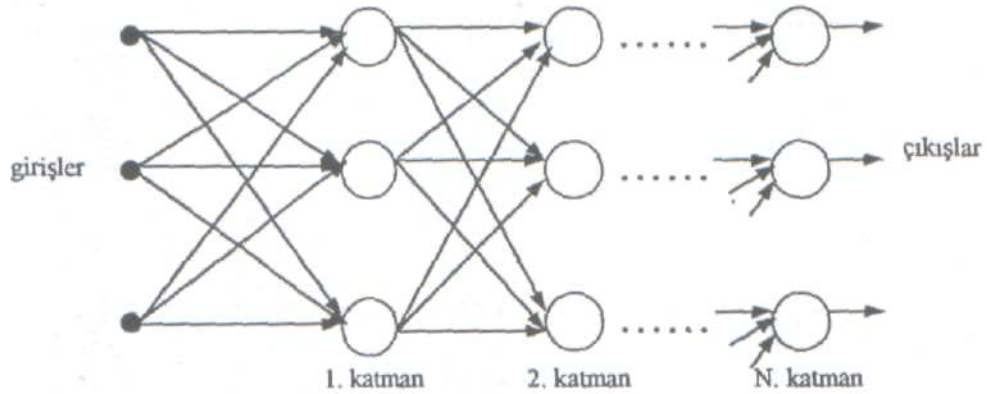
Şekil 3.7 Eşik fonksiyonları

3.2.1.7 Yapay sinir ağıının yapısı

Bir işlem elemanı (nöron) hiçbir zaman tek başına çalışamaz. Nöronlar çok sayıda bağlantı içeren bir ağ içinde bulunur.

Nöronlar arasındaki bağlantıların nasıl oluşturulduğu ağın yapısını belirler. Nöronların bir grubu, katman olarak adlandırılan bir yapı oluşturur. Katmanlarda bulunan nöronlar, tam bağlantılı, kısmi bağlı veya ikisinin kombinasyonu halinde olabilir. Genellikle bir katmandaki tüm işlem elemanları, aynı birleştirme ve transfer fonksiyonu ile aynı öğrenme kuralına sahiptir.

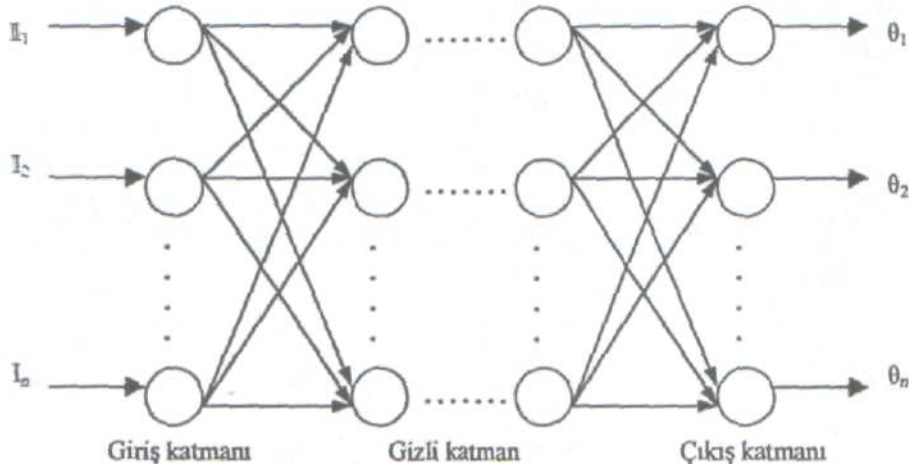
Araştırmacılar tek katmanlı ağların çeşitli uygulamalarda sınırlı yetenekler gösterdiklerini ispatlamışlardır. Bu ağlar, iki veya daha fazla nöral katmanını bağlayarak oluşturulan çok katmanlı ağlara yol açmışlardır. Çok katmanlı ağların çalışması, tek katmanlı ağların çalışmasına benzer şekilde her katmanın çıkışı, önceki katmanın çıkışlarından ağırlıklı toplam olarak üretilir. Çok katmanlı nöral ağ şekil 3.8’de gösterilmiştir .



Şekil 3.8 Çok katmanlı nöral ağ

Bir yapay sinir ağı temel olarak, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere birbirine bağlanan üç katmandan oluşur. Şekil 3.9’de bir yapay sinir ağı ve katmanları verilmiştir.

Giriş katmanı dışarıdan gelen bilgiyi kabul eder ve diğer katmanlara iletir. Bu katmanda bilgi İşlenmez.



Sekil 3.9 Çok katmanlı bir sinir ağı ve katmanları

Gizli katman bilgiyi işler. Girdi çıktı ilişkisini kavramada asıl ağırlık gizli katman üzerindedir. Gizli katman sayısı birden fazla olabilir.

Çıkış katmanı ise sorgulanan problem üzerinde ağıın kararını dış ortama bildiren işlem elemanlarından oluşan tabakadır.

3.2.1.8 Yapay sinir ağlarında öğrenme ve eğitim

Klasik bilgi işleme yöntemlerinin çoğu programlama yoluyla hesaplamaya dayanmaktadır. Yani bir problemin çözümü için probleme yönelik bir algoritma geliştirilmelidir. Bunu yanında bu yöntemler tam tanımlı olmayan problemleri çözemez. Yapay sinir ağları ise belirli bir probleme göre programlanmadığı halde o problemi çözmeyi öğrenebilir.

Genel anlamda Öğrenme, sinir ağının, giriş uyarıcılarını kullanarak kendini istenen sonuçları vermek üzere adapte etmesidir.

İşlem elemanı ve ağ yapısı tasarlandıktan sonra, YSA'nın öğrenme işlemi başlatılabilir. Eğitim ve öğrenme hemen hemen bütün nöral ağların temelidir. Öğrenme, ağdaki nöronların değiştirilmesi ile değil, nöronlar arasındaki bağlantı ağırlıklarının değiştirilmesi ile sağlanır. Tek bir nöronun çıkışının nasıl belirlendiği göz önüne alındığında; nöronun transfer fonksiyonunun sabit olması koşuluyla çıkışını yalnızca, giden işaretin ve nörona giriş bağlantı ağırlıklarının belirlediği bilinmektedir. Nöronun gelen işarete doğru cevap vermesinde ve performansının artırılmasında en önemli eleman bağlantı ağırlıklarıdır.

Öğrenme ile eğitim aynı anlama gelmez. Öğrenme, ağın içerisinde geliştirilen bir aktivite iken eğitim ağa dışarıdan bir etkidir. Eğitim algoritması eldeki problemin özelliklerine göre öğrenmeyi yapay sinir ağına nasıl adapte edeceğimizi belirtir. Ağın öğrenmesi, ağın ağırlıklarının uygun değer alması olup; eğitim ağın ağırlıklarının uygun değer alması için kullanılan yöntemdir.

3.2.1.8.1 Yapay sinir ağlarının eğitilmesi

Yapay sinir ağlarının örnekler ile eğitilmesi üç ayrı yol ile gerçekleştirilebilir.

1. Öğreticili eğitim (supervised training)
2. Skor ile eğitim (graded training)
3. Kendini düzenleme ile eğitim (self - organization training)

Yapay sinir ağları üç yöntemde de insan beyni gibi örnekler ile eğitilir. Ağ, bağlantı ağırlık değerleri değiştirilerek örnekler göre eğitilir ve herhangi bir örnek ile yeniden karşılaştırıldığında uygun cevabı üretir. Öğrenmede iyi bir model kullanılarak ağırlıkların bu modele göre değiştirilmesi esastır.

3.2.1.8.1.1 Öğretici eğitim

Öğretici eğitimde ağ giriş ve çıkış bilgisi (hedef vektörü) verilir. Ağ, verilen girişlere karşı doğru (hedeflenen) çıkışı üretmek hedefiyle kendini ayarlamaktadır. Ağ kendi çıkışını

doğru cevaplar ile karşılaştırır ve hatayı bulur. Çıkış hatası istenen düzeye ininceye kadar ağ nöronlar arasındaki ağırlıkları değiştirerek iterasyona devam eder. Bağlantı ağırlıklarındaki değişim, geri besleme olarak gelen hatayı azaltacak ve hatta mümkünse sıfırlayacak şekilde ayarlanır. Bunun için de çeşitli öğrenme kuralları bulunur.

Öğreticili eğitme şu örnekle açıklanabilir, yabancı bir dil öğrenirken, öğretmenden duyulan bir kelimenin telaffuzu beyne kaydedilir. Bu kelime öğrenci tarafından tekrar söylenir ve kayıtlı olan sesle karşılaştırma yapılır. Tekrarlama işlemi kelimenin doğru telaffuzu yakalanana kadar devam eder. Kelime doğru söylenmeye başlandığında tekrarlama durur.

3.2.1.8.1.2 Skor ile eğitme

Skor ile eğitme, öğretici eğitmeye benzemektedir. Ancak, bu yöntemde giriş işaretlerine karşılık çıkış işaretleri ağa tam olarak verilmez. Çıkış işareti yerine skor verilerek ağın değerlendirmesi yapılır. Yani ağ çıkışının durumuna göre, "doğru", "yanlış" veya "başarılısın", "başarısızsın" gibi değerlendirmelerle ağ performansı ile ilgili geri besleme bilgisi verilir. Kontrol uygulamalarında yaygın olarak kullanılır.

3.2.1.8.1.3 Kendini düzenleme ile eğitme

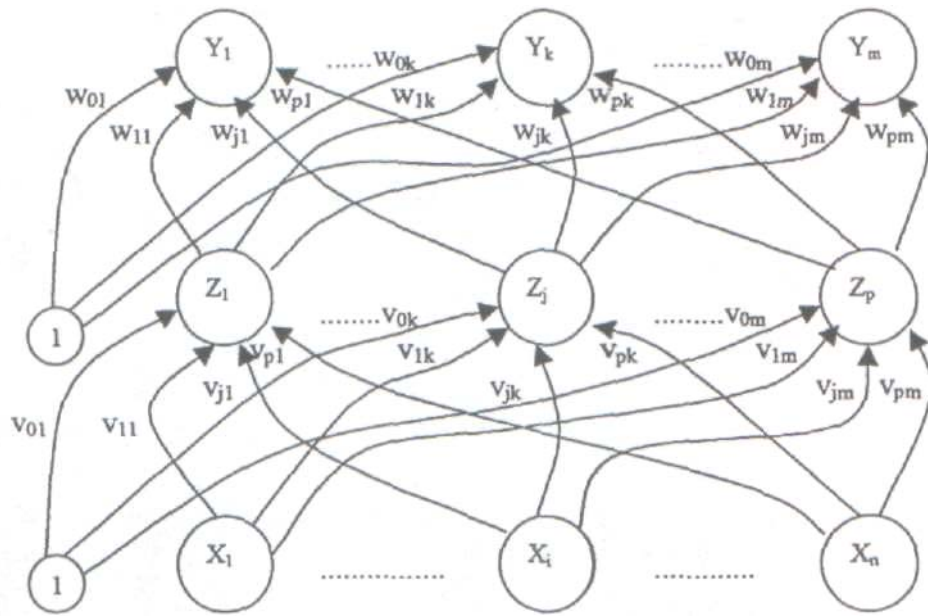
Kendini düzenleme ile eğitmede, ağa giriş bilgisi verilir, fakat ağın ürettiği çıkışın doğruluğuna dair bir geri besleme gelmemektedir. Yani hedef çıkışlar verilmemektedir. YSA'nın bağlantı ağırlık katsayıları, oluşan hatayı göz önüne almadan bir amaç ölçüt uyarınca, sadece girişlere göre değişmektedir.

3.2.1.9 Geri yayılım algoritması

Geri yayılım algoritması çok katmanlı yapay sinir ağlar için günümüzde en çok kullanılan sistematik bir metottur. Çok güçlü bir matematik esasa sahiptir. Geri yayılım algoritması yapay sinir ağların uygulanabildiği problem aralığının

genişlemesine yol açmıştır. Çok sayıda başarılı ve güçlü uygulamaların üretilmesine imkan vermiştir.

Geri yayılım ağı algoritmasını ilk olarak 1974 yılında Paul Werbos ortaya atmıştır. Ancak bilim dünyasında kabulü 1986 yılında Rummelhart ve Parker tarafından geliştirilmesiyle mümkün olmuştur. Geri yayılım yöntemi günümüzde, ses tanıma problemlerinden non-lineer sistem tanımlama ve kontrol problemlerine kadar yapay sinir ağı kullanılarak çözüm bulan birçok alanda başarıyla kullanılmaktadır.



Şekil 3.10 Geriye yayılım ağı mimarisi

Diğer ağlarda olduğu gibi geri yayılım ağını, nöronlar arası bağlantılar, nöronların kullandığı transfer fonksiyonları ve ağı eğitimi sağlayan ağırlıkların değişim kuralları belirler. Geri yayılım ağı çok katmanlı ileri beslemeli yapıya sahip bir ağıdır ve bir katmandaki nöronlar bir önceki ve bir sonraki katmanlardaki nöronlarla bağlantılıdır. Ancak katmanlardaki nöronlar arasında ve bir katmandan öteki katmana geriye doğru bağlantı bulunmaz. Geriye yayılım ağı, ağ çıkışı ile istenen çıkışın karşılaştırılması ile bulunan hatayı dikkate alarak, karesi alınmış hata fonksiyonunu

minimize eden kodlu bir algoritmadır. Şekil 3.10'da geri yayının ağ mimarisi gösterilmiştir.

Veri girişleri, giriş düğümleri $X_1 \dots X_2 \dots X_n$ 'den yapılır. Giriş düğümleri hesaplanabilir değildir. Her nöron için nöronun Bias'ı ya da referans seviyesi olarak adlandırılan, bir eşik değer girişi mevcuttur. Bütün saklı nöronlar tüm veri girişlerini alırlar. Fakat her biri farklı ağırlık kümesine sahip oldukları için, değerler kümesi de farklıdır. Her bir nöron kendisine gelen işareti işleyerek sonucu çıkış nöronuna iletir. Çıkışlar da farklı ağırlık kümesine sahiptir.

Geri yayının ağında eğitim; giriş eğitim örneklerinin ileri yayılması, hesaplama ve ilgili hataların geriye yayını ve son olarak ağırlıkların ayarlanması olmak üzere üç aşamada gerçekleşir. Eğitim tamamlandıktan sonra sadece ileri yayının fazı kullanılır. Eğitilmiş ağ oldukça hızlı çıkış üretebilir. Bir gizli katmanın genellikle yeterli olmasına rağmen bazı uygulamalarda birden fazla gizli katman faydalı olabilir.

Geri yayının ağında çıkış katmanını eğitmek bağıl olarak basittir. Sorun hiçbir hedef vektörünün olmadığı ara katman ağırlıklarını eğitirken ortaya çıkar. Bunun çözümü, hata işaretinin çıkıştan daha iç katmanlara yayını ve sağlamasıyla bulunur. Geri yayının algoritmasının matematiksel temelinden bahsedecek olursak, geri yayının algoritması çok katmanlı ağlar içinde özgün bir ağırlık azaltma yöntemidir. Tekniğin temel mantığı ayarlanabilir ağırlık faktörü "w" yi verilen giriş için hesaplamaktır. Algoritma sırasında her giriş verisi sonuç üretmek için gizli katmandan geçirilir. Çıkış katmanındaki hatanın bulunabilmesi için beklenen sonuç ile elde edilen sonuç karşılaştırılır. Daha sonra çıkış hatasının türevi çıkış katmanından geriye doğru gizli katmandan geçirilir. Hata değeri bulunduğundan sonra nöronlar kendi hatalarını azaltmak için ağırlıklarını ayarlarlar. Geri yayının eğitme algoritması hata ölçüsü olarak ortalama hata karesini kullanır. Bu bir gereklilik değildir. Herhangi türevlenebilir hata fonksiyonu kullanılabilir. Ancak seçilen hata fonksiyonu seçilen ve gerçek değer arasındaki farkın ölçüsünü sağlamalıdır.

Geri yayılım algoritması şekil 3.11’de gösterilmiştir. Bu algoritma ile i ve j kat işlem elemanları arasında ki ağırlıklardaki $\Delta w_{ji}(t)$ değişikliği hesaplanır. Bu ifade,

$$\Delta w_{ji}(t) = \eta \delta_j x_i + \alpha \Delta w_{ji}(t-1) \quad (3.1)$$

olarak verilir. Eşitlik 3.1’de η öğrenme katsayısı, α momentum katsayısı ve δ_j ara ve çıkış katındaki herhangi bir j nörona ait bir faktördür. η eğitme katsayısı eğitimin adım ölçüsünü belirleyen bir sabittir. η ’yu büyük seçmek osilasyona neden olur.

Çıkış katı için faktör (δ_j) aşağıdaki gibi verilir (eşitlik 3.2).

$$\delta_j = \left(\frac{\partial f}{\partial net_j} \right) (y_j^t - y_j) \quad (3.2)$$

Burada $net_j = \sum x_j w_{ji}$ ve $y_j(t)$ ise j. İşlem elemanının hedef çıkışıdır. Ara katmanlardaki nöronlar için ise bu faktör eşitlik 3.3’de gösterildiği gibi verilir.

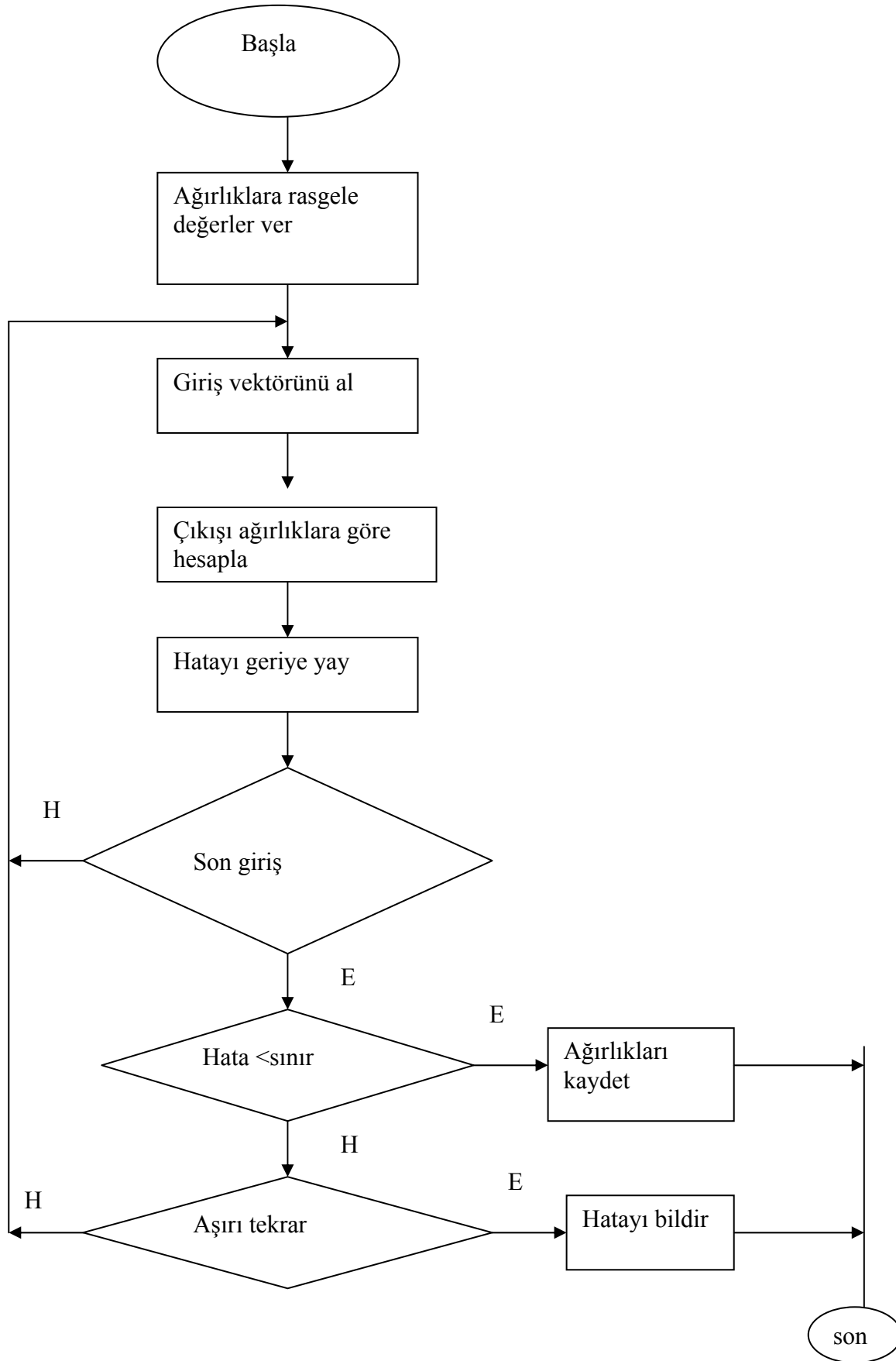
$$\delta_j = \left(\frac{\partial f}{\partial net_j} \right) \sum w_{qi} \delta_q \quad (3.3)$$

buradaki δ_q eşitlik 3.4’de ki gibidir.

$$\delta_q = y_j^t - y_j \quad (3.4)$$

W_{qi} ara katman ağırlığıdır. Ara katman için hedef çıkış değeri olmadığı için eşitlik 3.2 yerine eşitlik 3.3 kullanılır. Bu duruma bağlı olarak çıkış katından başlayarak δ_j faktörü, bütün katlardaki nöronlar için hesaplanır. Daha sonra eşitlik 3.1’deki formüle bağlı olarak bütün ağırlıklar için güncelleştirilmesi gerçekleştirilir.

Geriye yayılım öğrenme algoritmasının en büyük özelliği görüldüğü gibi, hatalara karşı verdiği cevapta, ağırlıkların değerini değiştirebilmesidir. Ağ çıkış hatalarının türevi çıkış katmanından geriye doğru ara katmanlara iletilir. Geriye yayılım algoritmasına İsmi veren de hataların geriye doğru yayılması özelliği olmuştur.



Şekil 3.11 Geri yayılım algoritması

3.2.1.9.1 Öğrenme oranı, momentum katsayısı ve bias

Öğrenme oranı, bağlantı ağırlıklarındaki değişme oranını belirler ve 0-1 arasında değer alır. Öğrenme oranı büyüdükçe değişim miktarı artmakta ve çok büyük değerleri kararsızlığa neden olmaktadır. Öğrenme oranının çok küçük değerleri ise öğrenme işleminin yavaşlamasına yol açar.

Momentum katsayısı, ağırlıkların daha hızlı toparlanmasına yardım eden ve bir önceki değişimin bir kısmını bir sonraki değişime ekleyerek ağırlıkları yenilemek için kullanılan bir faktör şeklinde ifade edilebilir. Bunun yapılmasının, öğrenme esnasında lokal minimum değerlerinin bulunmasını yani ağırlıkların salınımını engellemektir. 0-1 arasında değerler alabilen momentum katsayısının yüksek değerleri çoğunlukla öğrenme işlemini hızlandıracığından daha uygundur.

Bias sabit aktivasyon değeriyle özel işlem elemanıdır. Bir geri yayılım ağında bias girdileri kullanılarak, daha iyi öğrenmeye neden olacak, aktivasyon fonksiyonunun orijini dengeye getirilmeye çalışılır.

3.2.1.10 Levenberg-Marguardt metodu (LMM)

Temel olarak bu algoritma maksimum komşuluk fikri üzerine kurulmuş, genel olarak bu metod yavaş yakınsama problemlerinden etkilenmez. İleri beslemeli ağlarda en hızlı öğrenme metodudur.

$E(w)$ 'nin bir amaç fonksiyonu olduğu düşünülürse m tane hata terimi için $e_i^2(w)$ eşitlik 3.5'te verilmiştir.

$$E(w) = \sum_{i=1}^m e_i^2(w) = \|f(w)\|^2 \quad (3.5)$$

bu eşitlikte;

$$e_i^2(w) \equiv (y_i - y_{d_i})^2 \text{ dir.}$$

Burada amaç fonksiyonu $f(\cdot)$ ve onun jakobiyeini j 'nin bir noktada w bilindiği farz edilir.

LMM’de hedef, parametre vektörü w ’nin $E(w)$ minimum iken bulunmasıdır. LMM’nin kullanılmasıyla yeni vektör w_{k+1} farzedilen vektör w_k ’dan eşitlik 3.6’teki gibi hesaplanır.

$$w_{k+1} = w_k + \delta w_k \quad (3.6)$$

burada δw_k eşitlik 3.7’da ki gibi verilir.

$$(J_k^T J_k + \lambda I) \delta w_k = -J_k^T f(w_k) \quad (3.7)$$

eşitlikte;

J_k : f ’nin(hatanın) w_k değerlendirilmiş Jakobiyesi

λ : Marquardt parametresi

I : birim veya tanımlama matrisi

Levenberg-Marquardt algoritması aşağıdaki gibi özetlenebilir.

- i) $E(w_k)$ ’yı hesapla
 - ii) Küçük bir λ değeri ile başla(mesela $\lambda=0.01$)
 - iii) δw_k ’yı eşitlik (3.7) çöz ve $E(w_k + \delta w_k)$ değerini hesapla
 - iv) şayet $E(w_k + \delta w_k) \geq E(w_k)$ λ ’yı 10 kat artır ve (iii)’ye git
 - v) şayet $E(w_k + \delta w_k) < E(w_k)$ λ ’yı 10 kat azalt
- w_k : $w_k \leftarrow w_k + \delta w_k$ ’yı güncelleştir ve (iii)’ye git.

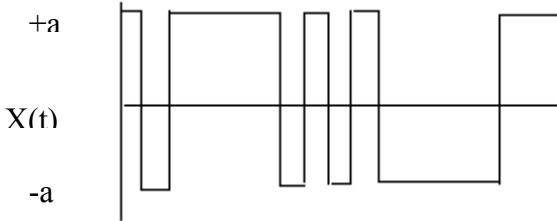
Hedef çıkışı hesaplamak için birçok katmanlı nöronun LMM kullanılarak öğretilmesi ağırlık dizisi w_0 ’a bir başlangıç değerinin atanması ile başlar ve hataların karelerinin toplamı e_i^2 ’nin hesaplanmasıyla devam eder. Her e_i^2 terimi, hedef çıkışı (y) ile gerçek çıkış (yd) arasındaki farkın karelerini ifade eder. Bütün veri seti için e_i^2 hata terimlerinin tamamının elde edilmesiyle, ağırlık dizileri (i)’den (v)’e kadar olan LMM adımlarının uygulanmasıyla açıklandığı gibi adapte edilir.

3.2.2 Sistem tanımlanması

Sistem tanımlamada girdi sinyali olarak kullanılan bir çok sinyaller mevcuttur, sistemin bu sinyallere karşı verdiği tepkiler ölçülerek kaydedilir. Sistem, bu kaydedilen verileri kullanarak modellenir. Bu çalışmada PRBS (sahte rasgele ardışık ikili sinyal) sinyali kullanılmıştır. PRBS sinyalinin özellikleri aşağıda verilmiştir.

- 1) Belli bir zaman aralığında ($t=0, \Delta t, 2\Delta t, 3\Delta t, \dots$) $+a, -a$ gibi iki genlik seviyesine ulaşır. Başka genlik değerlerini almaz
- 2) Gelecekte alacağı değerlerin bilindiği, gürültü içermediği varsayılan tanımlı bir sinyal olan PRBS tanımlı ve belirli periyotlarla tekrarlanan bir sinyaldir.

Bir sisteme PRBS sinyal uygulanacağı zaman $\pm a$ genlik seviyesi, Δt araştırmacı tarafından seçilir. Dikkat edilmesi gereken nokta PRBS sinyalinin büyüklüğünün prosesin işletim koşullarının içinde olmasıdır. Tipik PRBS sinyali şekil 3.12’de gösterilmiştir. PRBS sinyalinin oluşturulmasında geriye kaydırmalı sayaç yöntemi (shift register) kullanılır. Geriye kaydırmalı Sayaç yönteminde kullanılan sayaç sayısı kullanıcı tarafından belirlenir.



Şekil 3.12 PRBS sinyali

PRBS sinyali sisteme verdiği etki, doğrusal veya doğrusal olmayan modellerin parametrelerinin hesaplanmasında kullanılır.

Geleceği tahmin gelişmenin vazgeçilmez unsurlarındandır, tahminde zaman serileri kullanılmaktadır. Geleceği tahmin; incelenen zaman serisinin bugünkü ve geçmiş

değerlerine bakarak bu serinin gelecekte alacağı değerleri tespit etmeye çalışma işlemidir. Bilgisayar teknolojisinin gelişmesi ile tahmin metotlarının uygulanma işlemi daha da kolaylaşmış ve yaygınlaşmıştır.

Zaman serileri belirli bir zaman aralığına göre dizilmiş ve arka arkaya toplanmış deneysel değerlerden meydana gelmiştir. Yani gözlem verileri birbirlerine bağımlıdır. Bu özelliğe iç bağımlılık denir. Sıralamada t , bugünkü zamanı, X_t ise t zamanındaki deneysel değerini gösterir. Ayrıca $t=1,2,3,\dots$ için geçmiş zaman, $t-1, t-2, \dots$ ile geçmiş zaman gözlem değerleri de X_{t-1}, X_{t-2} ile gösterilir. Gelecek zaman değerleri $t+1, t+2, \dots$ şeklinde ve gelecek zaman gözlem değerleri de $X_{t+1}, X_{t+2}, \dots$ ile gösterilir. Bu öngörü modeli önemli stokastik (değişken) sürece dayanır. Doğrusal durağan süreç istatistiksel bir dengeyi ifade etmektedir. Özellikle, gözlem değerleri sabit ortalama etrafında değişim göstermektedir. Sistemde yapılan bazı deneyler sonucunda elde edilen veriler kullanılarak model belirlenmeye çalışılır. Varsayılan model deneysel verilerle karşılaştırılarak model belirlenmeye çalışılır. Varsayılan modelden elde edilen veriler deneysel verilerle karşılaştırılarak uygunluğu test edilir.

Sistem teorisinde, sistem tanımlaması, dinamik bir sistemin transfer fonksiyonunun matematiksel-fiziksel gösteriminin oluşturulması şeklinde ifade edilmektedir. Transfer fonksiyonunun oluşturulması için farklı pek çok model tanımlanmıştır. Bu tür modeller hem zaman hem de frekans bölgesinde incelenmektedir. Zaman bölgesinden frekans bölgesine geçiş ise Fourier dönüşümü ile gerçekleşmektedir. Bazı durumlarda sistemin davranışı fiziksel yasalar ile ideal bileşenler dikkate alınarak matematiksel model ile tanımlanabilmektedir. Bu tanımlama bir beyaz kutu (white-box) model adını almaktadır. Bu tip model de sürecin bütün bilgisi ele alınmaktadır. Ancak, sistemin bütün bilgisini ele almak her zaman mümkün olmamaktadır.

Beyaz kutu (white box) model ile sistem tanımlamada kullanılan etkin bir kestirim metodu ise Kalman-filtreleme tekniğidir. Hemen hemen bütün modellere uygulanabilmektedir. Temel mantığı, oluşturulan modelin sistem eşitliği ile gerçek modelin davranışlarının izlendiği gözlem eşitliğini en küçük kareler dengelemesi ile birleştirmektedir. Modelin sistem eşitliği, Sonlu Elemanlar metodu ya da benzer

metotlarla nümerik olarak belirlenmektedir. Bu şekilde sistem tanımlamakta ve model kalibre edilmektedir.

Beyaz kutu modeller ile sistem tanımlama çok faydalı bir yaklaşım olmasına rağmen, sistem çok karmaşık tanımlanabilmekte ya da yeterince öncül bilgi elde edilememektedir. Bu yüzden sistem, giriş (etki) ve çıkış (tepki) ölçülerinden faydalanılarak tanımlanmaktadır. Bu tür modeller de siyah-kutu (Black-box) model olarak adlandırılmaktadır. 3.8 eşitliği siyah kutu modelin genel bir ifadesidir.

$$y(t) = G(q)u(t) + H(q)e(t) \quad (3.8)$$

$y(t)$; model çıktısı (tepki), $u(t)$; model girişi (etki), $e(t)$; beyaz gürültü, $G(q)$; transfer fonksiyonu, $H(q)$; bozucu

etkileri göstermektedir. Model hem deterministik (belirleyici) $G(q)u(t)$, hem de stokastik(değişken, rastlantısal) $H(q)e(t)$ bölümlerden oluşmaktadır. Siyah kutu modeller parametre sayısına bağlı olarak parametrik ve parametrik olmayan modeller olarak ikiye ayrılmaktadır. Eğer modeldeki parametre sayısı sonlu ise simulasyon, prediksyon yada dizayn için parametrik modeller tercih edilmektedir. Çünkü bu tür modellerle çalışmak çok daha kolay olmaktadır. Zaman bölgesinde tanımlanan parametrik modellerin genel eşitliği 3.9 ile tanımlanmaktadır.

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (3.9)$$

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{na}q^{-na}$$

$$B(q) = b_1q^{-1} + b_2q^{-2} + \dots + b_{nb}q^{-nb}$$

$$C(q) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_{nc}q^{-nc}$$

$$D(q) = 1 + d_1q^{-1} + d_2q^{-2} + \dots + d_{nd}q^{-nd}$$

$$F(q) = 1 + f_1q^{-1} + f_2q^{-2} + \dots + f_{nf}q^{-nf}$$

Burada $A(q)$, $B(q)$, $C(q)$, $D(q)$, $E(q)$, ve $F(q)$, modelin polinomlarıdır. Aşağıda ifade edilen parametrik modeller 3.9 eşitliği ile tanımlanan genel modelin özel durumlarıdır. Model yapıları arasındaki temel fark bozucu etkilerin modellenme şekline kaynaklanmaktadır. Dış kaynak girişli oto-Regresif model (ARX(na nb nk)-Auto-Regressive with eXogenous Input): $C(q)=D(q)=F(q)=1$ ve $A(q)$ ile $B(q)$ ölçülere bağlı

olarak belirlenen polinomlardır. Bozucu etkiler otoregresif (AR) ile filtrelenmiş beyaz gürültü olarak modellenmektedir(eşitlik 3.10). n_a ve n_b parametreleri modelin derecesini, n_k ise gecikmeyi ifade etmektedir.

$$y(t) = \frac{B(q)}{A(q)} u(t) + \frac{1}{A(q)} e(t) \quad (3.10)$$

Dış kaynak girişli otoregresif Hareketli Ortalama model (ARMAX(n_a n_b n_c n_k)- Auto-Regressive Moving Average with eXogenous Input): $D(q)=F(q)=1$ ve bozucu etkiler, otoregresif ortalama hareket (ARMA) ile filtrelenmiş beyaz gürültü olarak modellenmektedir (eşitlik 3.11). n_a , n_b ve n_c parametreleri modelin derecesini, n_k ise gecikmeyi göstermektedir.

$$y(t) = \frac{B(q)}{A(q)} u(t) + \frac{C(q)}{A(q)} e(t) \quad (3.11)$$

Çıkış hatalı model (OE-Output Error (eşitlik 3.12)): $A(q)=C(q)=D(q)=H(q)=1$ dir.

$$y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \quad (3.12)$$

Box-Jenkins model (BJ-Box Jenkins(eşitlik 3.13)): $A(q)=1$ dir.

$$y(t) = \frac{B(q)}{F(q)} u(t) + \frac{C(q)}{D(q)} e(t) \quad (3.13)$$

Sistem tanımlamada genel modeller ortaya koymakta ve çeşitli stratejiler geliştirmektedir. İncelenen zaman serilerinin göstermiş olduğu niteliklere göre ortaya konan modellerin biri seçilir. Uygun olan modelin seçiminde genellikle eldeki verinin yapısı belirlemektedir. Öngörü modellerini kullananlar çeşitli model seçenekleri arasından uygun olanı seçme ve seçilen modelin kurulmasındaki her aşamada modelin incelenen probleme uygunluğunu denetleme gibi olanaklara sahiptir. O halde, öngörü modelleri kullanıldığı alanlar yönünden birçok kaynaklardan gelen veriler uygulanabilir. Bu yüzden sistem tanımlama modelleri son yıllarda büyük bir ilgi görmüş, ekonomistler, istatistikçiler, ve mühendisler tarafından oldukça sık kullanılmaktadır (Erdoğan vd. 2005). Öngörü modellerinin amacı en az sayıda parametre içeren uygun modeller elde etmektir. modelinin genel ifadesi Sistemde yapılan bazı deneyler

sonucunda elde edilen veriler kullanılarak model belirlenmeye çalışılır. Varsayılan model deneysel verilerle karşılaştırılarak model belirlenmeye çalışılır. Varsayılan modelden elde edilen veriler deneysel verilerle karşılaştırılarak uygunluğu test edilir

Parametrik modellerin belirlenmesi için Prediksiyon Hata Metotları (PEM- Prediction Error Methods) kullanılmaktadır. Bu metodun temel mantığı, modelden hesaplanan tepki ile ölçülen tepki arasındaki hatanın minimum olması ilkesine dayanmaktadır. Hatanın minimum olması, en basit anlamda hataların otokorelasyon katsayılarının belirli bir güven aralığında ve sınır değerleri içinde kalmalıdır. Otokorelasyon fonksiyonu , olası modellerin belirlenmesinde ve seçiminde kullanılan önemli analiz araçlarından biridir. Hataların sınır değerleri içinde kalması, seçilen modelin yapılacak simülasyon için uygun olduğunu gösterir. Otokorelasyon giriş ve çıkış değişkenleri arasındaki ilişkinin derecesini gösterir. Otokorelasyon belirli bir değer arasında olmalıdır bu yüzden otokorelasyon grafikleri çizilmelidir. Çizilen grafikte modelin sınırları belli bir band aralığı içinde kalıyorsa modelimiz sistem için uygundur. Kısmi otokorelasyon fonksiyonu geçici olarak belirlenen modelin giriş ve çıkış değişkenlerinin mertebesinin ne olması gerektiği ortaya koyar.

Zaman serileri ile sistemin beklenen değeri ve otokorelasyon fonksiyonu belirlenebilmektedir. Ayrıca, zaman serilerinin modellenmesi ile sistemin ileriye yönelik tahminleri yapılabilir. Etki-tepki büyüklükleri arasında gerçekleştirilecek bir çapraz-korelasyon işlemi ile de sistemin tepkisinin, etkiye göre gecikip-gecikmediği belirlenebilmektedir. Yani etkiyen büyüklük tepkiye sebep olmuş ise sistemde herhangi bir gecikme durumu meydana gelmemektedir. Eğer bir değişim söz konusu değilse sistemde gecikme meydana gelmemektedir.

3.2.3 Tahmine dayalı kontrol

Bu kısımda YSA model çıktısı tahmininde kullanılmaktadır.

3.2.3.1 YSA kullanarak nonlinear sistemin modellenmesi

Doğrusal olmayan prosesin tahmini için, YSA 'nın ağırlık vektörlerinin değerleri optimum değere erişince ye kadar eğitilmelidir. Bir çok uygulamada ileri beslemeli YSA kullanılmaktadır çünkü öğrenme algoritması daha az karmaşıktır. Bizim sistemimizde YSA'yı eğitmek için ARMAX modeli ve taşıyım olaylarına dayalı diferansiyel eşitlikler kullanılmıştır.

3.2.3.2 YSA bazlı tahmin

Sistem çıktısının gelecekteki tahmini için tahmin edici şarttır. Gelecek değerlerin tahmini için eğitilmiş ağda YSA girdi değerleri değiştirilerek ağın çıktısı elde edilmiştir. Bu çıktılar gelecekteki değerlerdir.

Sistem çıktısı:

$$y(t+1) = \sum_{j=1}^n w_j \sigma_j (w_j^u U(t-d) + w_j^y Y(t) + b_j) + b \quad (3.14)$$

eşitlik 3.14'de ki gibidir. Burada, n gizli katmandaki nöron sayısı, σ_j j. neronun aktivasyon fonksiyonu, w_j^u U(t-d-1) girdisinin j. Nöronun bilgisinin saklandığı ağırlık matrisi (sıra vektör), d ölü zaman, w_j^y ağırlık vektörü Y(t-1) girdisinin j. Nöronunun bilgisinin saklandığı ağırlık matrisi, b_j gizli tabakanın j. Nöronun bias değeri, w_j saklı tabakanın çıktı tabakasının ağırlığı ve b çıktının biası.

Eşitliği 3.14 genelleştirecek olursak;

$$y(t+i) = \sum_{j=1}^n w_j \sigma_j (w_j^u U(t-d+i-1) + w_j^y Y(t+i-1) + b_j) + b \quad (3.15)$$

eşitlik 3.15 elde edilir

burada

$$U(t-d+i-1)=[u(t-d+i-1),\dots,u(t-d+i-m)]^T \quad (3.16)$$

Eşitlik 3.16’de U ayar değişkenlerinin gösterimi

$$Y(t+i-1)=[y(t+i-1),\dots,y(t+i-n)]^T \quad (3.17)$$

Eşitlik 3.17’de sistem çıktısı verilmiştir.

YSA, tahmin edicisi kontrol sinyallerinin tahmini için tahmin kontrol algoritması ile uygulanır.

Tahmine dayalı kontrol stratejisinin uygulanması iki kısımdan oluşur.

- i) Gelecekteki değerlerin tahmini
- ii) Maliyet fonksiyonunun minimizasyonu

Maliyet fonksiyonu zaman zaman değişik formlarda bulunabilir, birçok durumda aşağıda verilen formülüzasyondan faydalanılır. Maliyet fonksiyonunun minimizasyonu girdi değeri için optimum kontrolü sağlar. Maliyet fonksiyonu eşitlik 3.18’de verilmiştir.

$$J = \sum_{N_1}^{N_2} [y(t+i) - y_m(t+i)]^2 + \xi \sum_{i=1}^{N_u} \Delta u^2(t+i-1) \quad (3.18)$$

$$1 \leq N_u < i \leq N_2 \quad (3.19)$$

Eşitlik 3.19’da N_u , N_2 ve ayar değişkeninin değişim aralığı verilmiştir. Burada N_u kontrol ufku, N_1 ve N_2 minimum ve maksimum tahmin ufku, i tahmin parametresi, y_m YSA çıktısı, y set noktası (referans değer), ξ ağırlık faktörü, Δ fark operatörü.

Minimum tahmin ufku (N_1) eğer zaman gecikmesi biliniyorsa zaman gecikmesine eşit alınır. Zaman gecikmesi hakkında hiçbir bilgi yoksa bir alınır. Minimum tahmin ufkunu gecikmeden küçük bir değer seçilmesi gereksiz hesaplamaya yol açar.

Maksimum tahmin ufku (N_2) sistem cevabı etkisi üzerinde etkisi fazla olmamakla beraber yeterince büyük seçilmelidir. Yaklaşık yirmi alınabilir, genelde pratikte 10 alınır.

Kontrol ufku basit prosesler için bir alınır. Daha karışık prosesler için (kararsız kökler varlığında) büyütmek gerekir yaklaşık iki veya üç alınır.

u 'nun büyüklüğü eşitlik 3.20'de verilmiştir.

$$u_{\min} \leq u(t + i) \leq u_{\max} \quad i=1,2,\dots, N_u \quad (3.20)$$

maliyet fonksiyonu ağırlık faktörü ($\xi=0$) genellikle sıfırdır. Büyüdükçe sistem tepkisi yavaşlar.

Maliyet fonksiyonun U 'ya göre türevi alınarak sıfıra eşitlenir, buradan u değerleri bulunur. Ancak prosesin ileri zamanlarında türev alma işlemi zorlaşır bu durumda yeni yaklaşımlar ile diferansiyel eşitlik çözülmeye çalışılır. Maliyet fonksiyonun minimizasyonu için Matlab'ın toolbox'ında hazır fonksiyonlar mevcuttur, bu fonksiyonlar kullanılarak optimizasyon problemi çözülür.

3.2.4 YSA geri beslemeli lineerizasyon kontrol

Bu kısımda nöron kontrol edici tanımlanmaya çalışılmıştır, nöron kontrolü iki isimle anılır bunlar : geri beslemeli linerizasyon (feedback linearization) ve NARMA-L2 kontrol. Sistemin modeli özel formdadır (arkadaş form). Linerizasyon nonlinear sistemlerin kontrollerine uygulanır. Sistemin asıl mantığı nonlinearliğin geri besleme ile giderilmesidir bu tür sistemlerde temel amaç doğrusal olmayan dinamik sistemin doğrusal dinamik sisteme dönüştürülmesidir. Bu kısımda arkadaş form sistem modeli ve YSA'nın bunu nasıl uyguladığı, sistemi nasıl tanımladığı anlatılacaktır.

NARMA-L2 kontrol edicideki ilk adım sistemin tanımlanıp kontrol edilmesidir. Genellikle uygulanan doğrusal olmayan standart model eşitlik 3.21'de verilmiştir (NARMA).

$$y(t + d) = F[y(t), y(t - 1), \dots, y(t - m + 1), u(t), u(t - 1), \dots, u(t - n + 1)] \quad (3.21)$$

burada $u(t)$ sistem girdisi, $y(t)$ sistem çıktısı ve d gecikme süresi, m kaç tane sistem çıktısı bulunduğunu, n kaç tane sistem girdisi bulunduğunu gösterir. YSA'nın kullandığı doğrusal olmayan dinamik model aşağıda ki gibi (eşitlik 3.22) tanımlanmıştır.

$$\hat{y}(t+d) = N[y(t), y(t-1), \dots, y(t-m+1), u(t), u(t-1), \dots, u(t-n+1)] \quad (3.22)$$

Burada, $N[.]$ yapısı tek tabakalı YSA'yı temsil etmektedir, $\hat{y}(t+d)$ tahmin edilen $y(t+d)$ sistem çıktısı. Sistem çıktısı ile referans değer (set noktası) arasındaki hatayı kullanarak YSA ağırlıkları ayarlanmaktadır. Modelin Kontrol edilmesi için sistem çıktısı ile referans değerinin aynı olması istenir ($y(t+d)=y_r(t+d)$ burada ki d gecikmedir), bu yüzden bir sonraki adımda kontrol edici tanımlanmalıdır ($u(t)$). Kontrol edici $u(t)$ (ayar değişkeni) eşitlik 3.23'deki gibi ifade edilir. Eşitlik 3.21'de $F[.]$ u ile değişmiyorsa kontrol sağlanamaz bu yüzden $\partial F / \partial u(t) \neq 0$ olmalıdır.

$$u(t) = G[y(t), y(t-1), \dots, y(t-m+1), y_r(t+d), u(t-1), \dots, u(t-n+1)] \quad (3.23)$$

Kontrol probleminin neticesi olarak G ölçülmüş değer girdi ve çıktısın, verilen referans değerinden $y_r(t+d)$ faydalanılarak kararlaştırılmıştır(veya tahmin) edilmiştir. Doğrusal olmayan Dinamik sistemin geri beslemeli kontrol edicisi için YSA'nın ayarlanan parametreleri (ağırlıkları) ayarlamak için sabit geri beslemeli yöntemler kullanılamaz (Levenberg-marquardt, geri yayılım algoritması). Bu doğrusal olmayan dinamik sistemleri eğitmek için dinamik geri beslemeli algoritmalar kullanılmalıdır buda çok ağır hesap yükü getirmektedir, buna Narendra ve Mukhopadhyay tarafından sistemin yaklaşık modeli kullanılarak bir çözüm bulunmuştur (doğrusallaştırılmıştır).

Bu nedenle kontrol edici yaklaşık olarak NARMA-L2 modelini baz alır. $F[.]$ 'nin Taylor serisini kullanarak $u(t)$ etrafında yazılırsa, bu model eşitlik 3.24'deki gibi olur. Eşitlik 3.24'de $u(t)$ kontrol girdisi, f ve g nonlinear olmayan arkadaş fonksiyonlardır.

$$y(t+d) = f[y(t), \dots, y(t-m+1), u(t-1), \dots, u(t-n+1)] + g[y(t), \dots, y(t-m+1), u(t-1), \dots, u(t-n+1)] * u(t) \quad (3.24)$$

Bu formun avantajı kontrol girdisini düzenleyerek sistem çıktısı ile eşitler ($y(t+d)=y_r(t+d)$).

$$u(t) = \frac{y_r(t+d) - f[y(t), \dots, y(t-m+1), u(t-1), \dots, u(t-n+1)]}{g[y(t), \dots, y(t-m+1), u(t-1), \dots, u(t-n+1)]} \quad (3.25)$$

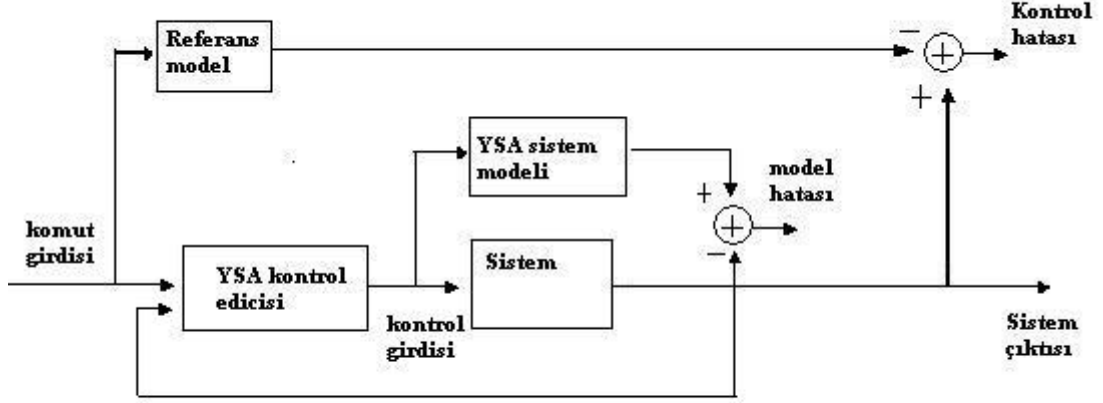
Eşitlik 3.24 ayar değişkeni $y(t+d)=y_r(t+d)$ koşulunu sağlayacak şekilde düzenlenecek olursa eşitlik 3.25 elde edilir; amaç t anında ki $u(t)$ değil $t+1$ anındaki u 'yu bulmaktır. bu eşitliği direk kullanmak uygulanabilirlik problemine neden olur bu yüzden kontrol girdisinin hesabı için eşitlik 3.24 $u(t+1)$ için yazılırsa eşitlik 3.26 elde edilir. Bu eşitliği $t+1$ için kontrol edici çıktısı hesabı eşitlik 3.27 gibi olur.

$$y(t+d) = f[y(t), \dots, y(t-m+1), u(t), \dots, u(t-n+1)] + g[y(t), \dots, y(t-m+1), u(t), \dots, u(t-n+1)] * u(t+1) \quad (3.26)$$

$$u(t+1) = \frac{y_r(t+d) - f[y(t), \dots, y(t-m+1), u(t), \dots, u(t-n+1)]}{g[y(t), \dots, y(t-m+1), u(t), \dots, u(t-n+1)]} \quad (3.27)$$

Burada $d \geq 2$ olmalıdır. Sistemin blok diyagramı şekil 4.30'daki gibidir.

3.2.5 Model referanslı kontrol



Şekil 3.13 Model referanslı kontrol şeması

YSA ile model referanslı kontrol yapısında iki tane ayrı YSA kullanılmıştır: Biri kontrol edicisi YSA ile tanımlanmış, diğeri YSA ile sistem modellenmesidir. YSA ile sistem modellenmesinde ilk olarak sistem tanımlanmaya çalışılmış, sistem girdisine karşı elde edilen çıktı ile YSA sistem modeli çıktısı arasındaki hatayı (model hatası) kullanarak YSA sistem modeli eğitilmiş ve bu şekilde sistem davranışları modellenmiştir. Sistemin çıktısının istenilen set noktasında olması için, sistem çıktısı ile referans modeli arasındaki hatadan (kontrol hatası) faydalanılarak kontrol edici eğitilmiştir. Burada kontrol edici girdisinde üç tane değişken ayarlanabilir bunlar: Referans girdileri gecikmesi, kontrol edici çıktıları gecikmesi; sistem çıktı gecikmesidir. YSA sistem modelinde ise iki tane değişken ayarlanabilir bunlar: kontrol edicisi çıktısı gecikmesi, sistem çıktısı gecikmesidir. Sistemin birinci mertebe iletim fonksiyonuna uyduğu kabul edilmiş ve referans modeli ARMAX ve taşıyım olayları ile ilgili eşitlikleri denklemleri için ayrı ayrı eşitlik 3.28 ve 3.29 gibi önerilmiştir. Model referanslı kontrolün blok diyagramı şekil 3.13'teki gibidir.

$$q(z) = \frac{\frac{0.42}{5.3}}{1 - \exp(-\frac{0.42}{5.3})z^{-1}} \quad (3.28)$$

$$\frac{\partial y}{\partial t} = \frac{0.42}{5.3} e^{-\frac{t}{5.3}} \quad (3.29)$$

önerilen referans modeller birinci derece referans modellerdir. Zaman sabiti (τ) ve kazanç (K) değerleri deneysel olarak elde edilmiş olup literatürden alınmıştır.

3.2.6 Karıştırmalı biyoreaktörde Oksijen kütle aktarım katsayısının bulunması

Biyoreaktörlerin tasarımında ve işletme koşullarının belirlenmesinde oksijen aktarımı çok önemli bir parametredir. Oksijen aktarımı bir çok parametreden tarafından etkilenir (biyoreaktör şekli ve tipi, ortam bileşimi, ortam derişimi, ortamdaki mikroorganizmanın yapısı, gaz dağıtıcı tipi, sıcaklık, vizkozite vb). Bu çalışmanın bu kısmında literatürden elde edilmiş oksijen için kütle aktarımı katsayısını hesaplayan bağıntıyı kullanarak, kütle aktarım katsayısını tahmin etmek ve bu tahmin edilen kütle aktarım katsayılarını, denklemden elde edilen kütle aktarım katsayısıyla karşılaştırıp YSA kütle aktarım katsayısında tahminde kullanılabilirliğini göstermektir.

Mikroorganizmalar fiziksel ve kimyasal çevrelere karşı oldukça hassastırlar. Metabolik faaliyetlerindeki farklılık nedeniyle farklı ürünler ürettikleri bilinmektedir. Bu farklılığa mikroorganizmaların genetik ve/veya çevresel koşullarındaki değişiklikler neden olmaktadır. Hatta aynı mikroorganizmadan farklı ürünler üretilmesi daha söz konusudur. Bu çalışmada kullanılan *S. Cerevisiae* mikroorganizması etanol, ekmek mayası, gliserol ve invertaz enzimi üretiminde kullanılmaktadır. Bu mikroorganizma havalı koşullarda çoğaldığında maya hücreleri, havasız koşullarda çoğaldığında ise etanol üretilir. Çoğalma ortamında glukoz miktarı inhibisyon etkisi yaratmayacak kadar olsa da çözünmüş oksijen derişimi az ise *S. Cerevisiae* etanol üretir. Besi ortamında çözünmüş oksijen derişimi, kritik değerin altına düştüğünde (doğunluğun %5-10) *S.cerevisiae* mikroorganizması etanol üretmeye başlar. Etanol mayalar için inhibe edici

özelliğindedir. Genel olarak ortamda %2 olursa maya çoğalması yavaşlar,%10 olduğunda durur (Boyacıoğlu 2002).

Literatürden alınan (Galaction *et al.* 2004) formülün oluşturulması için şu deneysel şartlar sağlanmıştır. 5 L hacimli biyoreaktör (4L'si kullanılmış, elipsiyodal dipli), veriler bilgisayar tarafından kaydedilmiştir. Biyoreaktör karıştırıcısı 2 türbinli ve 3 engellidir. Karıştırma hızları 0 ve 700 rpm aralığındadır (güç katsayısı 50-200 W m⁻³ tür). Dağıtıcı sistemi tek yüzük dağıtıcısı 64 mm çapındadır. Mikroorganizma kuru ağırlığı 33.5-150 g L⁻¹ dir. Ortamdaki oksijen miktarı InPro 6000 serisi tipi ölçücü tarafından ölçülmüştür. Sistem koşullarının tanımlanması için her deney üç veya dört kez tekrarlanmış, oksijen kütle aktarım katsayısının ortalama değeri alınmıştır. Maksimum deneysel hata %± 3.63 bulunmuştur. Çözelti ortamında karboksil metil selülozu vardır. Yapılan deney sonucunda kütle aktarım katsayısı eşitlik 3.30 gibi bulunmuştur.

$$k_l a(s^{-1}) = 52.44 \frac{v_s^{0.514}}{C_x^{0.702} \left(\frac{P_a}{v} \right)^{0.0762}} \quad (3.30)$$

deneysel sonuçlarla maksimum sapma %± 8.4'tür

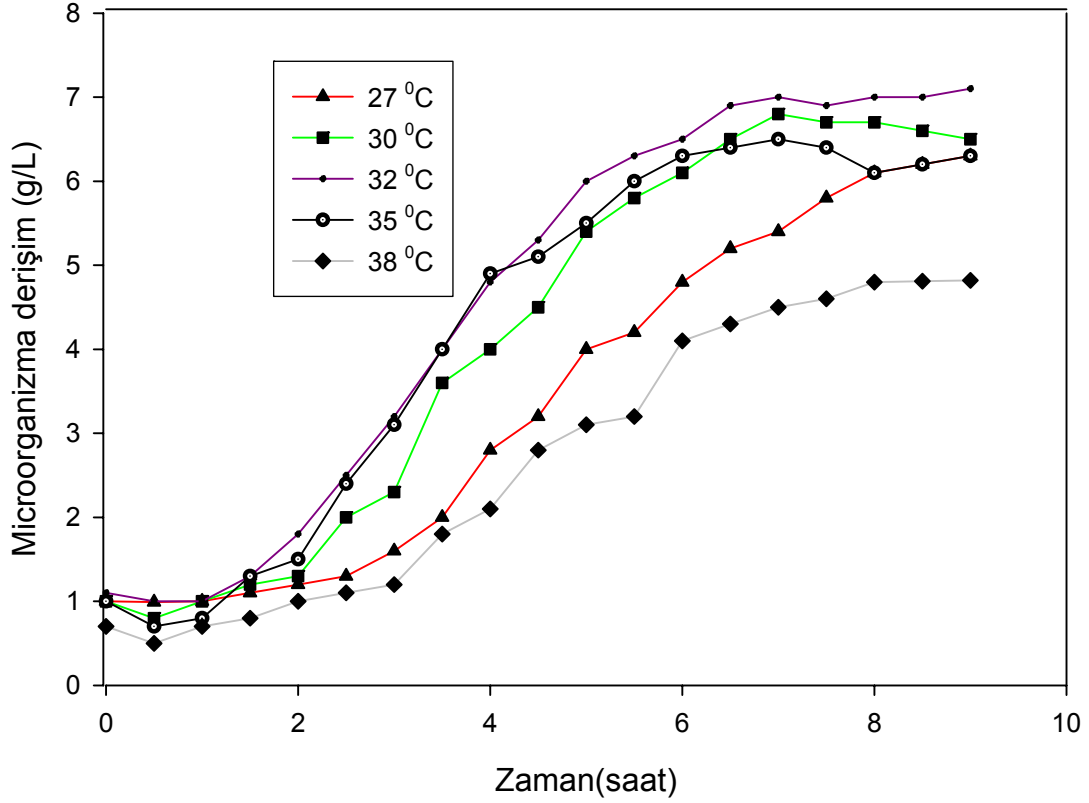
4. BULGULAR

4.1 Yapay Sinir Ağı ile Reaksiyon Hızı Tahmini

Daha önceki yapılan çalışmalarda *S.cerevisiae* mikroorganizmasının üretiminde sıcaklık, pH ve başlangıç mikroorganizma derişiminin etkili işletim parametreleri olduğu görülmüştür. Özellikle mikroorganizma çoğalmasına etki eden parametre sıcaklık ve pH olup, en uygun değerlerinde prosesin işletilmesi ve kontrol edilmesi ürün verimliliği açısından oldukça önemlidir. Bu nedenle yapılan deneysel çalışmalardan elde edilen deneysel veriler bu çalışmada kullanılarak *S.cerevisiae* mikroorganizmasının reaksiyon hızı, YSA ile tahmin edilmeye çalışılmıştır. Reaksiyon hız tahmini için iki ayrı YSA kullanılmıştır. Birinci ağ farklı pH değerlerinde, ikinci ağ farklı sıcaklık değerlerinde reaksiyon hızının bulunmasında kullanılmıştır. Birinci ağda giriş değişkenleri: maya başlangıç derişimi, maya derişimi ve sıcaklık değerleri ağ çıktısı ise reaksiyon hız ifadesidir, ikinci ağda ise giriş değişkenleri pH, maya başlangıç derişimi ve maya derişimidir. Ağ çıktısı ise reaksiyon hız ifadesidir.

4.1.1 Sıcaklığın mikroorganizma çoğalmasına etkisi ve deneysel veriler

Akay 1998 yılında yapmış olduğu çalışmada *S.cerevisiae* mikroorganizmasının maksimum hücre derişiminde üremesi için gerekli olan reaktör sıcaklığını bulmak maksadıyla beş farklı sıcaklık ve sabit pH'ta deneyler yapmıştır. Elde ettiği deneysel veriler şekil 4.1 de verilmiştir. Böylece her sıcaklık için mikroorganizma derişim-zaman grafiği çizilmiştir. Şekilden de görüldüğü gibi araştırmacı çalışmasında en yüksek mikroorganizma derişimine ulaştığı sıcaklığı 32°C bulmuştur. Bu sıcaklıklarda ki deneysel veriler kullanılarak hesaplanan reaksiyon hızları, YSA eğitiminde ve test edilmesi aşamasında kullanılmıştır.

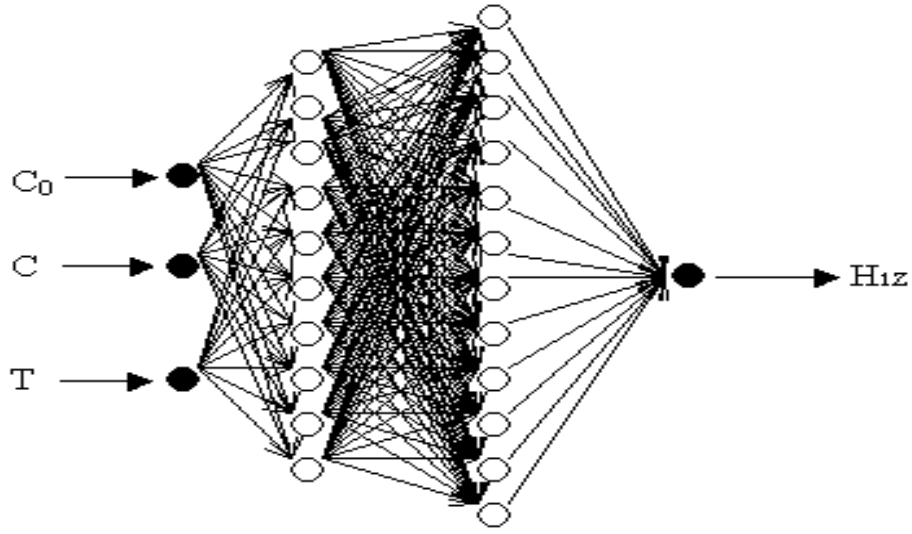


Şekil 4.1 Farklı sıcaklıklarda zamanla mikroorganizma derişiminin değışimi (Akay 1998)

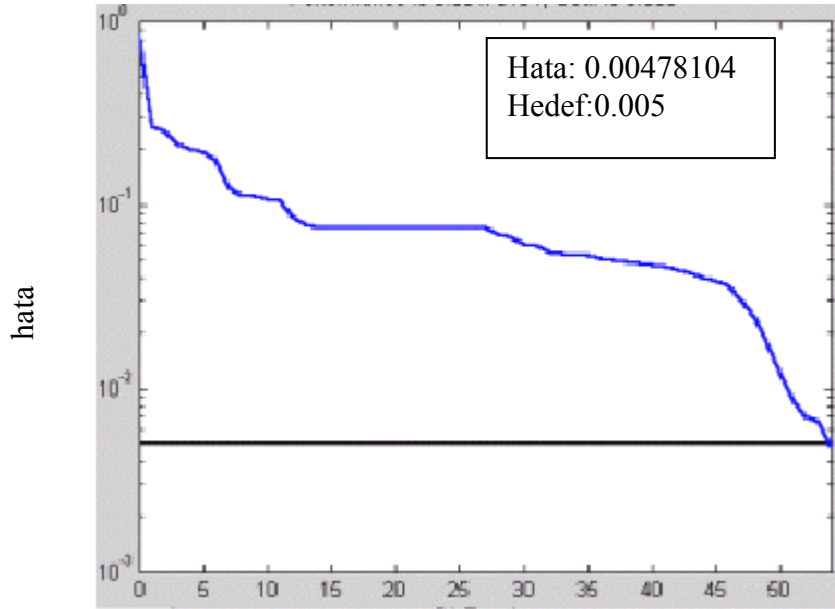
4.1.2 Farklı sıcaklıklarda yapay sinir ağıları ile reaksiyon hızı tahmini

Çalışmanın bu kısmında YSA kullanılarak reaksiyon hız verileri deneysel verileri kullanarak monod kinetiğine baş vurulmadan tahmin edilmeye çalışılmıştır. Bu çalışmada Matlab 7.0'a ait neural network yapay sinir ağlarının eğitimi ve hesaplamasında kullanılmıştır. Eğitimin tamamlanması iki durumdan birinin gerçekleşmesi ile belirlenir. Bu iki durum: iterasyon sayısı kriteri ve hatasının belli bir değerinin altına düşmesi kriteridir. Bu durum Matlab'ta yazılan kodlar ile belirtilmelidir. Ham veriler Excel formatında okutulularak matlab ortamına aktarılması sağlanmıştır.

Eđitim iin 81 veri test iin ise 9 veri seilmiřtir. Deneysel veri ile YSA karřılařtırılması izelge 4.1’de verilmiř, deneysel ve teorik deęerin birbirlerine yakın olduęu grlmřtir. Aęın yapısı řekil 4.2’de grlmektedir. Uygulanan aęın yapısı 10-12-1’dir. Aęda  giriř deęiřkeni bir ıkıř deęiřkeni vardır giriř deęiřkenleri; bařlangı deriřimi, maya deriřimi ve sıcaklık verileri ve ıkıř deęiřkeni ise mikroorganizma oęalma hızıdır. Birinci katmanda 10 hcre kullanılmıřtır, buradaki transfer fonksiyonu ilk katmanda doęrusal ikinci ve nc katmandaki transfer fonksiyonu hiberbolik tanjant’tır. İterasyon sayısı en fazla 2000 deęerine ayarlanmıř, aę hata deęeri olarak 0.005 deęeri atanmıř ve 54 iterasyonda belirlenen hata deęerine ulařılmıřtır (řekil 4.3). Deneysel verilere karřı YSA tarafından bulunan hız verileri grafięe geirilerek řekil 4.4 elde edilmiřtir. Burada R^2 ’nin bire yakın olduęu grlmřtir. Aę ıktısından bulunan hız ile literatrden alınan hız verileri izelge 4.1 ve izelge 4.2’de verilmiř olup deneysel hız ile YSA tarafından hesaplanan hızın yakın olduęu sonucuna varılmıřtır. alıřmada kullanılan deneysel veriler Ek 8’de verilmiřtir.



Şekil 4.2 YSA şematik gösterimi

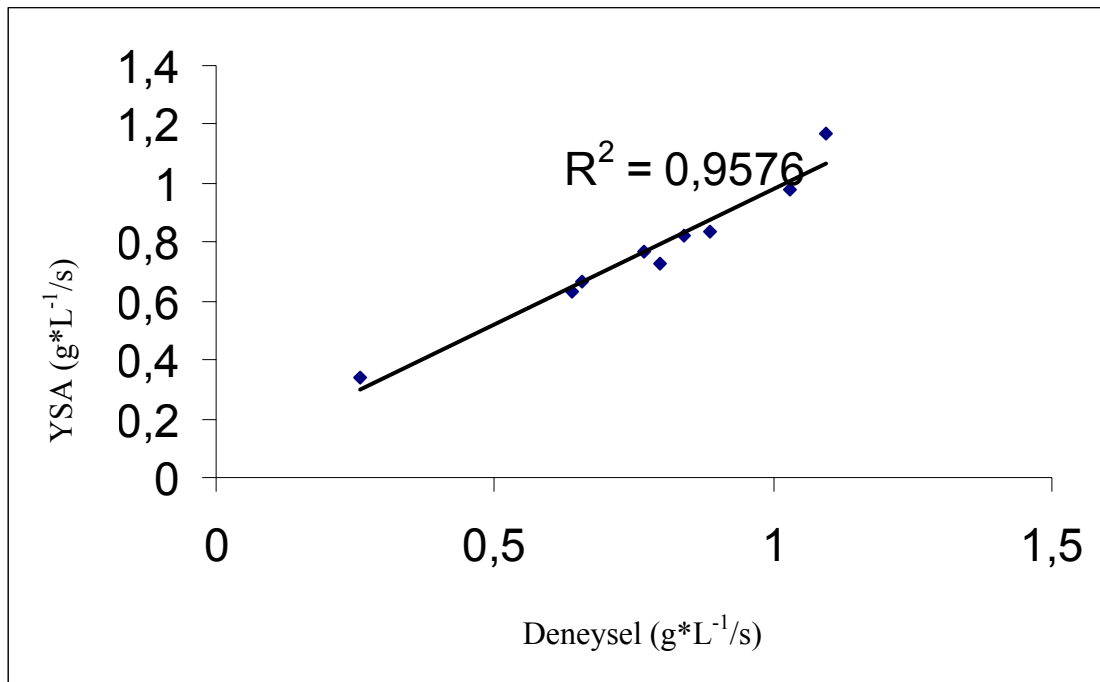


54 epochs

Şekil 4.3 10-12-1 nörona sahip YSA için eğitimdeki hata

Çizelge 4.1 Deneysel hız ile YSA tarafından tahmin edilen hızın karşılaştırılması

Deneysel ($\text{g}\cdot\text{L}^{-1}/\text{s}$)	YSA ($\text{g}\cdot\text{L}^{-1}/\text{s}$)	Mutlak fark
0.6562	0.6651	0.009
0.7968	0.7296	0.0672
0.6372	0.6347	0.0025
1.0288	0.9755	0.0533
0.2575	0.3368	0.0793
0.8846	0.8357	0.0489
1.0950	1.166	0.071
0.8409	0.8211	0.0198
0.7691	0.7649	0.0042



Şekil 4.4 Deneysel hıza karşı YSA tarafından hesaplanan hız verileri

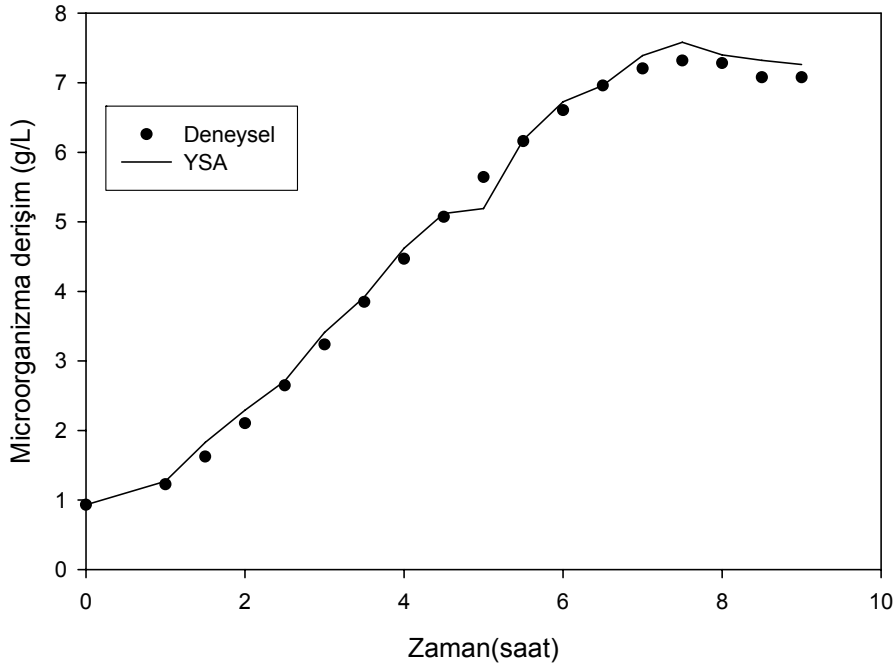
YSA ile hesaplanan reaksiyon hızlarını kullanarak mikroorganizma derişim değerlerine geçilerek, deneysel mikroorganizma değerleri yerine kullanılıp kullanılmayacağı test edilmiştir. Buna göre mikroorganizma derişiminin zamanla değişimi kesikli bir biyoreaktörde eşitlik 4.1 ile ifade edilir.

$$r = \frac{dC}{dt} \quad (4.1)$$

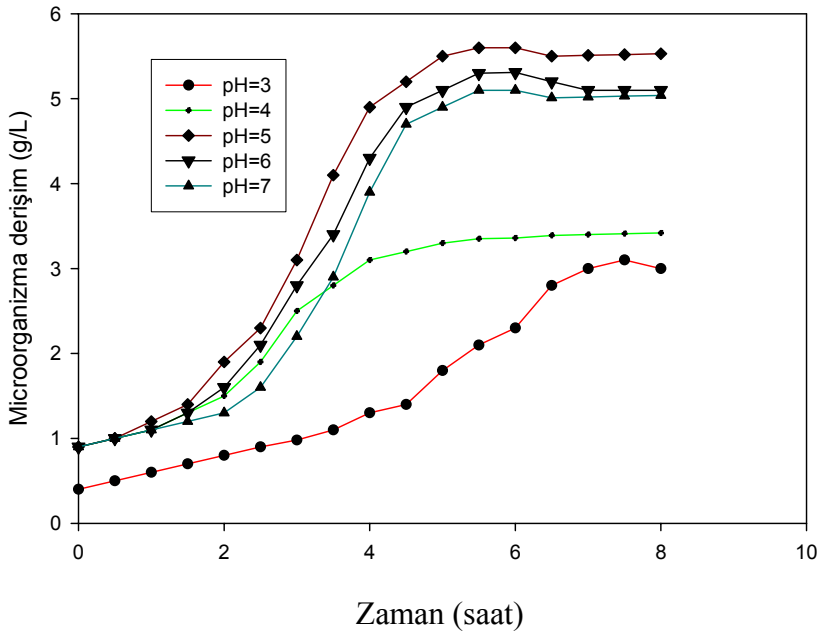
Buradaki r reaksiyon hızıdır.

$$C_{i+1} = C_i + r_i * \Delta t \quad (4.2)$$

Fark yaklaşımına göre eşitlik 4.1 düzenlenip eşitlik 4.2 elde edilmiştir. Tahmin edilen hız kullanılarak eşitlik 4.2'den elde edilen derişimler deneysel derişimlere karşı grafiğe geçirilerilmiştir. Deneysel ve teorik verilerin karşılaştırılması 32°C'da şekil 4.5'de gösterilmiştir. Şekil 4.5'den de anlaşılacağı gibi YSA tarafından hesaplanan maya derişimi ile deneysel maya derişiminin birbirine yakın olduğu görülmüştür.



Şekil 4.5 32 °C’de deneysel derişimle YSA tarafından hesaplanan hızlardan elde edilen derişim



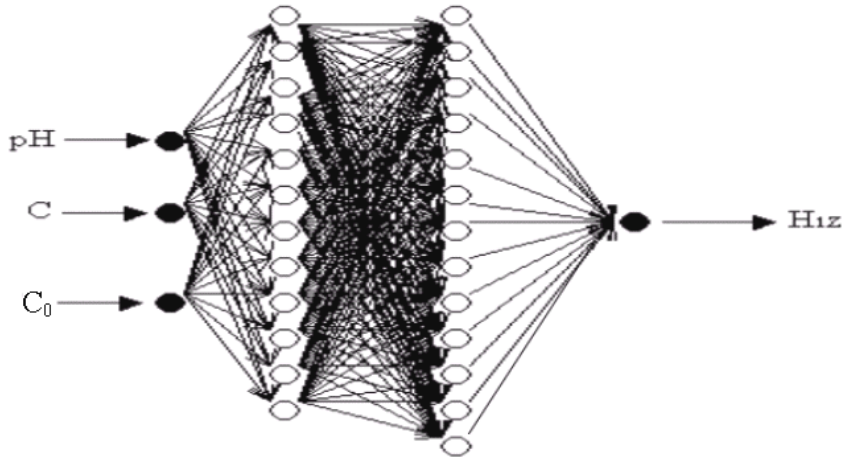
Şekil 4.6 Çoğalma ortamı pH değerinin mikroorganizmanın çoğalmasına etkisi (Akay 1998)

4.1.3 pH'ın mikroorganizma çoğalmasına etkisi ve deneysel veriler

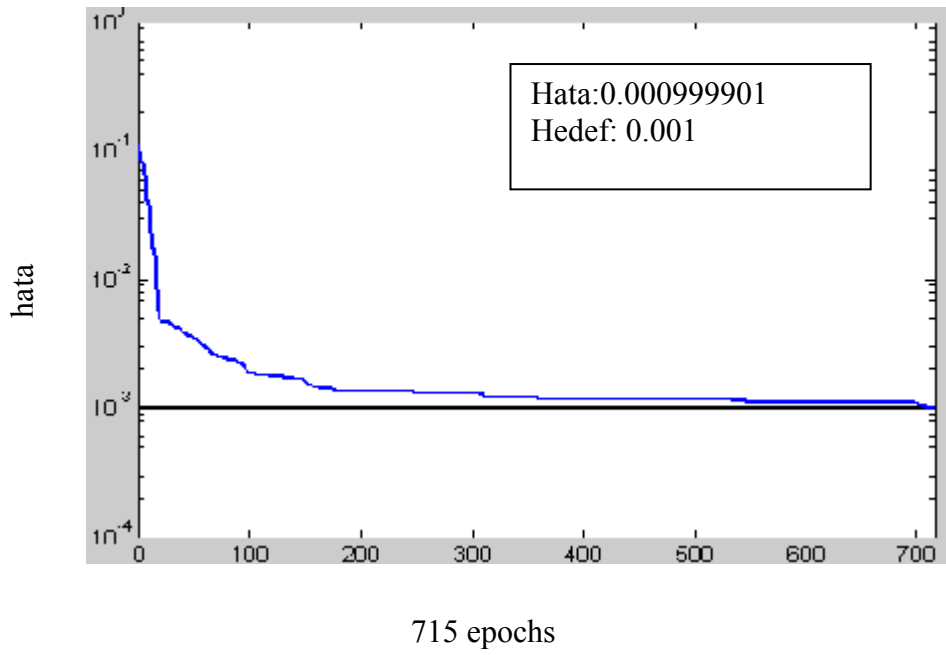
Akay 1998, yapmış olduğu çalışmaya göre 3, 4, 5, 6, 7 başlangıç pH değerinde 32°C'da 12 saat süre ile deneyler yapmış, deneysel veriler şekil 4.6'da gösterildiği gibidir. Bu çalışmada veriler YSA ile işlenmiştir.

4.1.4 Farklı pH değerlerinde yapay sinir ağları ile reaksiyon hızı tahmini

Teorik hesaplama için farklı pH değerlerinde elde edilen deneysel derişim verileri zamana karşı grafiğe geçirilip, elde edilen grafikten bir fonksiyon geçirilmiştir. Bu fonksiyonun zaman karşı türevi alınması ile mikroorganizma üreme hızı verileri bulunmuştur. Yapay sinir ağının eğitim ve test verileri ağa tanıtılmıştır. Eğitim için 70 veri test için ise 10 veri seçilmiştir. Deneysel hız ile YSA tarafından hesaplanan hızın karşılaştırılması çizelge 4.2'de verilmiş ve birbirlerine yakın olduğu görülmüştür. Ağın yapısı şekil 4.7'de görülmektedir. Uygulanan ağın yapısı 12-13-1'dir. Ağda üç giriş değişkeni bir çıkış değişkeni vardır bunlar; mayanın başlangıç derişimi, maya derişimi ve pH verileri ağa verilmiş ve çıkış değişkeni ise mikroorganizma çoğalma hızıdır. Birinci katmanda 12 hücre kullanılmıştır, buradaki transfer fonksiyonu ilk katmanda doğrusal ikinci ve üçüncü katmanda transfer fonksiyonu hiberbolik tanjant'tır. İterasyon sayısı en fazla 2000 değerine ayarlanmış, ağ hata değeri olarak 0.001 değeri atanmış ve 715 iterasyonda belirlenen hata değerine ulaşılmıştır (şekil 4.8). Deneysel verilere karşı YSA tarafından bulunan hız verileri grafiğe geçirilerek şekil 4.9 elde edilmiştir. R^2 'nin bire yakın olduğu görülmüştür. Ağ çıktısından bulunan hız ile literatürden alınan deneysel hız verileri şekil 4.9'da karşılaştırılmıştır. Çalışmada kullanılan veriler Ek 9'da verilmiştir.



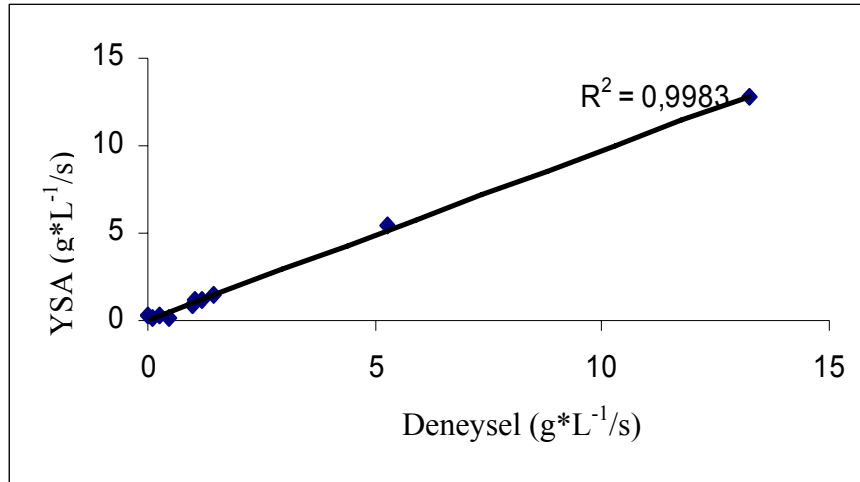
Şekil 4.7 YSA şematik gösterimi



Şekil 4.8 Eğitilen 12-13-1 nörona sahip YSA için eğitimdeki hata

Çizelge 4.2 Deneysel hız ile YSA tarafından bulunan hızın karşılaştırılması

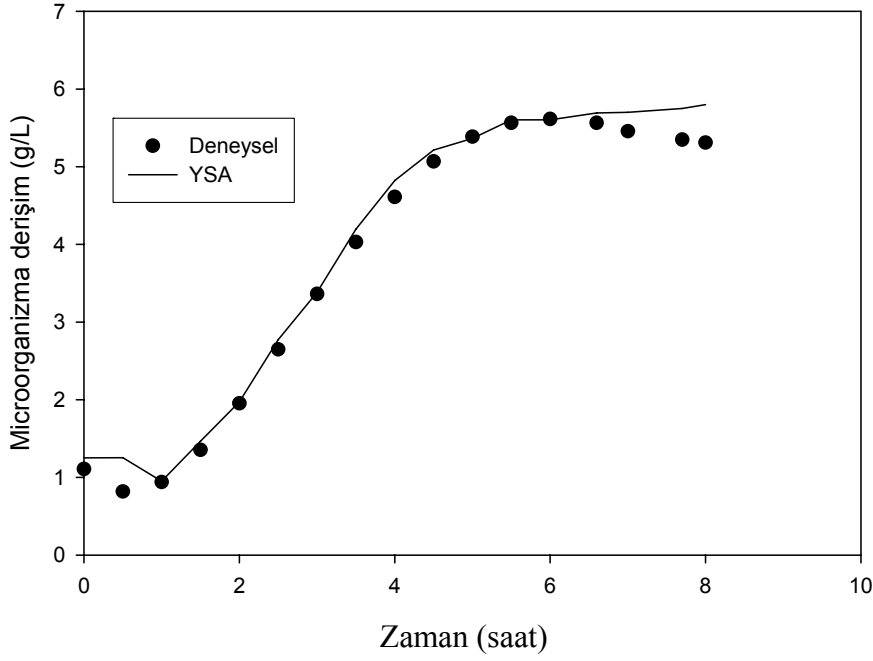
Deneysel ($\text{g}\cdot\text{L}^{-1}/\text{s}$)	YSA ($\text{g}\cdot\text{L}^{-1}/\text{s}$)	Mutlak fark
0.2329	0.3282	0.0953
5.282	5.3704	0.092
13.22	12.7532	0.4668
0	0.2444	0.2447
1.43	1.4377	0.0077
0.49	0.2049	0.2851
1.03	1.1384	0.1084
0.0844	0.2183	0.1339
0.9813	0.8433	0.138
1.199	1.1476	0.0514



Şekil 4.9 Deneysel hıza karşı YSA tarafından tahmin edilen hız

Farklı sıcaklık ve sabit pH verileri kullanılarak YSA ile bulunan reaksiyon hızları ve bundan da teorik derişim değeriine geçme mantığı bu kez, farklı pH ve sabit sıcaklık deneysel verileri kullanılarak teorik derişim değeriine geçirilmiştir. Elde edilen sonuçlar şekil 4.10’da gösterildiği gibidir.

Deneysel mikroorganizma derişimi veriler ile eşitlik 4.2'den hesaplanan derişimlerin birbirlerine yakın olduđu görölmektedir. Buradan da anlaşılacağı gibi YSA reaksiyon hızı hesaplanmasında kullanılabilir bir yöntemdir



Şekil 4.10 pH=5 için deneysel derişim ile YSA tarafından bulunan hızdan elde edilen derişimin karşılaştırılması

Şekil 4.5 ile şekil 4.10 karşılaştırıldığında her iki şeklin aynı olması gerekirken az fark olmasının sebebi her iki YSA algoritmasında kullanılan pH 5 ve sıcaklık 32°C için verilerin farklı olmasından kaynaklanmaktadır.

4.2 *Saccharomyces Cerevisiae* Mikroorganizmasının YSA ile Kontrolü

YSA dinamik sistemlerin kontrolüne başarılı şekilde uygulanabilmektedir. *Saccharomyces cerevisiae* mikroorganizması ekmek yapımında kullanılan bir mayadır. Çoğaltılması sırasında kullanılan oksijenin belirli miktarda olması gereklidir. Bu mikroorganizma havalı koşullarda çoğaldığında maya hücreleri, havasız koşullarda ise etanol üretir. Bu yüzden ortamda bulunan oksijen belirli bir değerin altına (1 mg/l) düşmemesi gerekmektedir. Kontrol problemleri için üç tane kontrol algoritması kullanılmıştır. Bu çalışmada YSA kullanılarak *Saccharomyces cerevisiae* mikroorganizması için oksijen kontrolünün sağlanmasına çalışılacaktır. Kontrol probleminin temel amacı oksijen derişimin istenilen değerler arasında kalmasını sağlamaktır. Ayar değişkeni oksijenin akış hızı seçilmiş, kontrol değişkeni oksijen derişimidir. Ortama gönderilen hava miktarının fazla olması pompalama maliyetini artırır.

Fermantasyon sistemlerinin kontrolü sistemin modelinin doğru tanımlanması, doğrusal olmayan ve zaman değişkenli sistemlerde prosesin yavaş yanıt vermesi, güvenilir sensör sistemleri ölçümleri gibi parametrelerin varlığından dolayı kolay değildir.

Proses iki ayrı modelleme yöntemiyle tanımlanmıştır.

a) Taşınım olaylarına dayalı diferansiyel model (governing equations)

b) Polinom tipi (ARMAX) modelle

a) Taşınım olaylarına dayalı modeller (governing equations): Taşınım olayları ile elde edilen eşitlikte biyokütle (x), substrate (s), çözünmüş oksijen (c_0) mol kesri, tam karışmalı biyoreaktörde, taşınım olayları eşitlikleri ile mayanın üstsel üreme faz bölgesi için elde edilmiş olup bu eşitlikler aşağıda verilmiştir (Cardello and San, 1987).

$$\frac{dx}{dt} = \mu x \quad (4.4)$$

$$\frac{ds}{dt} = -\mu \frac{x}{Y_{sx}} \quad (4.5)$$

eşitlik 4.4 biyokütlenin zamanla değişimini, eşitlik 4.5 substrate değişimini, eşitlik 4.6 çözünmüş oksijen miktarının zamanla değişimini, eşitlik 4.7 oksijen mol kesrinin zamanla değişimini, eşitlik 4.8 oksijen tüketim hızını (OUR), eşitlik 4.9 kütle aktarım katsayısının değişimi ifade eden denklemdir.

$$\frac{d(DO)}{dt} = k_L a [Hc_0 - DO] - OUR \quad (4.6)$$

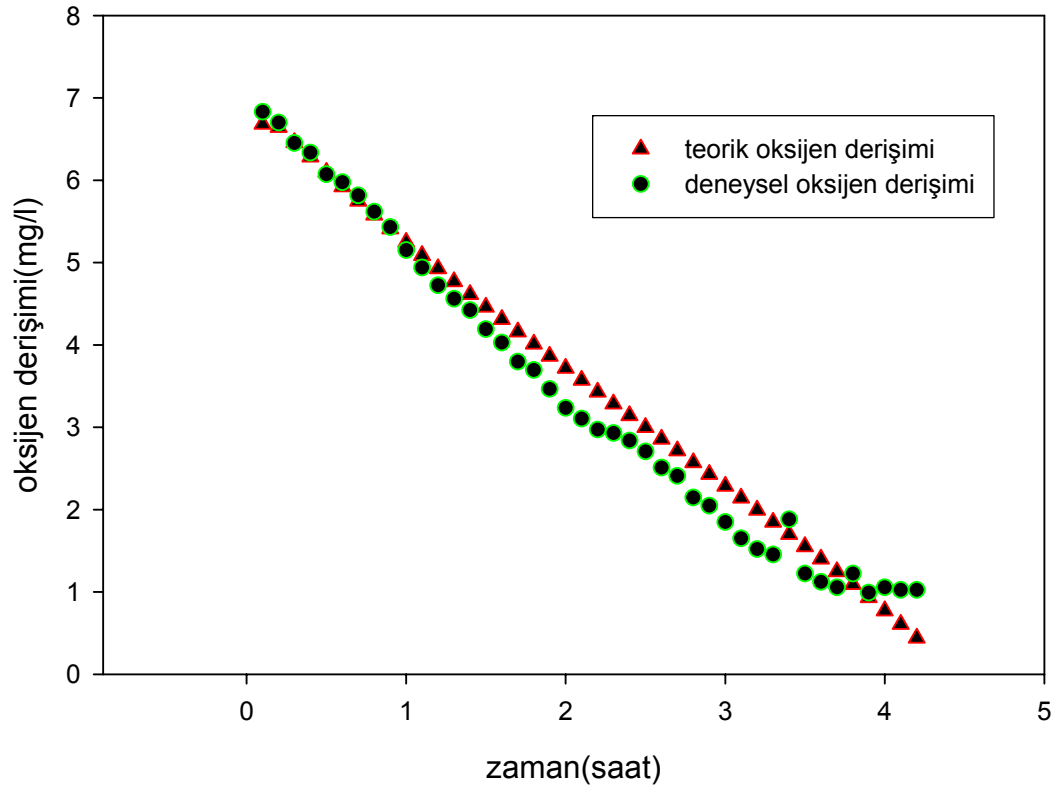
$$\frac{dc_0}{dt} = \frac{RT}{\varepsilon p M} \left[-k_L a [Hc_0 - DO] + \frac{pMQ[y_{in} - c_0]}{RTV} \right] \quad (4.7)$$

Burada;

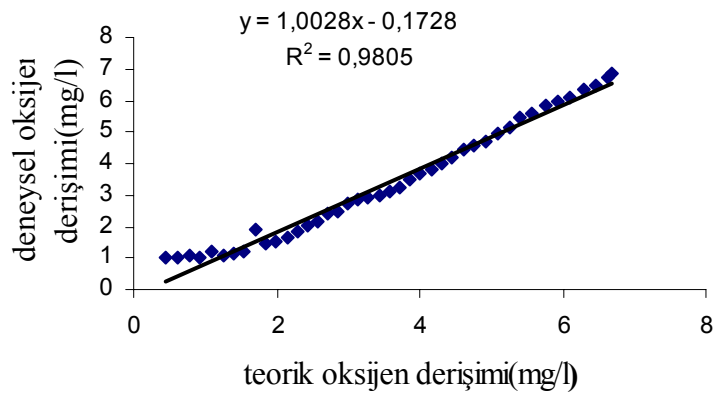
$$OUR = \frac{\mu X}{Y_{o_2}} \quad (4.8)$$

$$k_L a = \frac{OUR}{[Hc_0 - DO_{set}]} \text{ 'dir} \quad (4.9)$$

ε reaktördeki boşluk kesri değeri ise 0.15'dir. M oksijenin molekül ağırlı, H Henry sabiti (0.03488 g/l), p basınç (1 atm), μ spesifik büyüme hızı (0.5 l/saat) dir. Kullanılan denklemlerin sonuçları deneysel verilerle kontrol edilip bu denklemlerin kullanıma uygunluğu araştırılmış, karşılaştırılma sonucu Şekil 4.11 ve şekil 4.12'de gösterilmiştir. Bu şekillerden de anlaşılacağı gibi deneysel ve taşınım olaylarına dayalı modelden elde edilen verilerin birbirine yaklaşık dört saat boyunca yakın olduğu görülmüştür. Mayanın üstsöl üreme fazı için kullanılabileceğı görölmektedir. Polinom tipi model, sistem işletim şartlarının değışmemesi koşulu ile mayanın tüm üreme fazlarını içermektedir. Taşınım olaylarına dayalı diferansiyel modelde ise yaklaşık dört saat boyunca etkili iken, polinom tipi modelde sistemin işletim koşulları değışmediğı sürece zaman sınırlaması olmaksızın mayanın tüm fazları için kullanılabilir. Bu yüzden polinom tipi modelle modellenme yoluna gidilmiştir.

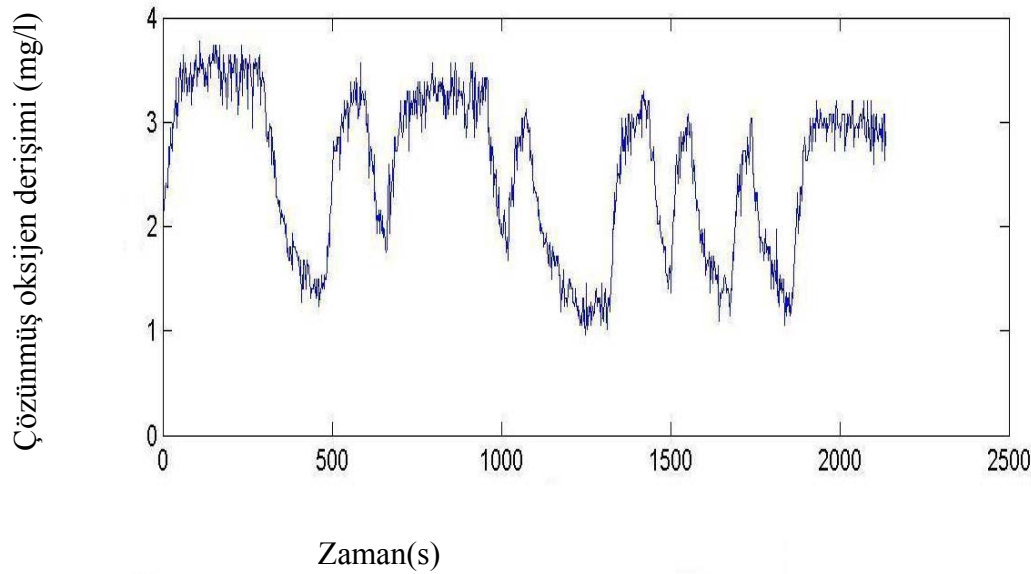


řekil 4.11 Deneysel ve teorik oksijen deriřiminin karřılařtırılması

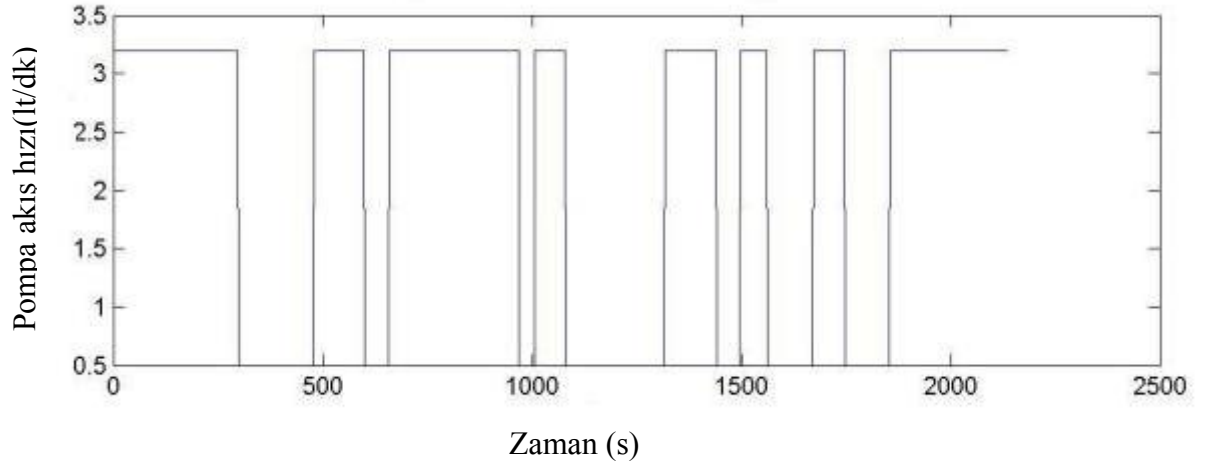


řekil 4.12 Deneysel oksijen deriřimine karřı teorik oksijen deriřimi

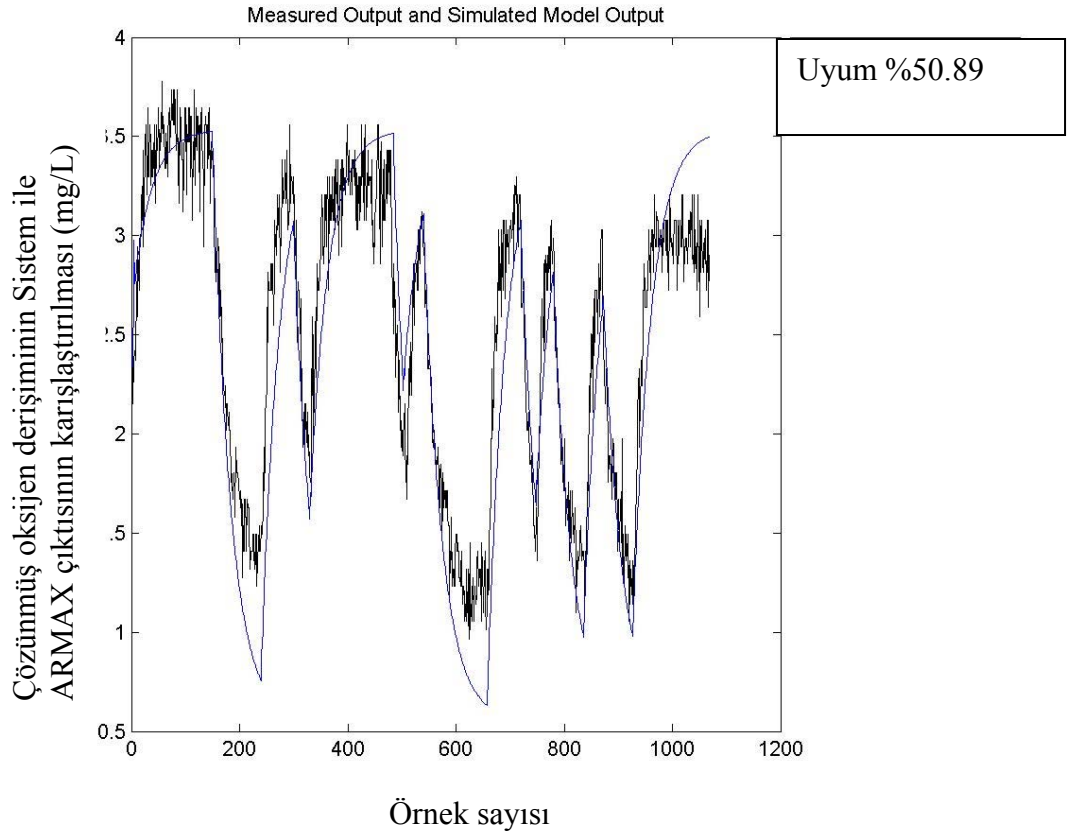
b) polinom tipi modeller (ARMAX): Deney sistemi olarak biyoreaktör kullanılmıştır deneysel veriler daha önceki çalışmalardan elde edilmiştir. Bu biyoreaktörün ayar değişkeni olan hava akış hızına PRBS etki verilerek çıkış değişkeni olan çözünmüş oksijen derişimi ölçülmüştür. Bu bölümde procesten elde edilen veriler kullanarak sistem tanımlanmaya çalışılmıştır. Veriler MATLAB ortamına aktarılmış ve prosesin ARMAX modeli System Identification Toolbox'ı kullanarak elde edilmiştir. Sistem tanımlama, sistemden elde edilen verilere dayanılarak sistemin dinamik modelinin oluşturulmasıdır. Bu işlem verilen modelin çıktıları ile istemden alınan çıktılar birbirine yakın olana kadar model parametrelerinin ayarlanması yapılır. Modelin iyi olup olmadığı gerçek verilerle uyumlu olup olmadığına bağlıdır. Sistemde kullanılan ARMAX modelinin parametreleri $n_a=5$, $n_b=3$, $n_c=0$, $n_k=0$ alınmış, burada n_a , n_b , n_c polinomun derecesini n_k ise gecikmeyi ifade etmektedir. Sistem çıktısının zamanla değişimi şekil 4.13'de PRBS etkisi şekil 4.14'te gösterilmiştir.



Şekil 4.13 Sistem çıktısının PRBS etkisiyle değişimi

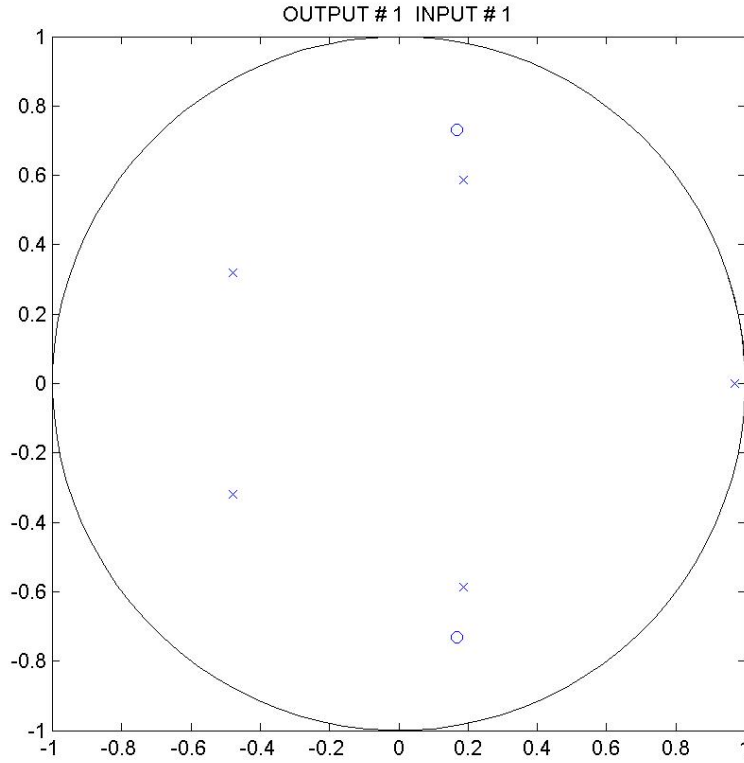


Şekil 4.14 PRBS etki



Şekil 4.15 Sistem ile ARMAX çıkışının karşılaştırılması

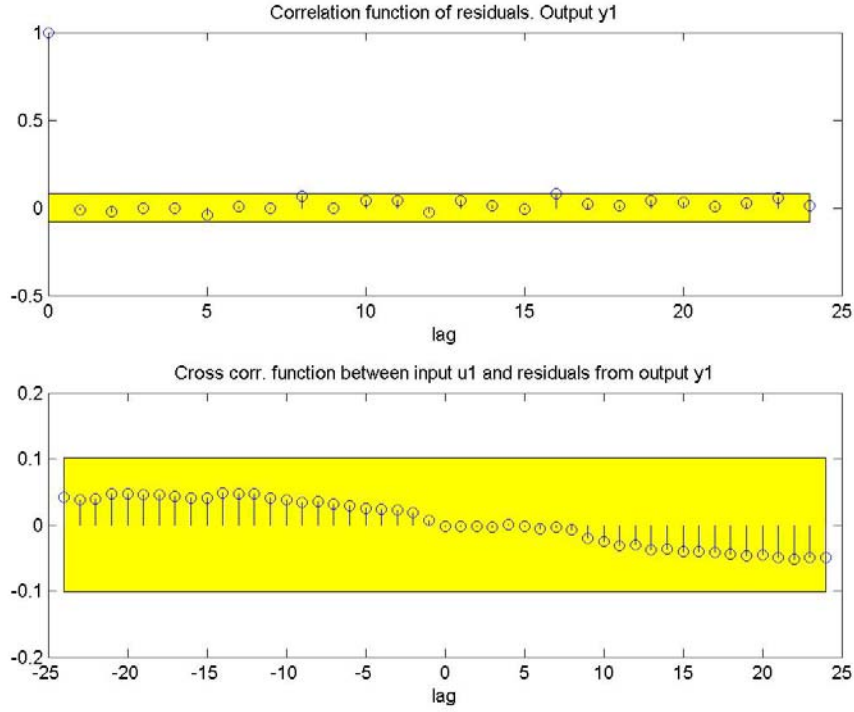
Şekil 4.15 ARMAX modeli prosesin yanıtını ile karşılaştırılmış ve modelin sistemi tanımladığı sonucuna varılmıştır. Zikzağı fazla olan deneysel, az olan ARMAX'tan elde edilen verilerdir. Sistemden örnekler iki saniyede bir alınmıştır.



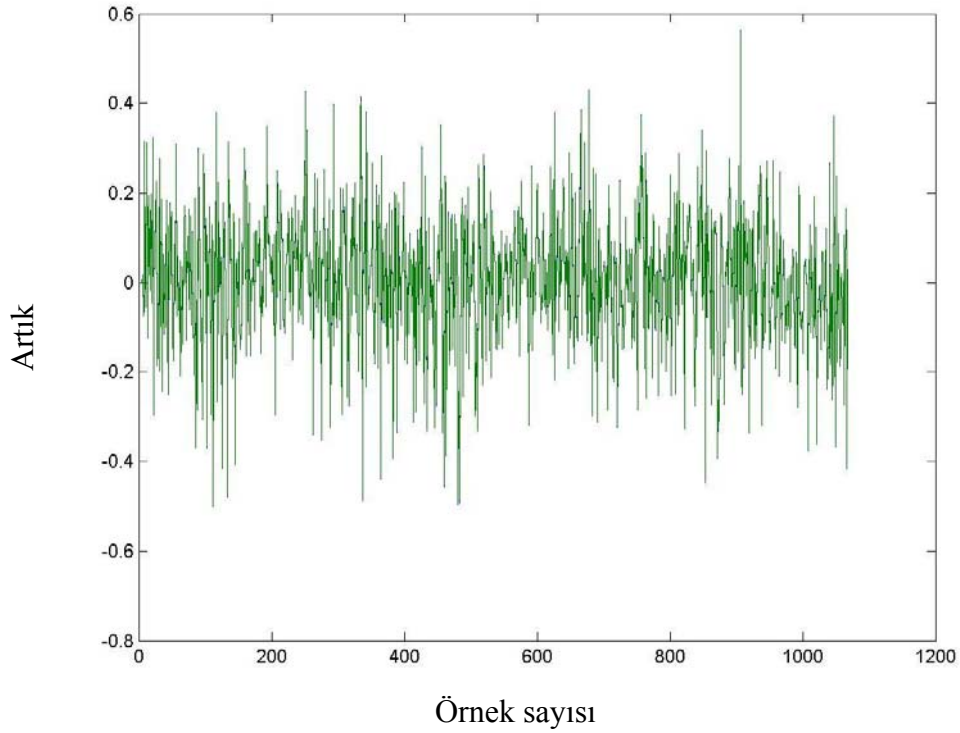
Şekil 4.16 ARMAX sıfırları ve kutupları

Şekil 4.16 ile ARMAX kutupları çember içine düşmüştür, Yani belirtilen n_a , n_b , n_c ve n_k değerlerinde ARMAX modeli ile kullanılabilir olduğu ve sistemin kararlı olduğu şekil 4.16'dan anlaşılmaktadır. Bulunan bu ARMAX parametrelerinin tesadüfi olup olmadığını anlamak için oto korelasyon, kros korelasyon grafikleri çizilmiştir (şekil 4.17). Otokorelasyon giriş ve çıkış değişkenleri arasındaki ilişkinin yönünü ve derecesini gösterir. Otokorelasyon belirli bir değer arasında olmalıdır bu yüzden otokorelasyon grafikleri çizilmelidir. Çizilen grafikte modelin sınırları belli bir band aralığı içinde kalıyorsa modelimiz sistem için uygundur. Etki-tepki büyüklükleri arasında gerçekleştirilecek bir çapraz-korelasyon işlemi ile de sistemin tepkisinin, etkiye göre gecikip-gecikmediği belirlenebilmektedir. Yani etkiyen büyüklük tepkiye sebep olmuş ise sistemde herhangi bir gecikme durumu meydana gelmemektedir. Eğer bir değişim söz konusu değilse sistemde gecikme meydana gelmektedir. Sistemin parametrelerinin tesadüfi olmadığı görülmüştür. Sistemle model arasındaki artıkları (tahmin hataları) görmek için şekil 4.18 çizilmiştir. ARMAX'ın artıkları kabul edilebilir aralıktadır.

ARMAX modeli teorik çalışmalarda mayanın tüm üreme fazları için proses modeli olarak kullanılmıştır.



Şekil 4.17 Oto korelasyon ve kros korelasyon grafikleri



Şekil 4.18 ARMAX model artıkları

Çalışmadan bulunan sistemin A ve B ARMAX parametreleri eşitlik 4.13'te verildiği gibidir. Bu katsayıları hesaplamak için kullanılan Matlab programı, modelin simülasyonu ve zamana bağlı olarak ifadesi ekte verilmektedir.

$$A(q) = 1 - 0.3939q^{-1} - 0.2256q^{-2} - 0.08576q^{-3} - 0.09255q^{-4} - 0.1353q^{-5}$$

$$B(q) = 0.0594 - 0.01965q^{-1} + 0.03407q^{-2}$$
(4.13)

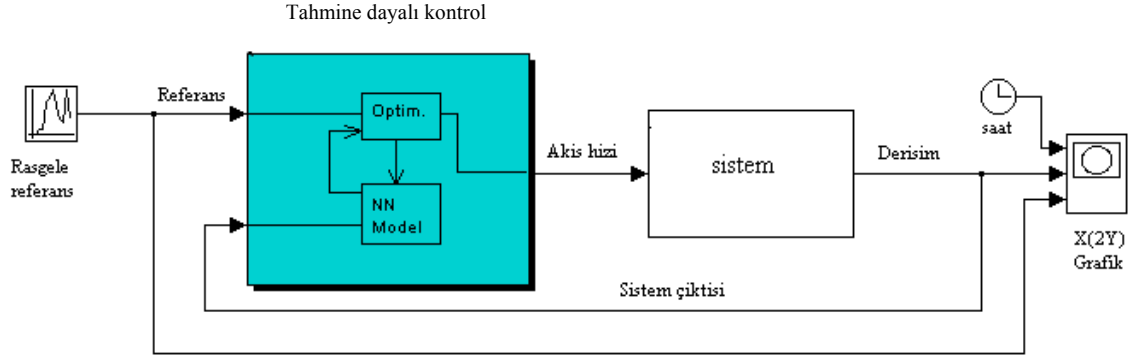
4.2.1 Tahmine dayalı kontrol

a) Taşınım olaylarına dayalı eşitlikleri kullanılarak modellenen sistemin tahmine dayalı oksijen kontrolü (Governing equations)

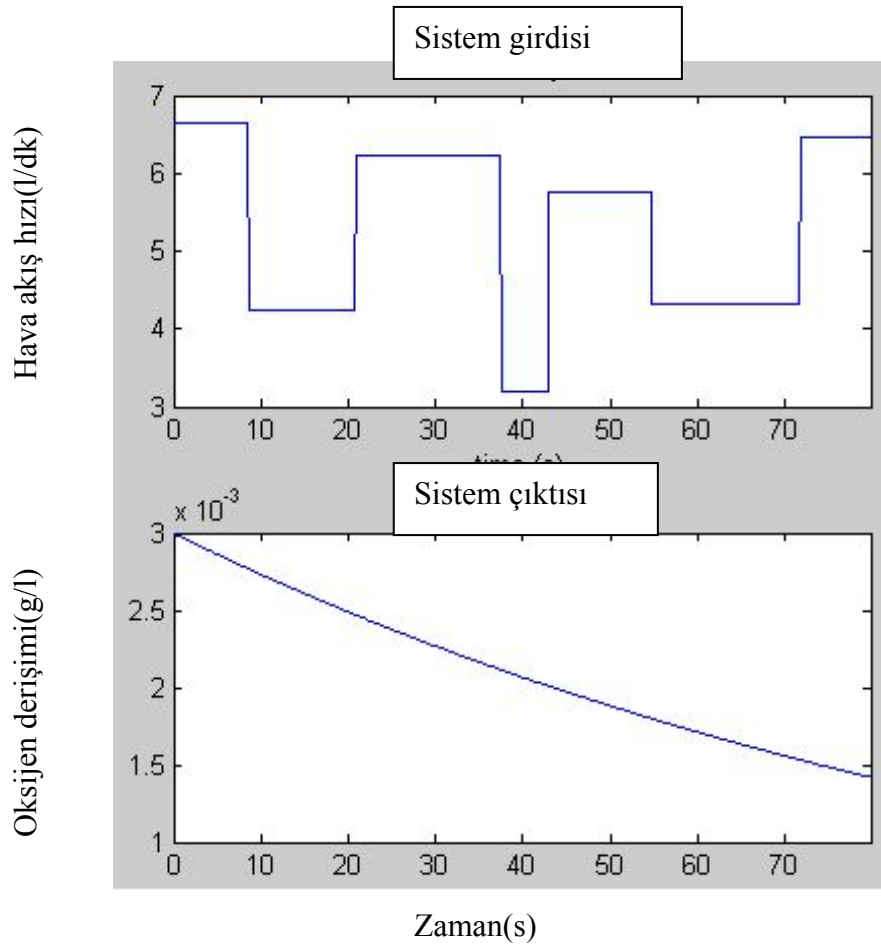
S. cerevisiae mikroorganizmasının üretildiği bir biyoreaktörde Oksijen derişimi tahmine dayalı kontrol algoritması ile kontrolü sağlanmıştır. Eşitlik 4.14'te verilen amaç fonksiyonu önerilmiştir.

$$J = \sum_{j=N_1}^{N_2} (y_r(t+j) - y_m(t+j))^2 + \xi \sum_{j=1}^{N_u} (u'(t+j-1) - u'(t+j-2))^2 \quad (4.14)$$

Eğitimdeki hava akış hızı 0-7 L/dk arasında seçilmiştir. Seçilen akış hızına karşı proses çıktısı olan oksijen derişimi şekil 4.20'de görölmektedir. Kontrol edilen oksijen derişimi $1 \cdot 10^{-3}$ ile $3.4 \cdot 10^{-3}$ g/l değerleri arasında tutulmaya çalışılmıştır. Tahmine dayalı kontrol algoritmasını matlab toolbox kullanılarak hesaplanmıştır (şekil 4.19). Kullanılan ağ 10 nöronludur. Örnekleme aralığı 0.5 saniyeye ayarlanmış, örnekleme sayısı kullanıcı tarafından seçilmiştir.

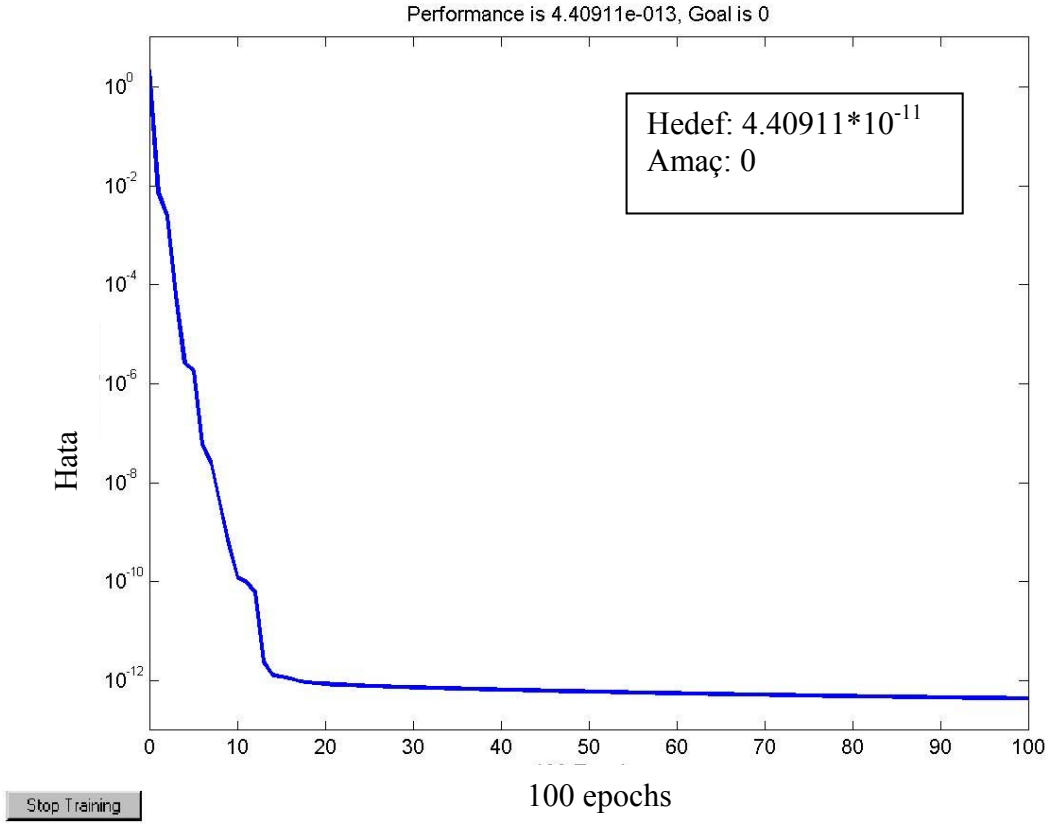


Şekil 4.19 Tahmine dayalı kontrol blok diyagramı



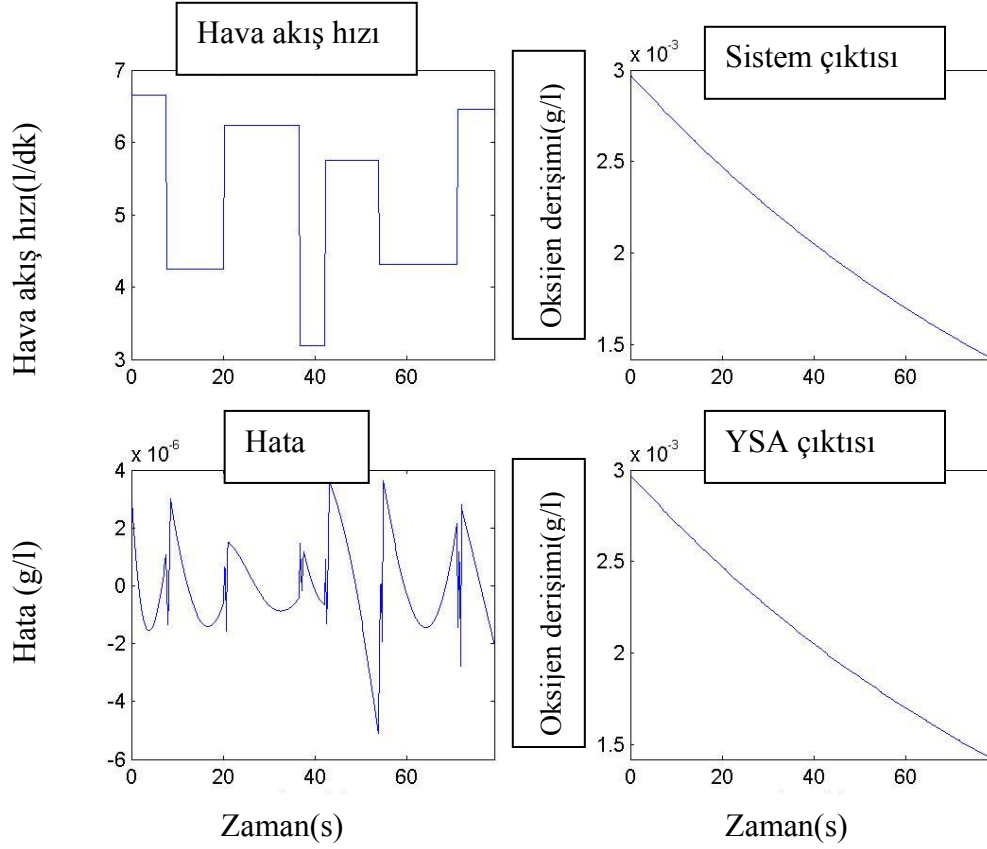
Şekil 4.20 Sistem girdisi ve sistemin bu girdilere olan tepkisi

YSA 100 epochs sonucu ağıın elde ettiği hata değeri (hatalarının karelerinin toplamının hata sayısına bölümü) şekil 4.21’de verilmiştir. Eğitim için Levenberg Marquardt algoritması kullanılmıştır.



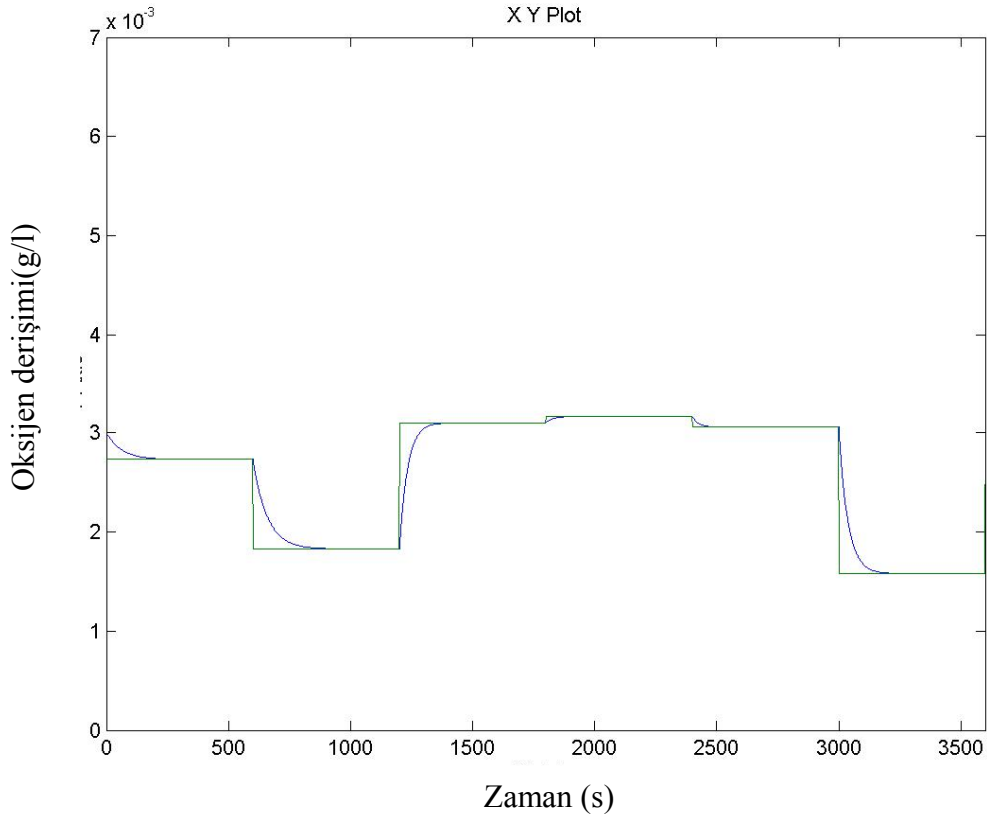
Şekil 4.21 Yüz deneme sonucunda YSA hata değeri

Hava akış hızının, sistemdeki oksijen derişiminin zamanla değışimi ve oksijen derişiminin YSA çıktısı ve hata değeri toplu bir şekilde şekil 4.22’de gösterilmiştir.



Şekil 4.22 Hava akış hızının, Taşınım olaylarına bağlı diferansiyel (governing) eşitliklerinden elde edilen çıktının, YSA çıktısının ve iki çıktı arasındaki fark (hata) değeri

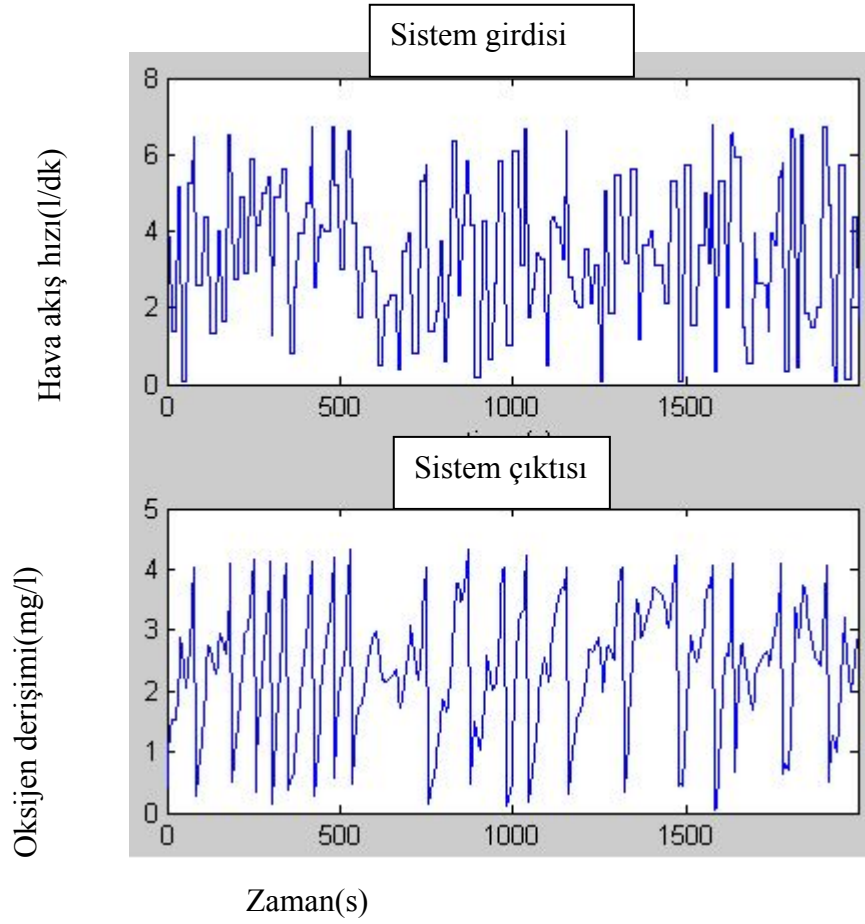
Oksijen derişimi 1 mg/L ile 3.4 mg/L arasında tutulmaya çalışılmıştır. Bu değerler arasında üstsel üreme fazı için değişken set noktası için şekil 4.23’de görüldüğü gibi kontrol sağlanmıştır. Şekil 4.23’den de anlaşılağı gibi küçük set noktası değişimlerinde oturma süresi kısalmış set noktası değişimi büyüdüğünde oturma süresinin uzadığı görülmüştür.



Şekil 4.23 Sistemin oksijen kontrolü

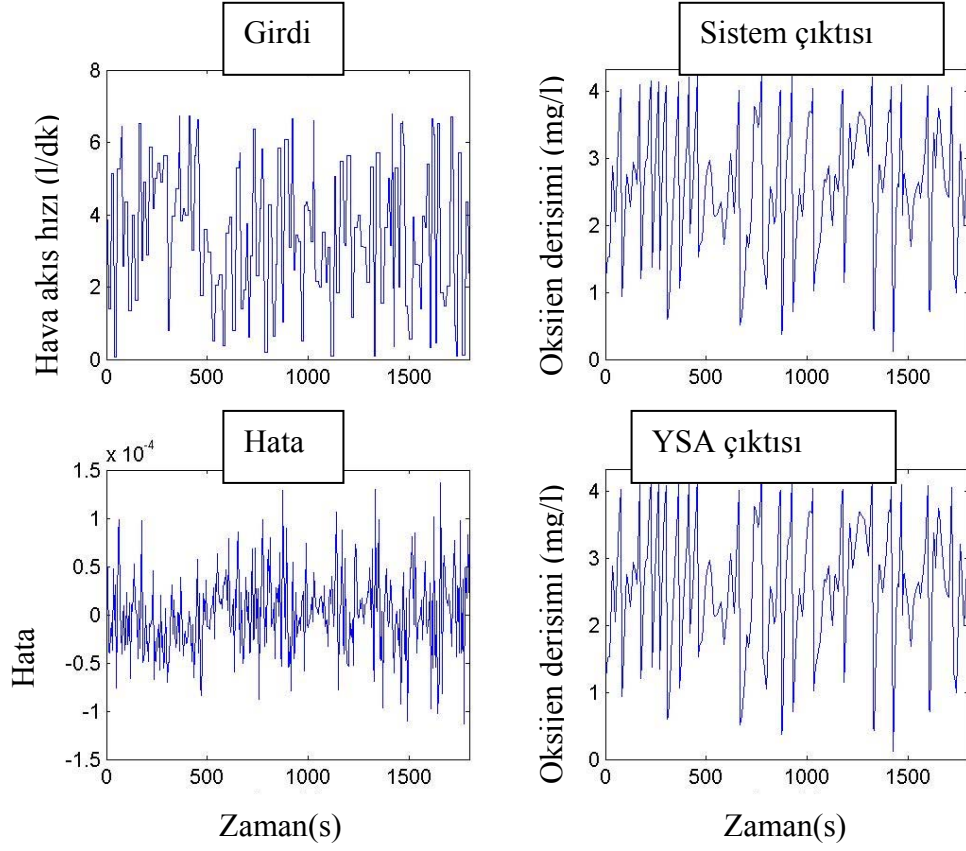
b) Polinom tipi (ARMAX) model kullanılarak modellenen sistemin oksijen kontrolü

Eşitlik 4.14'deki amaç fonksiyonu önerilmiştir. Eğitimdeki hava akış hızı 0-7 l/dk arasında seçilmiştir. Seçilen u ayar değişkenine karşı (hava akış hızı) prosesi çıktısı şekil 4.24'da görülmektedir. Hava akış hızına rasgele giriş değerleri verilmiştir. Kontrol edilen oksijen derişimi 1 ile 3.4 mg/l değerleri arasında tutulmaya çalışılmıştır. Tahmine dayalı kontrol algoritmasını matlab toolbox kullanılarak hesaplanmıştır. On nöronlu ağ yapısı kullanılmış olup örnek alma süresi 5 s olarak ayarlanmıştır. Sistem çıktısı ve girdi gecikmesi iki alınmıştır.

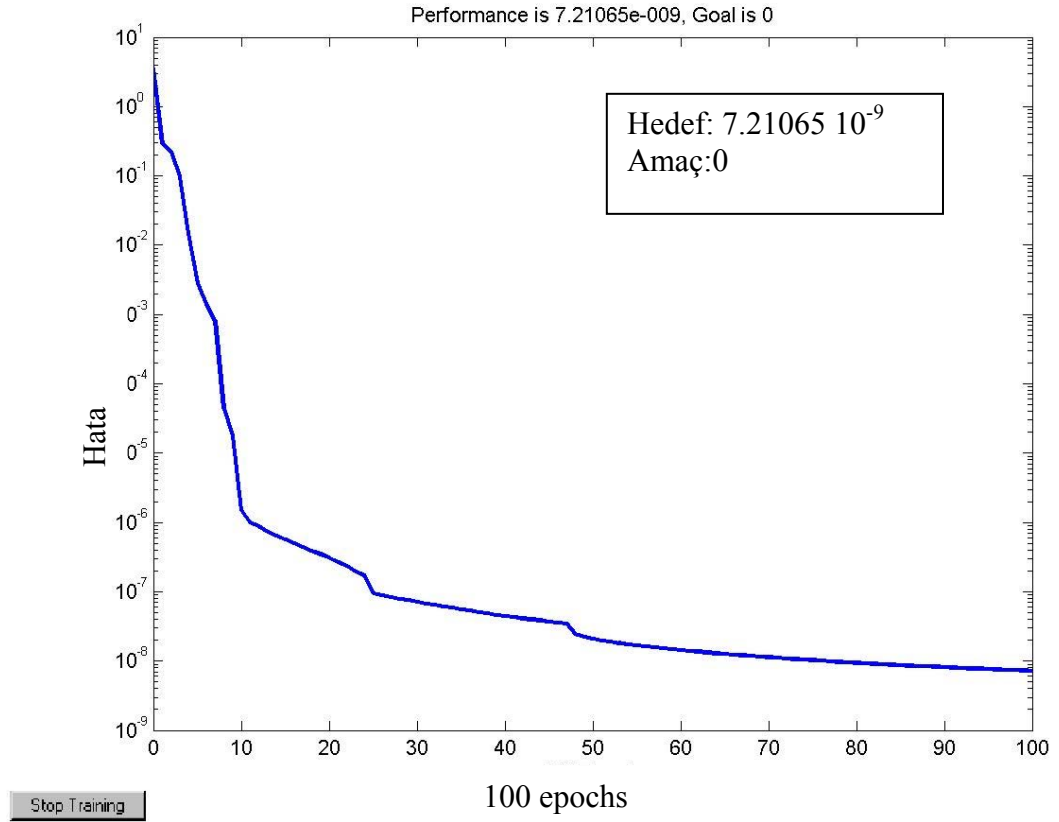


Şekil 4.24 ARMAX modelinin giriş ve çıkış değerleri

Sistemin girdi ve çıkışı, YSA çıkışı ve sistem ve YSA çıkışı arasındaki hata şekil 4.25’de gösterilmiştir. Yapay sinir ağının 100 epochs sonucu elde ettiği hata değeri (hatalarının karelerinin toplamının hata sayısına bölümü) şekil 4.26’da verilmiştir. Hata değeri $7.21065 \cdot 10^{-9}$ bulunmuştur. Eğitim için Levenberg Marquardt algoritması kullanılmıştır.

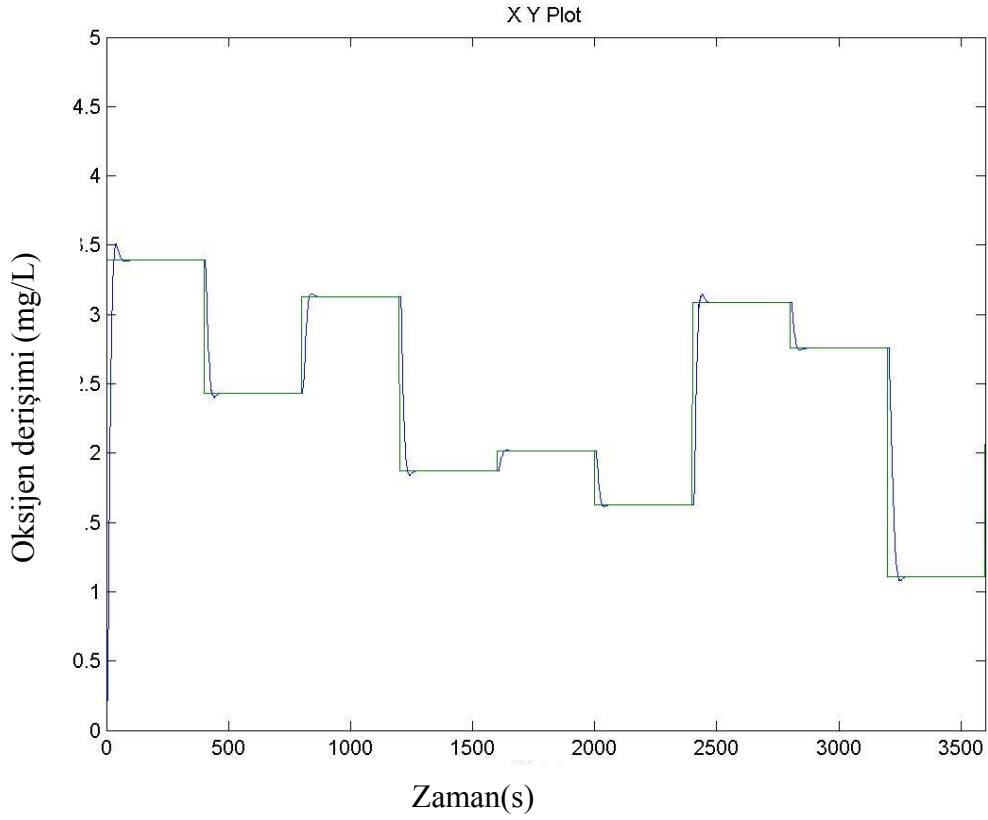


Şekil 4.25 Ayar değişkeni hava akış hızının, polinom tipi modelden elde edilen çıktının, YSA çıktısının ve YSA çıktısı ile model arasındaki hata değeri



Şekil 4.26 Yüz Epochs için hata değeri

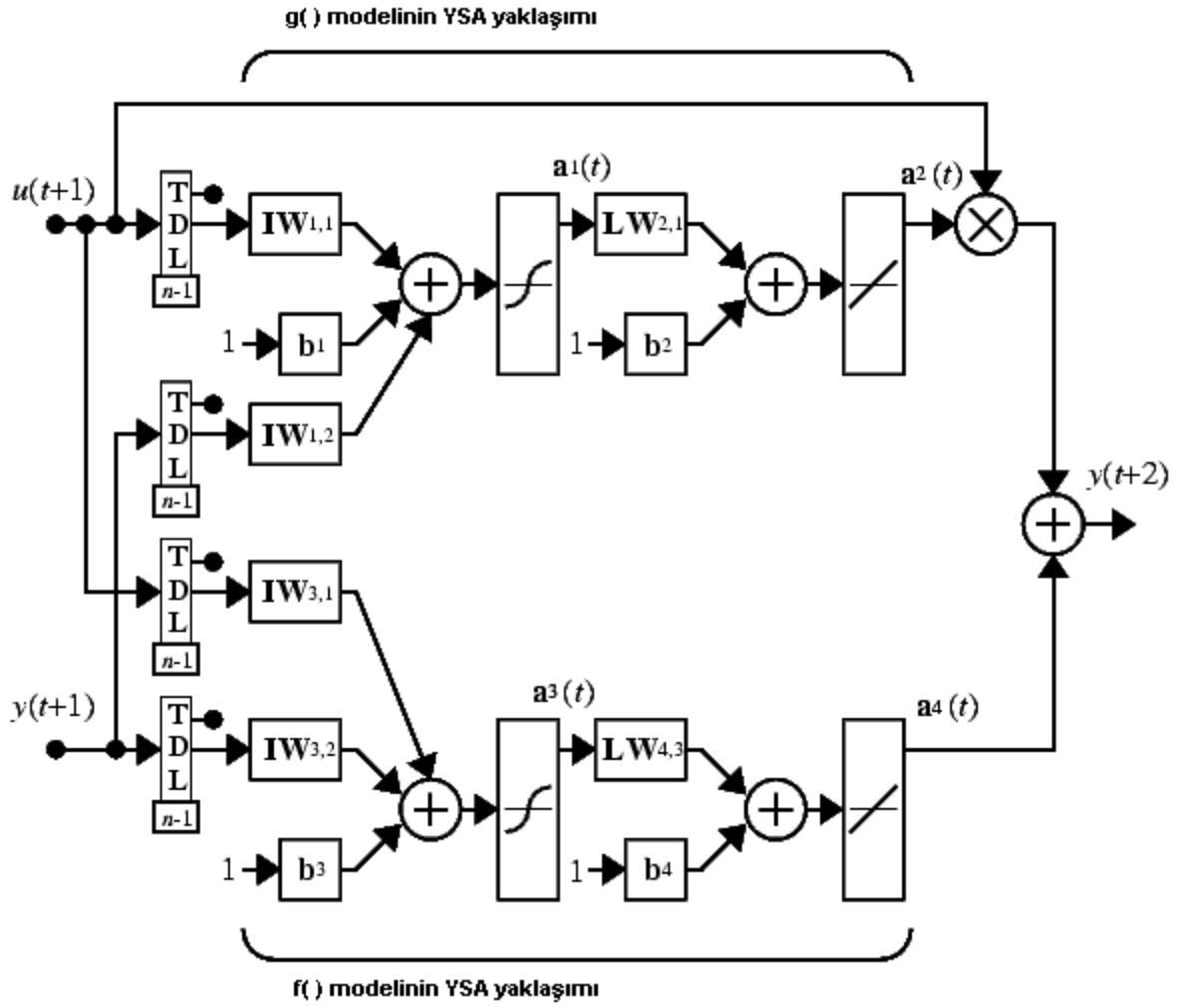
Oksijen derişimi deneysel olarak 1 mg/L ile 3.4 mg/L arasında değıştirilmiştir. Bu değerler arasında değışken set noktası için kontrol şekil 4.27’de görüldüğü gibi sağlanmıştır. Polinom tipi modelle modellenen sistemin oturma süresinin taşınım olaylarına dayalı diferansiyel eşitlik ile modellenen modelden daha hızlı belirtilen set noktasına ulaştığı görülmüştür. ARMAX modelinde hava akış hızının daha fazla etkili olduğu sonucuna varılmıştır.



Şekil 4.27 ARMAX modelinin oksijen kontrolü

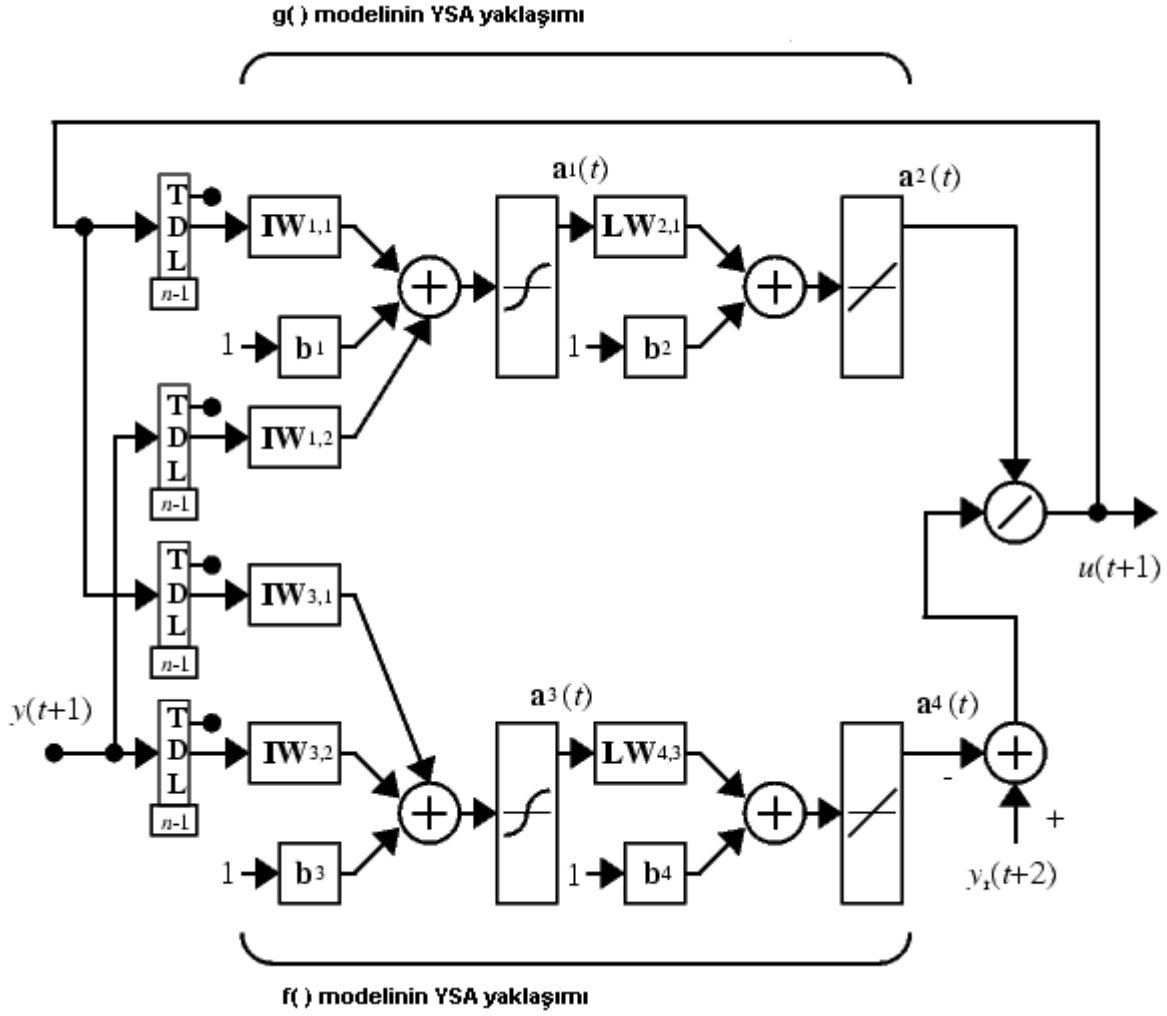
4.2.2 NARMA-L2 (Geri beslemeli linerizasyon) kontrol

Sistemin modellenmesi için taşıyım olaylarına bağlı eşitlikler ve ARMAX modeli kullanılmıştır, bu iki model için ayrı ayrı NARMA-L2 kontrolü sağlanmıştır. eğitim için veriler her bir modelden ayrı ayrı elde edilmiş bu veriler 10 nöronlu YSA eğitmek için kullanılmıştır ve bu model kullanılarak sistemin oksijen kontrolü başarılı şekilde sağlanmıştır. Tahminde kullanılan modelin ağ yapısı (eşitlik 3.26’da önerilen denklem) şekil 4.28’de gösterilmiştir.



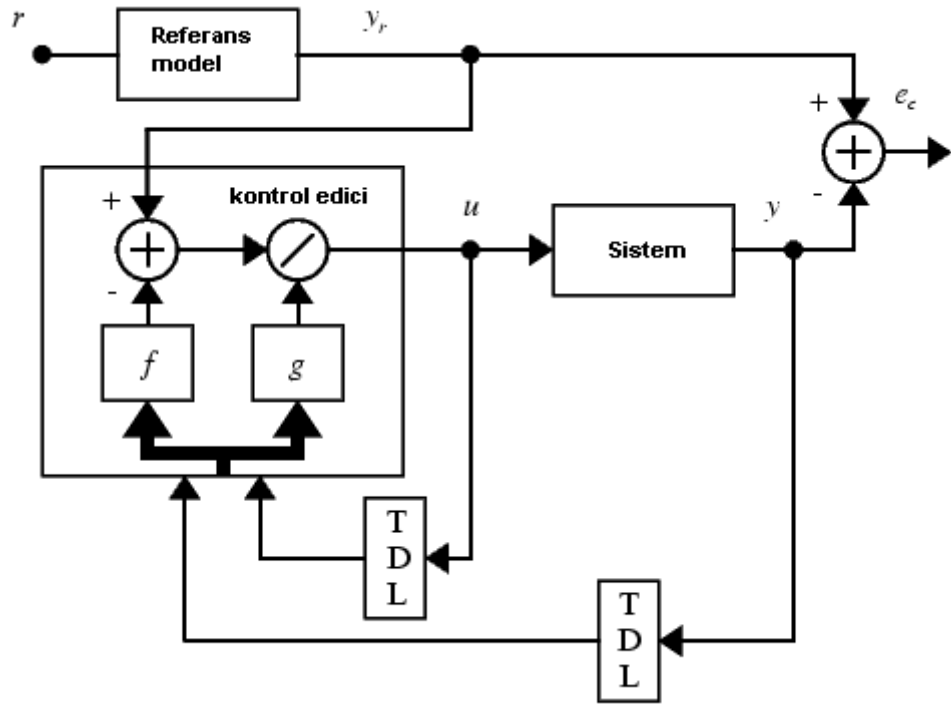
Şekil 4.28 Tahmin için kullanılan modelin yapısı

Kontrol edici ağ yapısı şekil 4.29’de (eşitlik 3.27’de önerilen denklem) gösterilmiştir. Şekil 4.29’da TDL ayar değişkeninin ve sistem çıktısının geçmiş ve şuan ki değerleridir.

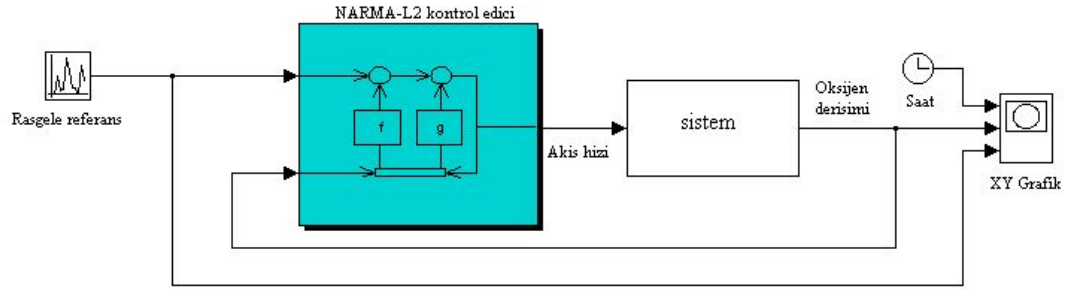


Şekil 4.29 Kontrol edicinin ağ yapısı

Tüm sistemin blok yapısı şekil 4.30’de verilmiştir. Sistemin toolbox kullanımın blok diyagramı gösterimi şekil 4.31’da gösterilmiştir. Şekil 4.30’dan anlaşılacağı üzere kontrol ediciye ayar değişkeni ($u(t), u(t-1), \dots$), sistem çıktısı ($y(t), y(t-1), \dots$), set noktası (r) girmektedir. Kontrol edici çıktısı ise ayar değişkenidir.

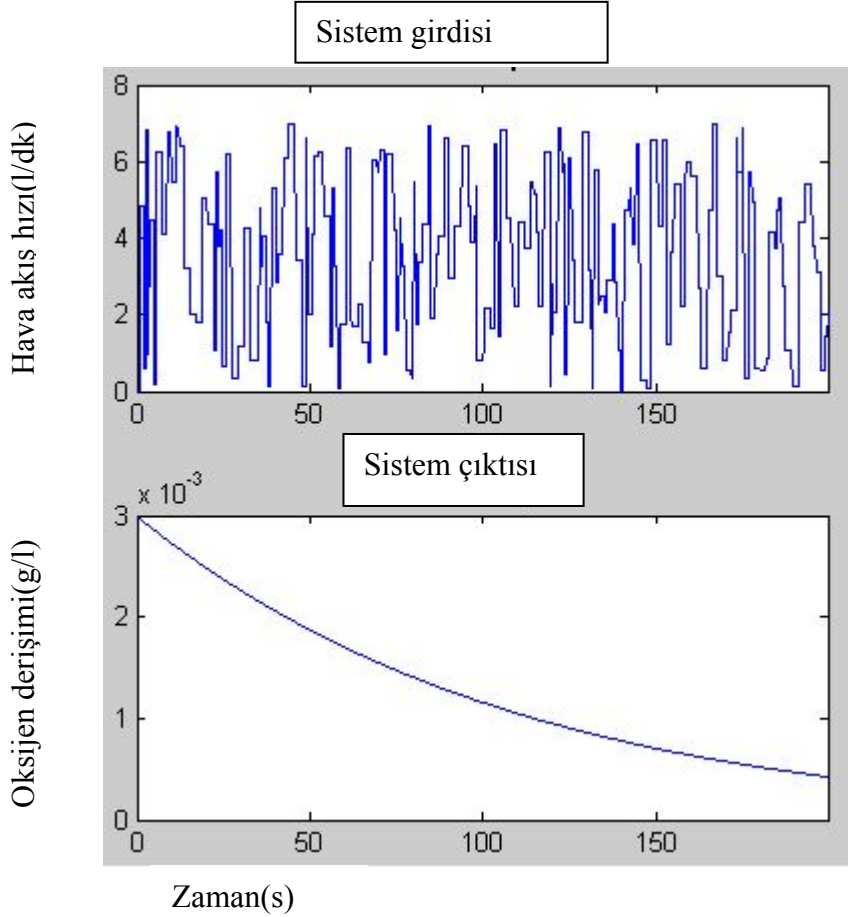


Şekil 4.30 Tüm sistemin blok diyagramı



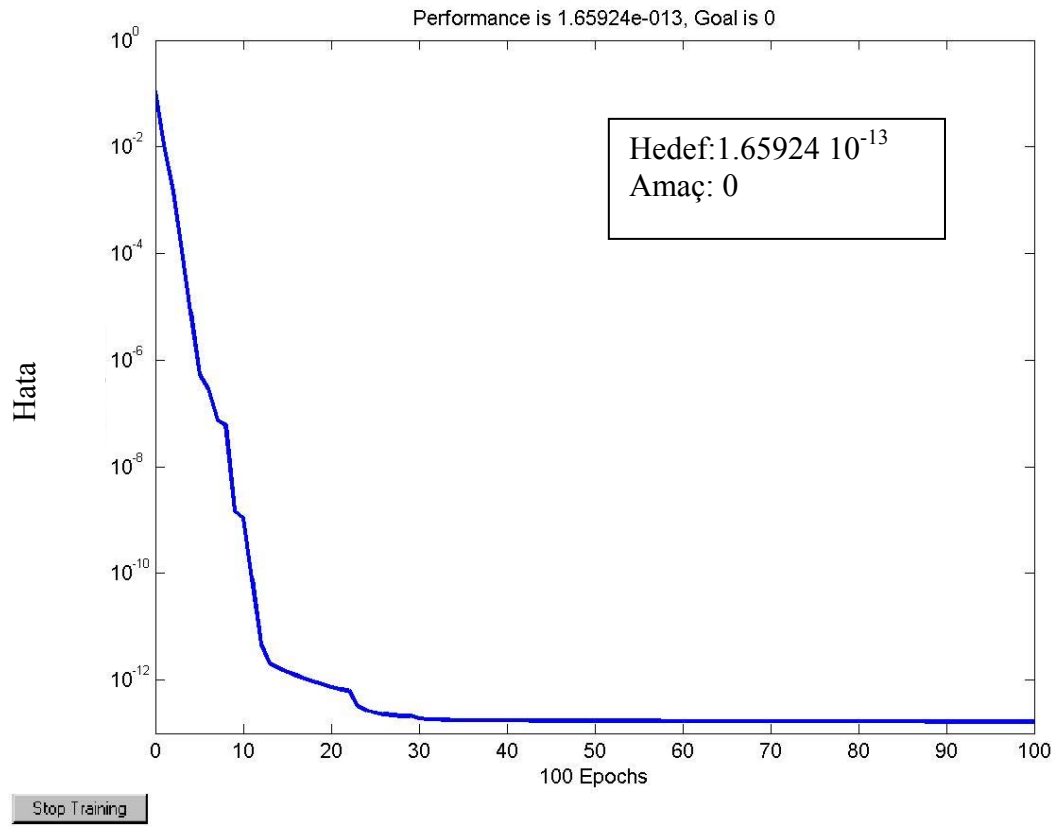
Şekil 4.31 Sistemin blok diyagramının matlab toolbox kullanımı

a) Taşınım olaylarına dayalı modelli kullanılarak modellenen sistemin kontrolü

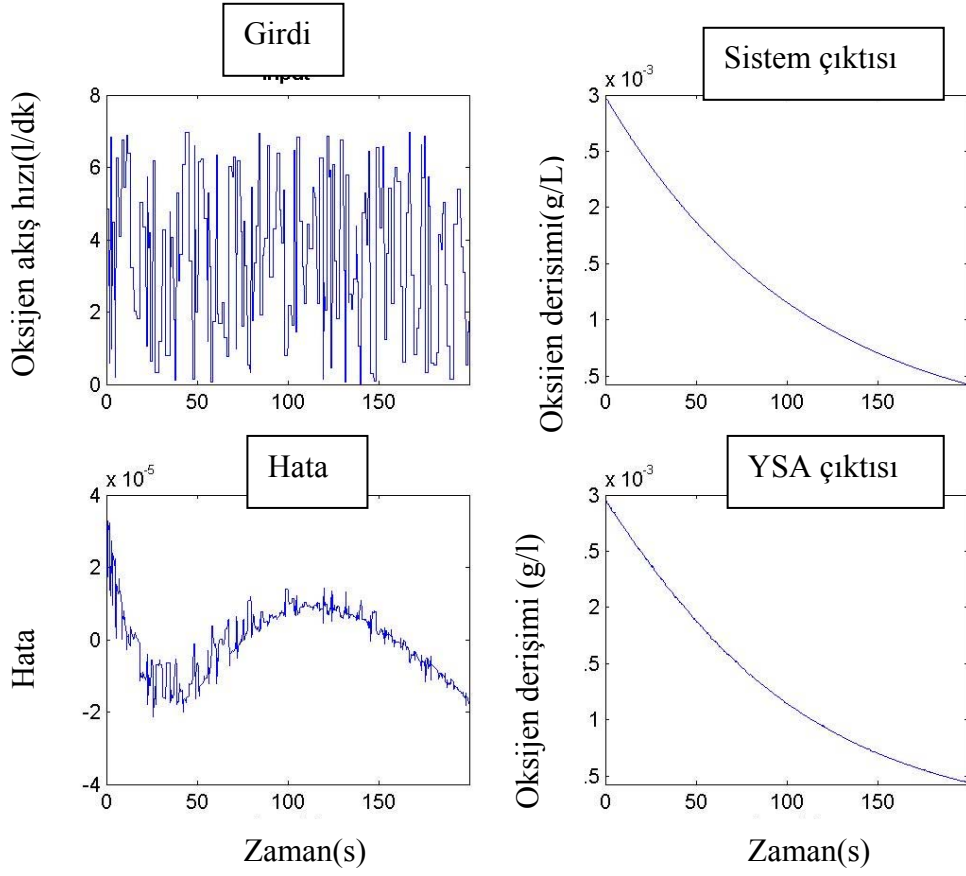


Şekil 4.32 Governing eşitliklerinin giriş ve çıkış verileri

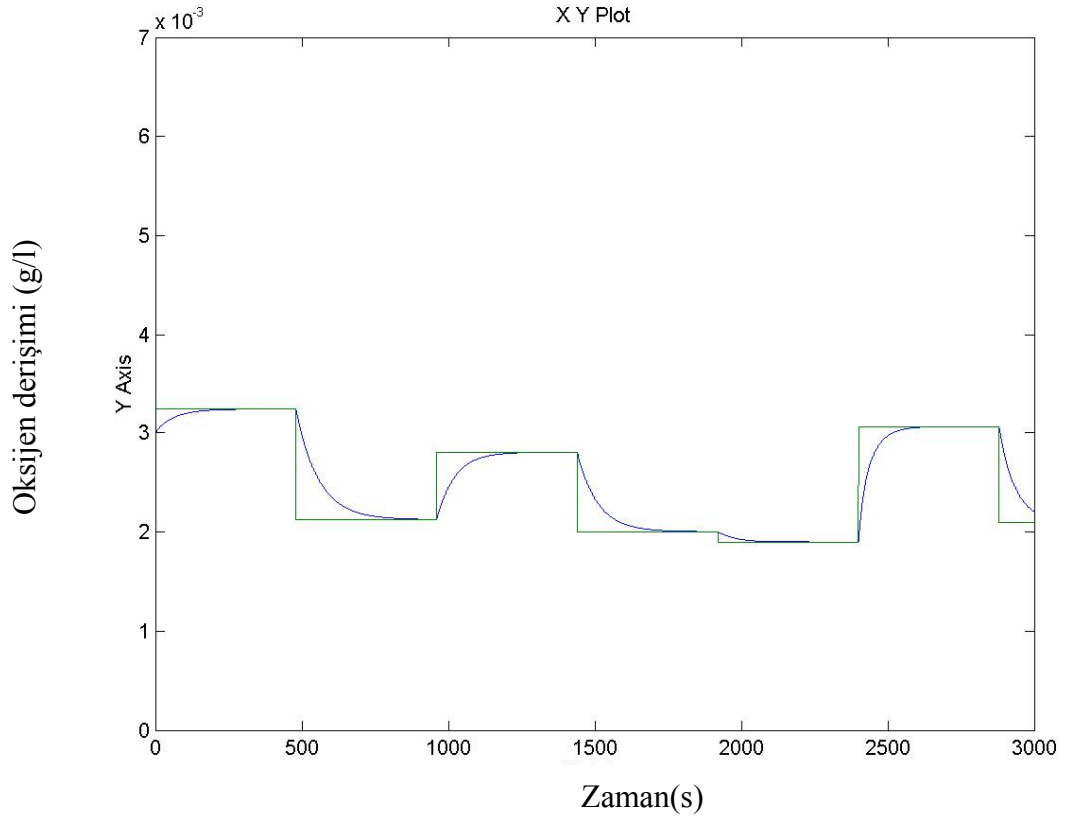
Taşınım olaylarına dayalı modelli kullanılarak kontrol sağlanmaya çalışılmıştır. modelin girdi ve çıkışı değişik hava akış hızlarında şekil 4.32’de gösterilmiştir. 100 epochs sonucu elde ettiği hata değeri (hataların kareleri toplamının toplam hata sayısına bölümü) şekil 4.33’de verilmiştir hata değeri $1.65924 \cdot 10^{-13}$ bulunmuştur. Eğitim için Levenberg Marquardt algoritması kullanılmış ve on nöronlu ağ yapısı kullanılmıştır. Örneklem aralığı 0.1 saniye alınmıştır. Şekil 4.34’de model girdisi, çıkışı, eğitilen YSA çıkışı ve model çıkışı ile YSA çıkışı arasındaki fark gösterilmektedir. Sistem girdisi gecikmesi üç, çıkışı gecikmesi iki alınmıştır.



Şekil 4.33 Yüz Epochs için hata değeri



Şekil 4.34 Sistem girdisi, çıktısı,YSA çıktısı ve hata değeri

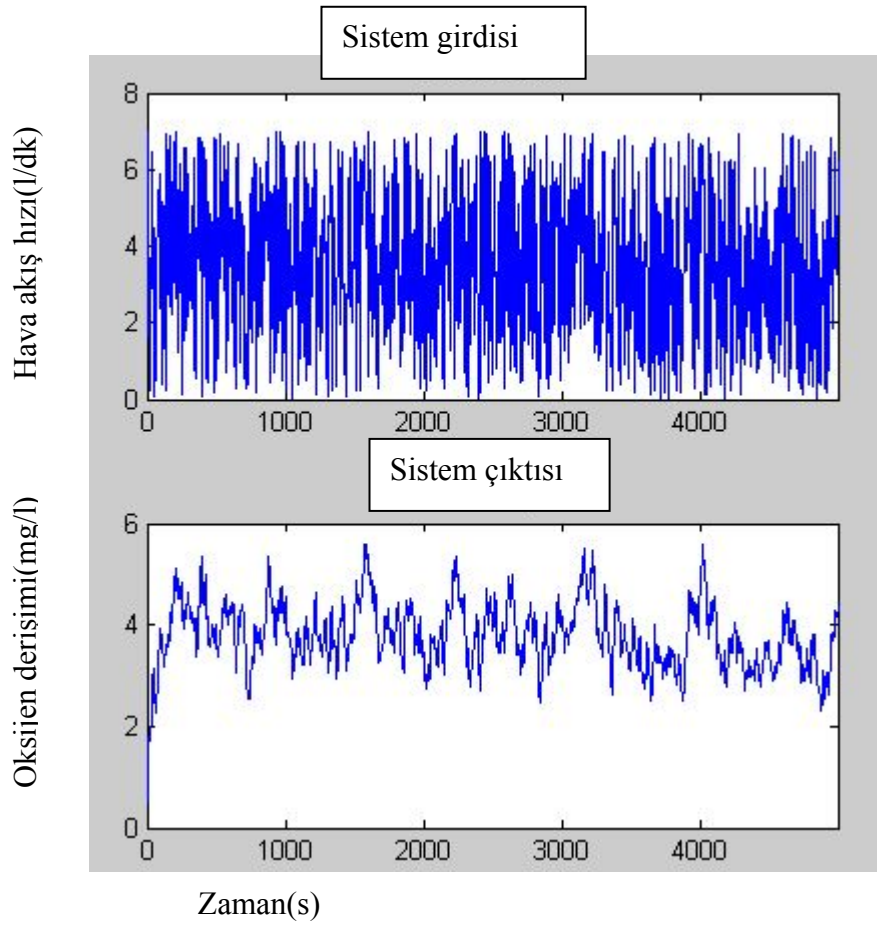


Şekil 4.35 NARMA-L2 yöntemiyle oksijen kontrolü

Şekil 4.35’de sistemin değişik set noktaları için oksijen kontrol grafiğı verilmiştir. Şekilden de anlaşılacağı gibi sistemin kontrolü başarılı bir şekilde sağlanmıştır.

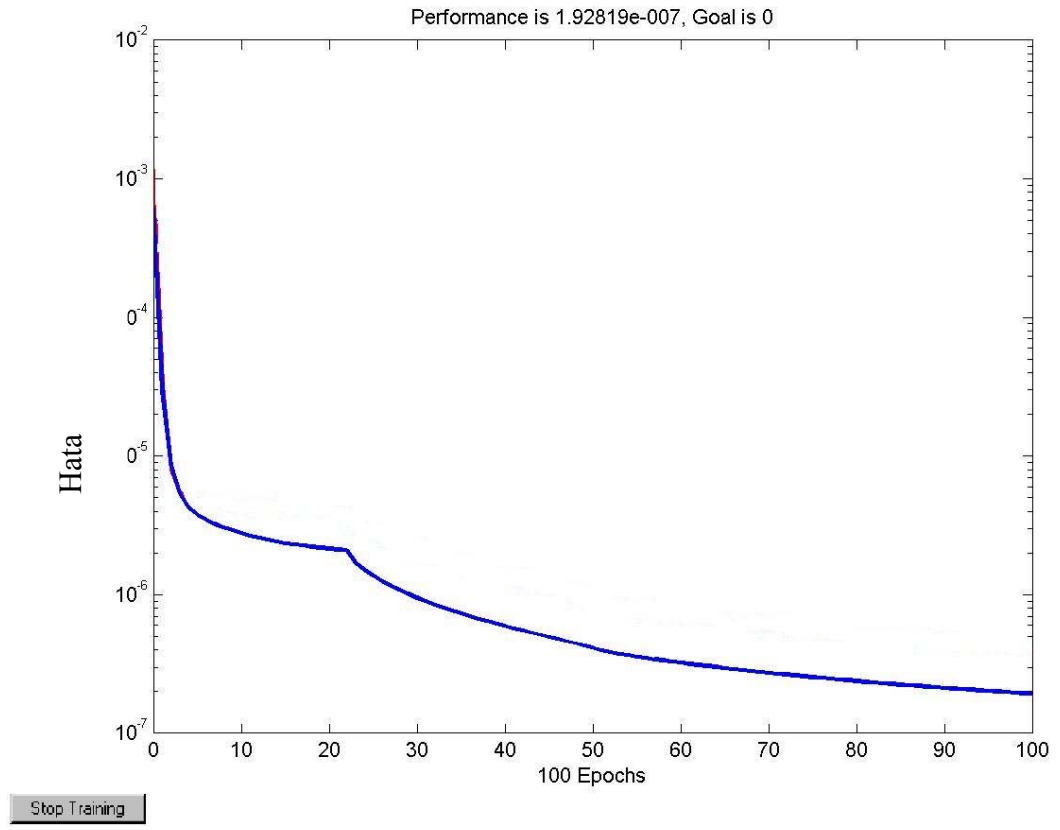
b) Polinom tipi (ARMAX) model kullanılarak modellenen sistemin kontrolü

Oksijenin akış hızının zamanla değişimi ARMAX ile modellenmiş elde edilen denklemlerin giriş ve çıkışı değerleri şekil 4.36’da görölmektedir. NARMA-L2 algoritmasını matlab toolbox kullanılarak hesaplanmıştır. ARMAX denkleminde ağı eğitilmesi için 5000 veri elde edilmiştir. On nöronlu YSA kullanılmış örnekleme aralığı 5 saniye alınmıştır. Sistem girdi ve çıktı gecikmesi iki alınmıştır.

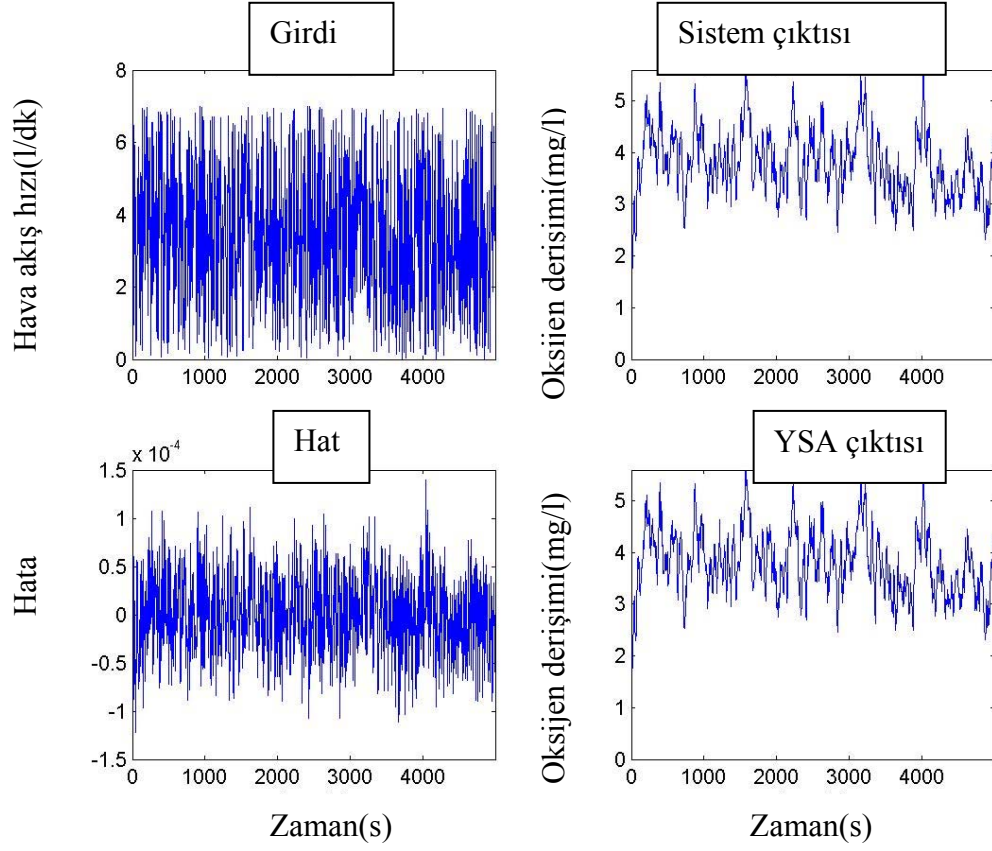


Şekil 4.36 Model girdisi ve çıktısı

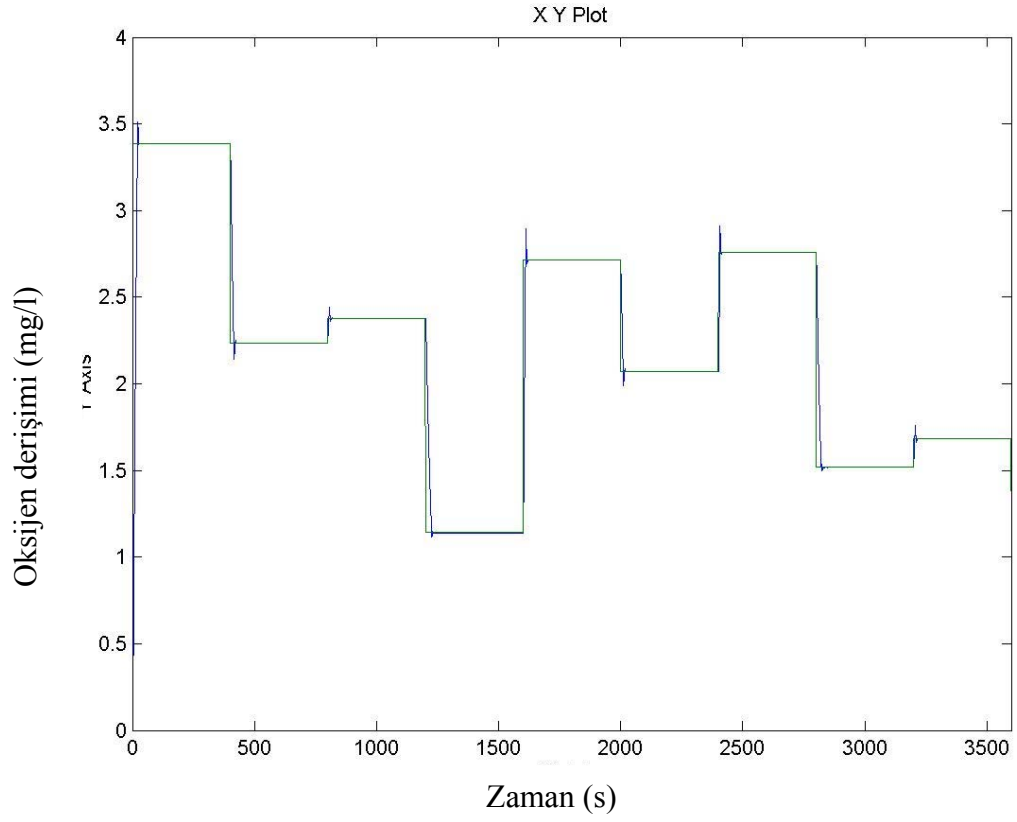
Test için 100 epochs denenmiş ve hata değeri (hatalarının kareleri toplamının hata sayısına bölümü) $1.92819 \cdot 10^{-7}$ bulunmuştur (şekil 4.37). Eğitim için Levenberg Marquardt algoritması kullanılmıştır. Model çıktısı, YSA çıktısı ve bunların arasındaki hata değerleri toplu halde Şekil 4.38’de gösterilmiştir.



Şekil 4.37 Eğitimdeki hata değeri



Şekil 4.38 Model çıktısı, YSA çıktısı ve bunlar arasındaki hata değerleri

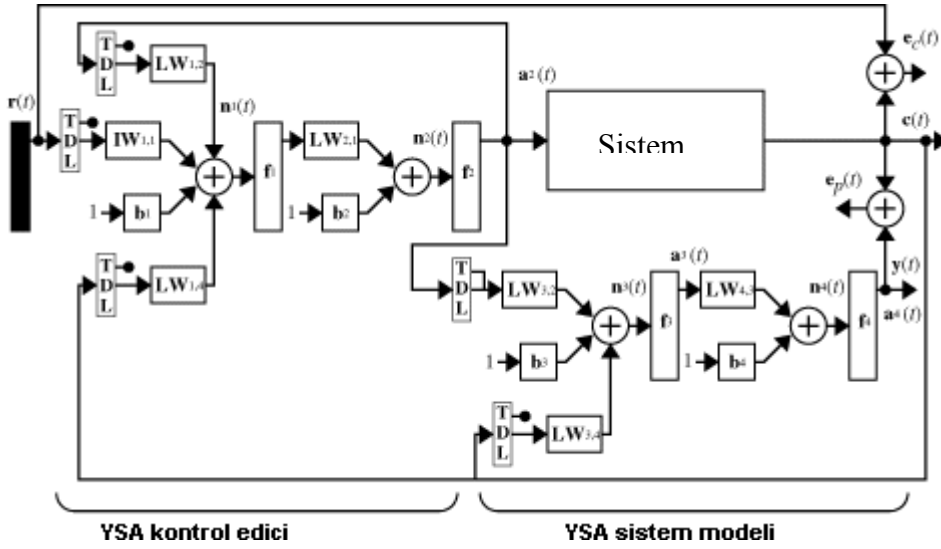


Şekil 4.39 NARMA kontrol edici ile ARMAX'tan elde edilen modelin kontrolü

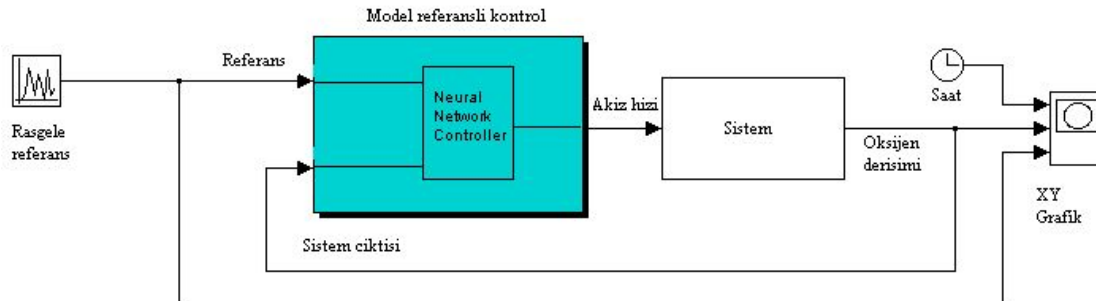
ARMAX ile modellenen sistemin oksijen kontrolü şekil 4.39'de görüldüğü gibi başarı ile sağlanmıştır.

4.2.3 Model referanslı kontrol

Model referanslı kontrol yapısında iki tane YSA kullanılmıştır biri kontrol edici için öbürü sistemin modellenmesi için kullanılmıştır. Sistem modeli olarak taşıyım olaylarına bağlı eşitlikler (governing equation) ve ARMAX modeli kullanılmıştır. Her bir model için ayrı ayrı kontrol sağlanmaya çalışılmıştır. Kullanılan ağ yapısı 10'ar nöronlu tek tabakalı YSA modeli kullanılmıştır. Kullanılan sistemin yapısı şekil 4.40'da gösterilmiştir. Toolbox kullanımı ise şekil 4.41'de gösterilmektedir.



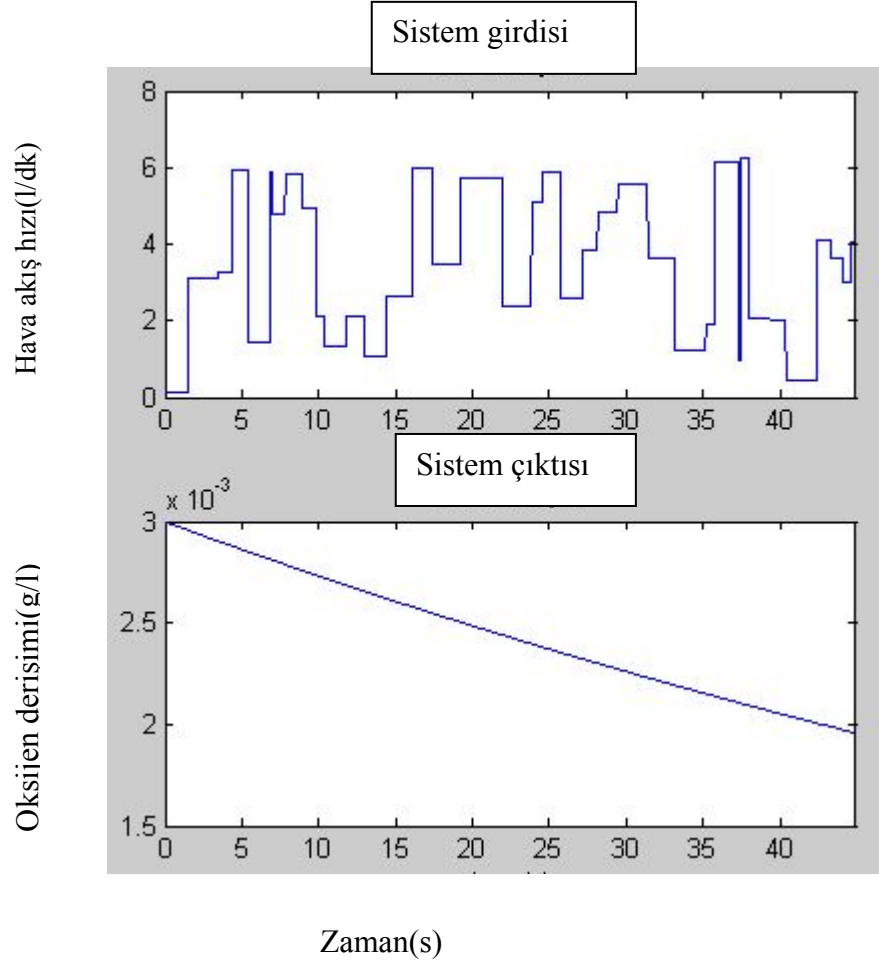
Şekil 4.40 Model referanslı kontrol edici yapısı



Şekil 4.41 Matlab toolbox model referanslı kontrol edici

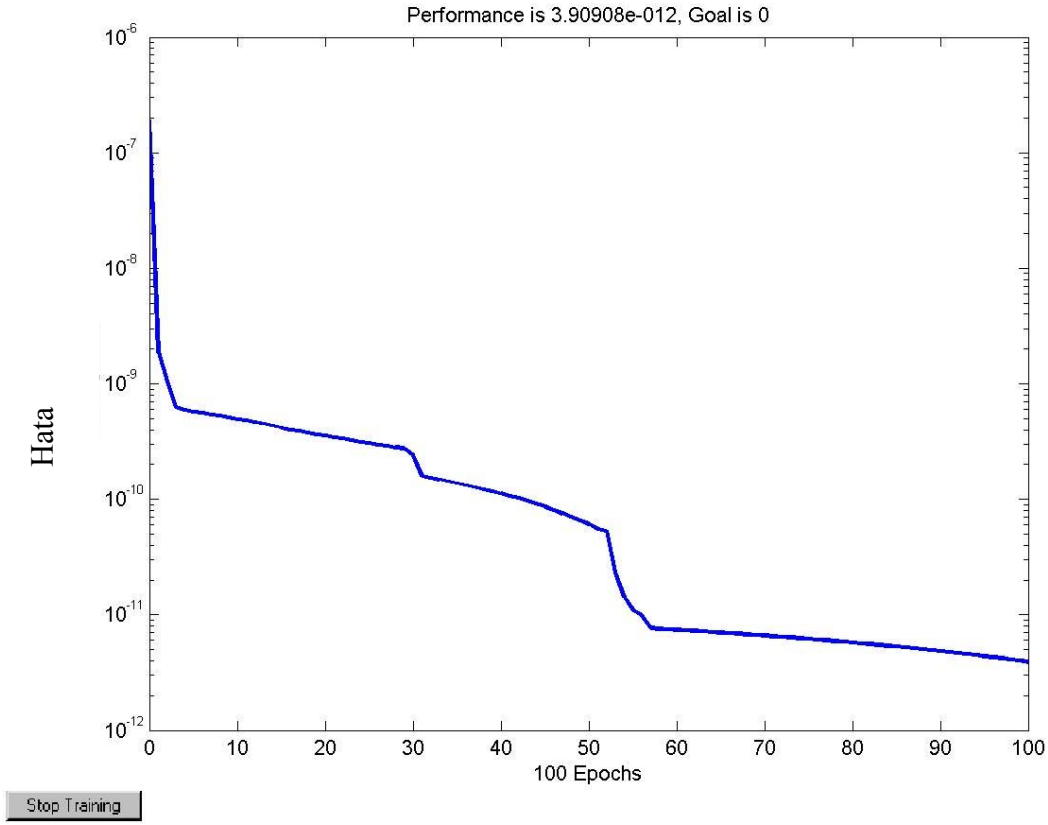
a) Taşınım olaylarına dayalı model kullanılarak modellenen sistemin kontrolü

Eğitim için 500 veri kullanılmış bu veriler kullanılarak sistem modellenmiştir.



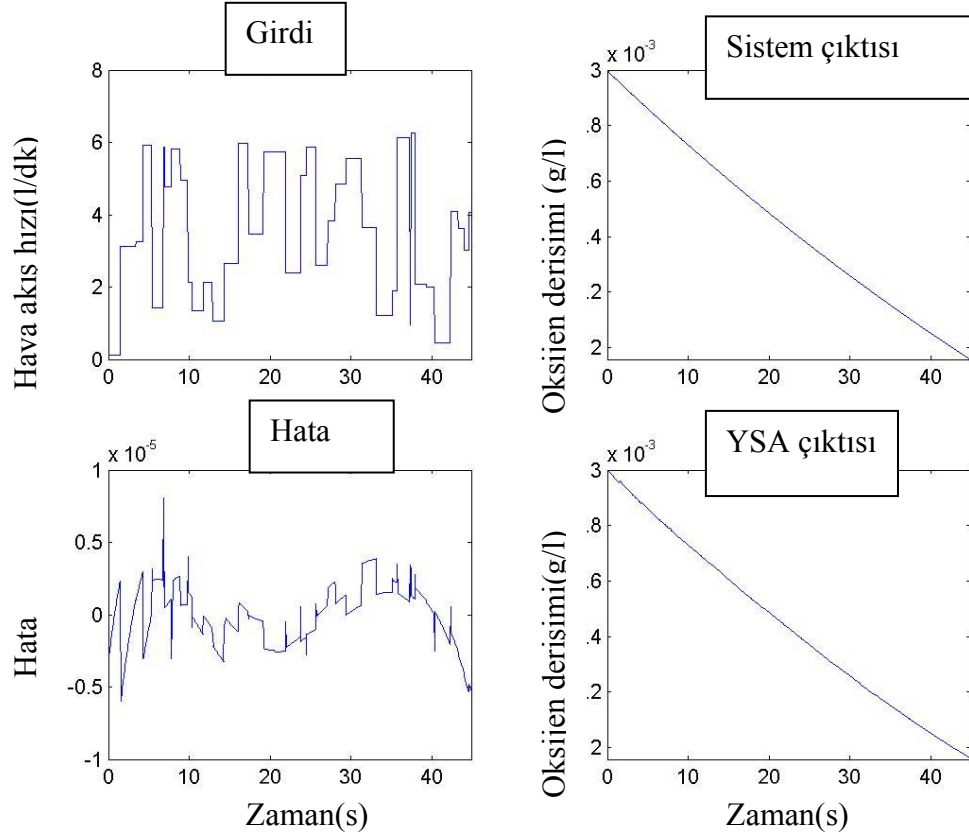
Şekil 4.42 Model çıktısı ve girdisi

Sisteme verilen step etkilere karşı sistemin verdiği çıktı değerleri şekil 4.42’de görüldüğü gibidir. Verilen step etkilere karşı sistemin verdiği çıktılar sürekli azalım gösterilmiştir. Sistemin başlangıç oksijen derişimi 3×10^{-3} g/l ayarlanmıştır. Ayar değişkeni olan hava akış hızı 0-7 l/dk olarak ayarlanmıştır.



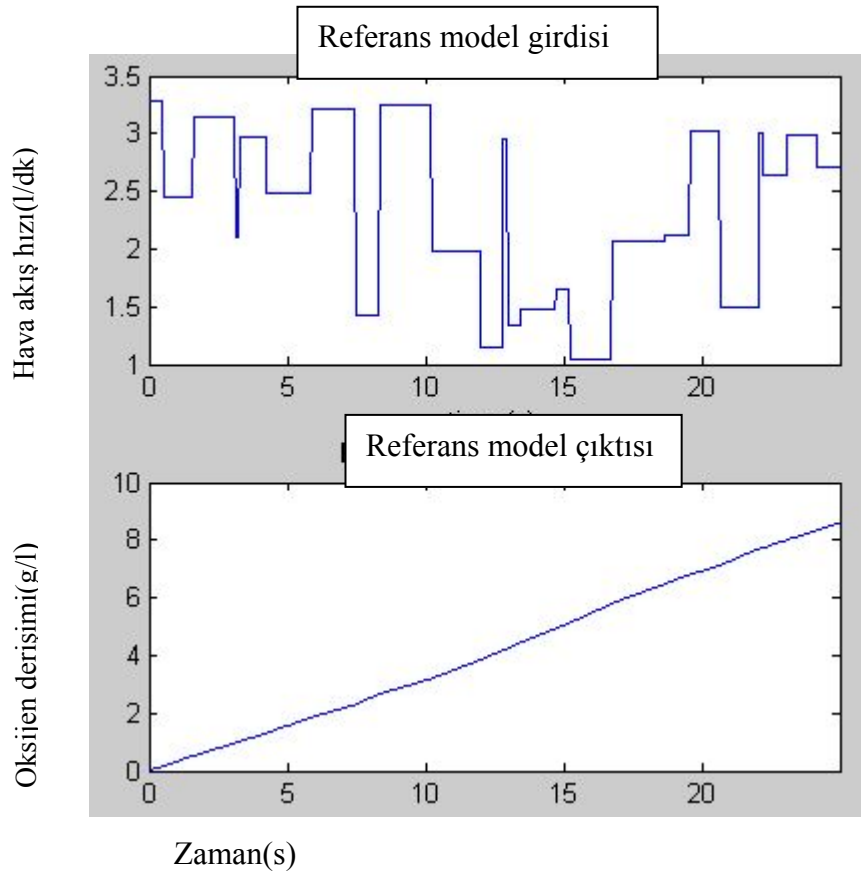
Şekil 4.43 Sistem çıktısı ile model çıktısı arasındaki eğitimde hata değeri

şekil 4.43’de sistem çıktısı ile YSA çıktısı arasındaki hatadan YSA eğitilmiştir. Şekilden de anlaşılacağı gibi sistem ile YSA arasındaki hata (hataların karelerinin toplamının toplam hataya oranı) $3.90908 \cdot 10^{-12}$ bulunmuş ve 100 epochs için hesaplanmıştır. Eğitim için Levenberg Marquardt algoritması kullanılmıştır.



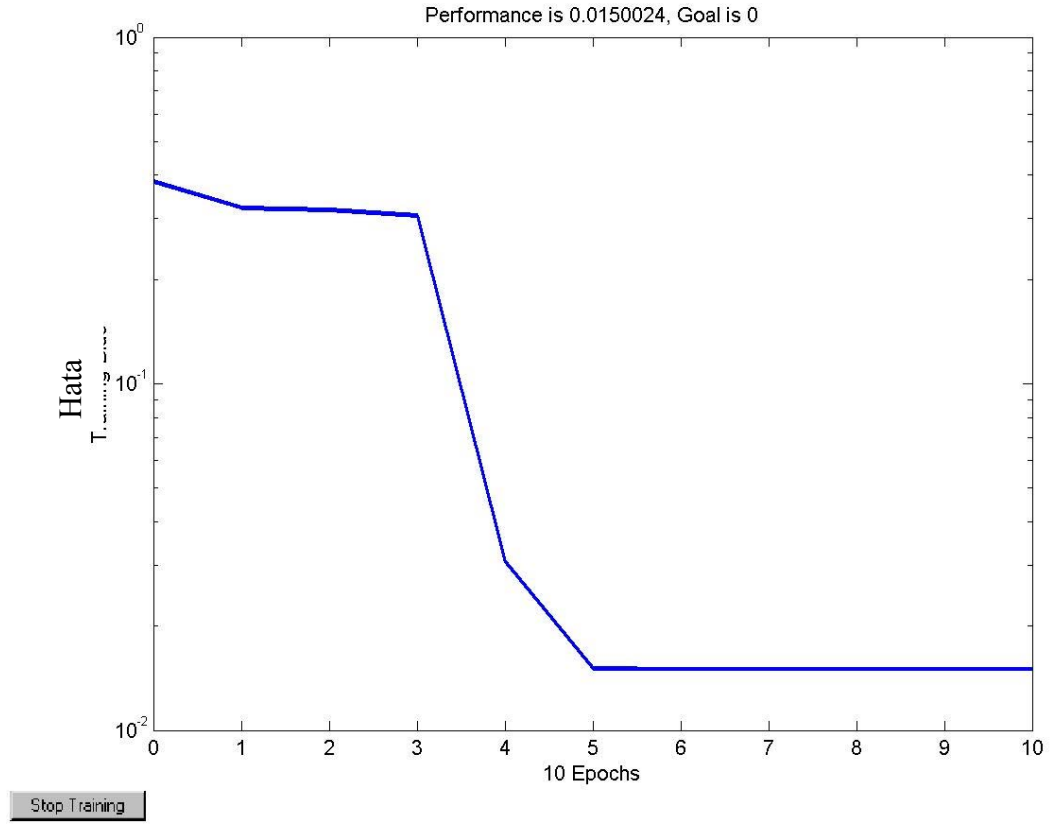
Şekil 4.44 Model girdi ve çıktısı, YSA çıktısı ve YSA çıktısı ile Model çıktısı arasındaki hata

Şekil 4.44 sisteme verilen step etkiler ve sistemin ve step etkilere verdiği çıkış değerleri eğitilen YSA'nın bu step etkilere verdiği yanıt ve sistem ile YSA arasındaki fark değeri gösterilmektedir.



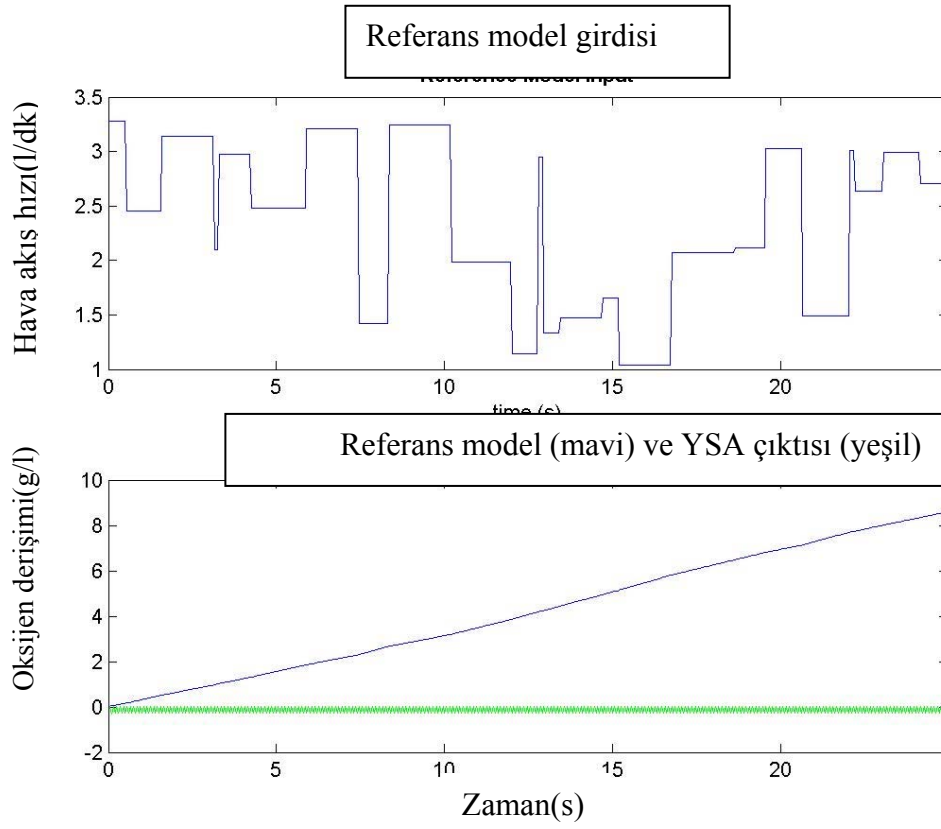
Şekil 4.45 Kontrol edici girdi ve çıktısı

Şekil 4.45’de referans modele verilen step etkilere karşı referans modelin verdiği çıktı değerleri verilmiştir. Verilen step etkilere karşı referans model çıktısı sürekli artış göstermiştir.



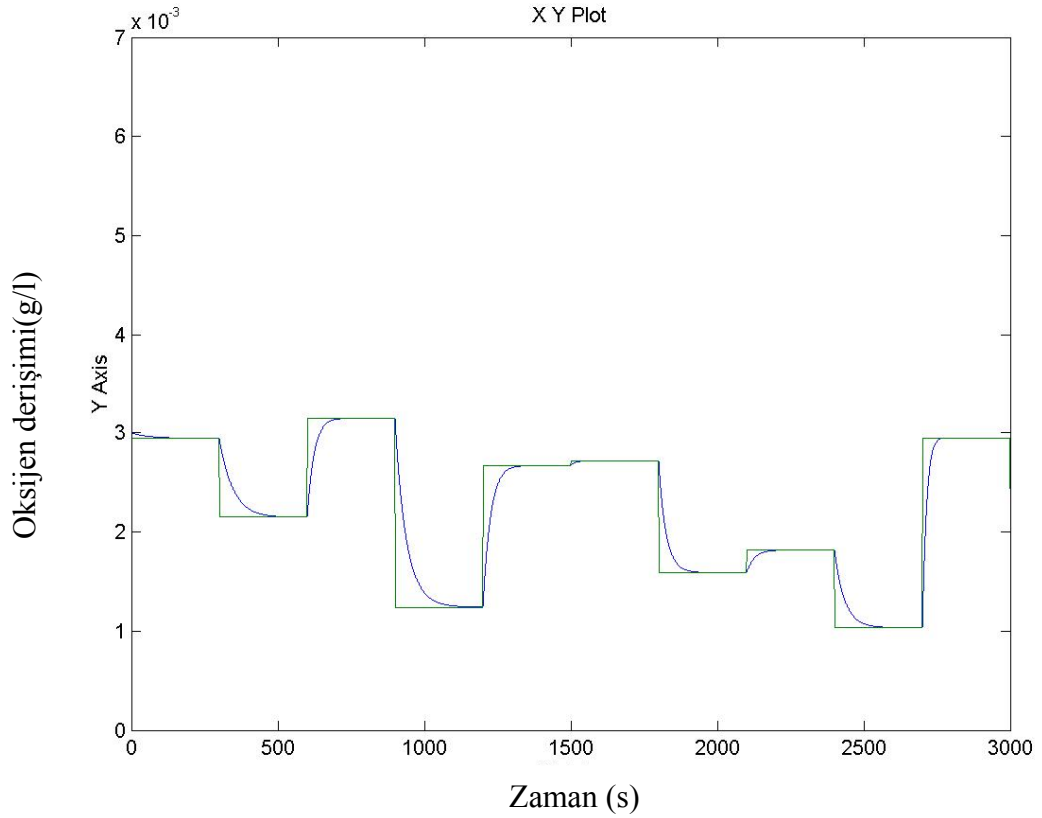
Şekil 4.46 Sistem çıktısı ile referans model arasındaki hata

Şekil 4.46’da sistem çıktısı ile referans model çıktısı hatasını kullanarak kontrol edici eğitilmiş ve eğitimde kullanılan hata değeri (hataların kareleri toplamının hata sayısına bölümü) 0.0150024 bulunmuştur.



Şekil 4.47 Referans model ve model çıktısı

Şekil 4.47’de referans model girdisine verilen step etkilerde sistem çıktısı ve referans model çıktıları verilmiştir. Referans model ile sistem çıktısı arasındaki farktan kontrol edici eğitilmiş, eğitilen kontrol edici kullanılarak sistemin kontrolü şekil 4.48’de gösterildiği gibi başarıyla sağlanmıştır. Küçük set noktası değişimlerinde set noktasını yakalama sürelerinin kısaldığı görülmüştür.

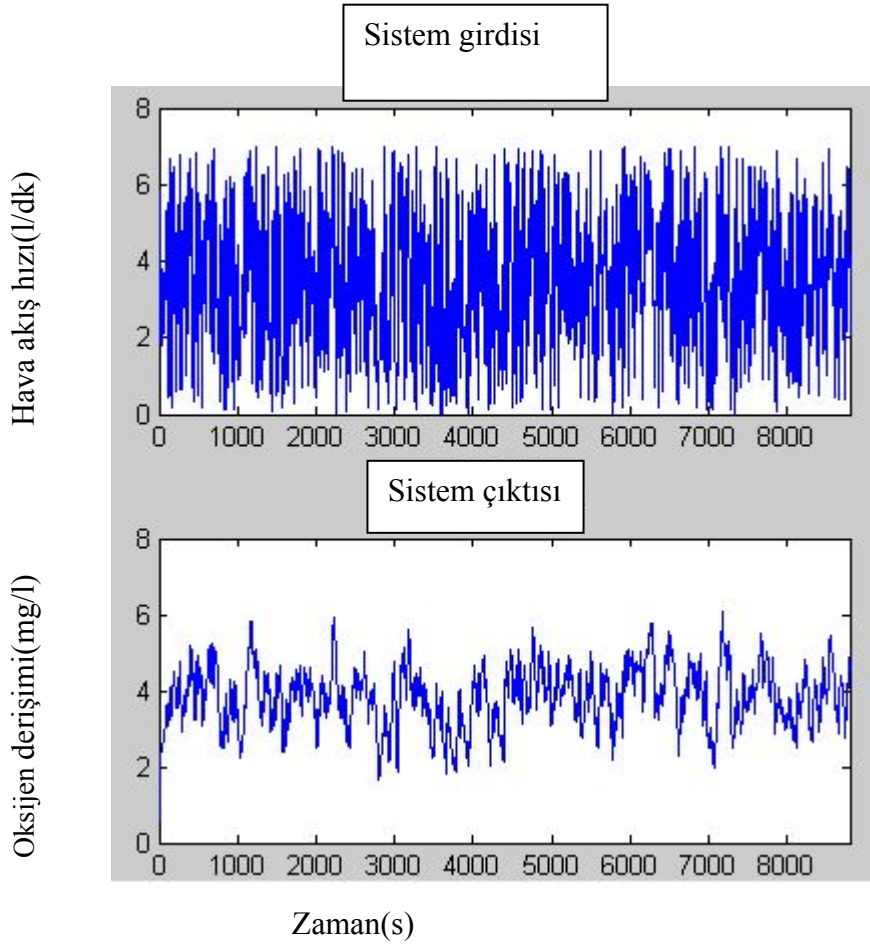


Şekil 4.48 Governing eşitliğinden elde edilen modelin oksijen kontrolü

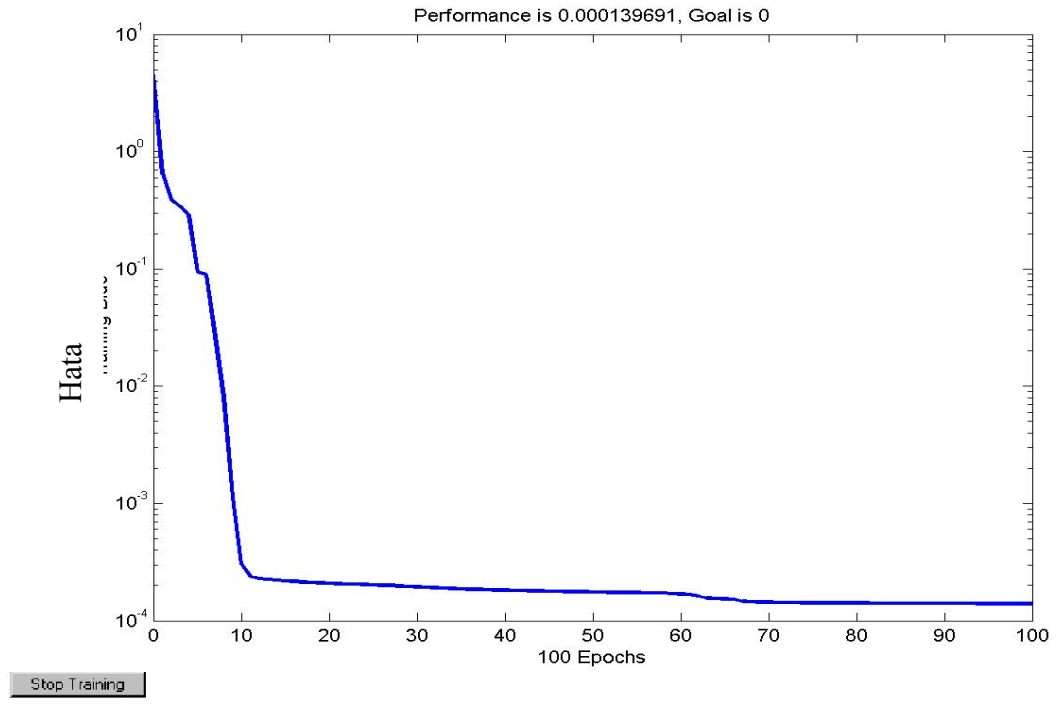
b) ARMAX ile modellenen sistemin kontrolünün sağlanması

ARMAX ile modellenmiş sistemin değişik hava akış hızında çıktısı şekil 4.49’da görülmektedir. Model referanslı algoritmasını matlab toolbox kullanılarak hesaplanmıştır. ARMAX denkleminde ağırların eğitilmesi için 5000 veri elde edilmiştir. Bu kısımda ARMAX modeli kullanılarak sistemin kontrolü sağlanmaya çalışılmıştır fakat uygun referans model bulunamadığından iyi bir kontrol sağlanamamıştır. Sistem çıktısı ile YSA çıktısı arasındaki eğitimde ki hata (hatalarının karelerinin toplamının hata sayısına oranı) şekil 4.50’de gösterilmiştir. Eğitim için Levenberg Marquardt algoritması kullanılmıştır. Şekil 4.51 model çıktısı, ARMAX çıktısı ve YSA çıktısı aralarındaki hata toplu bir şekilde gösterilmektedir. Şekil 4.52’de referans model girdisi ve çıktısı, şekil 4.53’de referans model ile sistem çıktısı arasındaki YSA hata değeri Şekil 4.54 referans model girdi değeri, YSA ve referans model çıktısı ve şekil 4.55

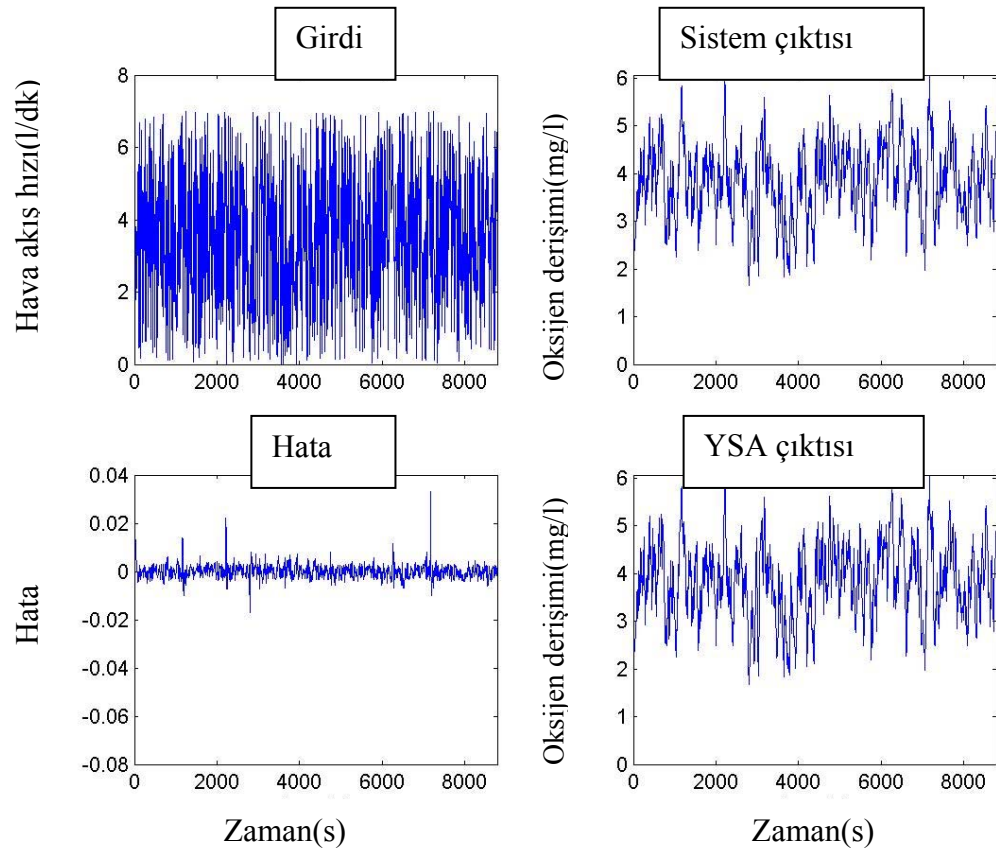
çalışmada kullanılan ARMAX modelinin oksijen kontrolü gösterilmiştir. Uygun Referans model bulunamadığından kontrol tam olarak sağlanamamış, ofset ile sağlanabilmiştir.



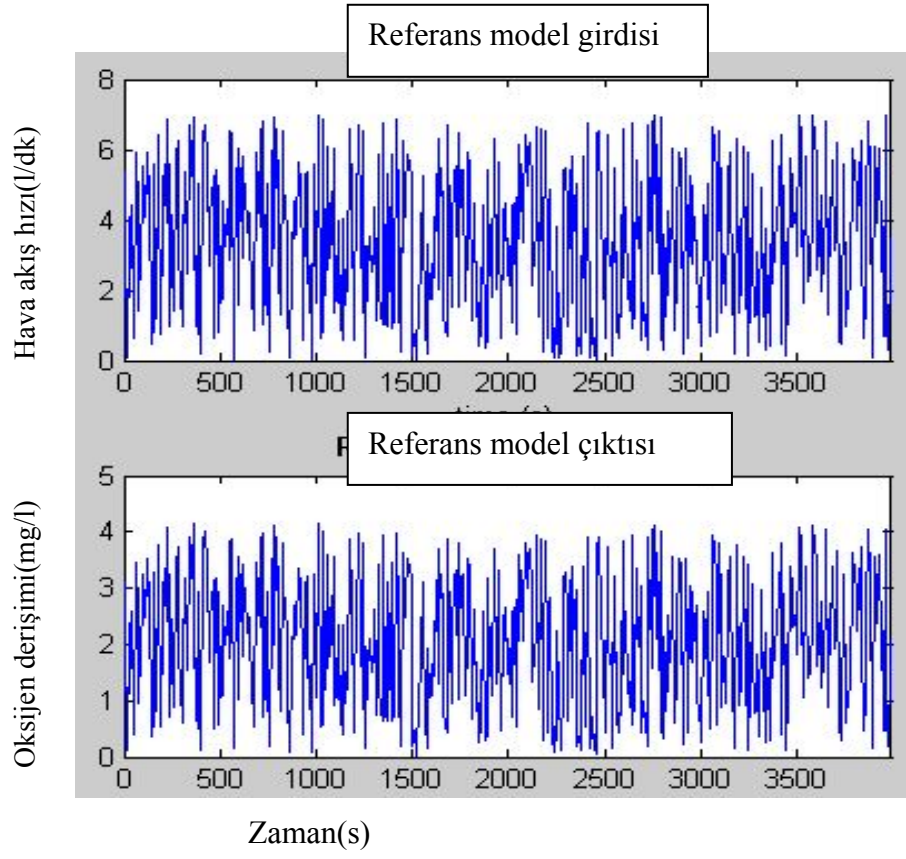
Şekil 4.49 Model girdisi ve çıkışı



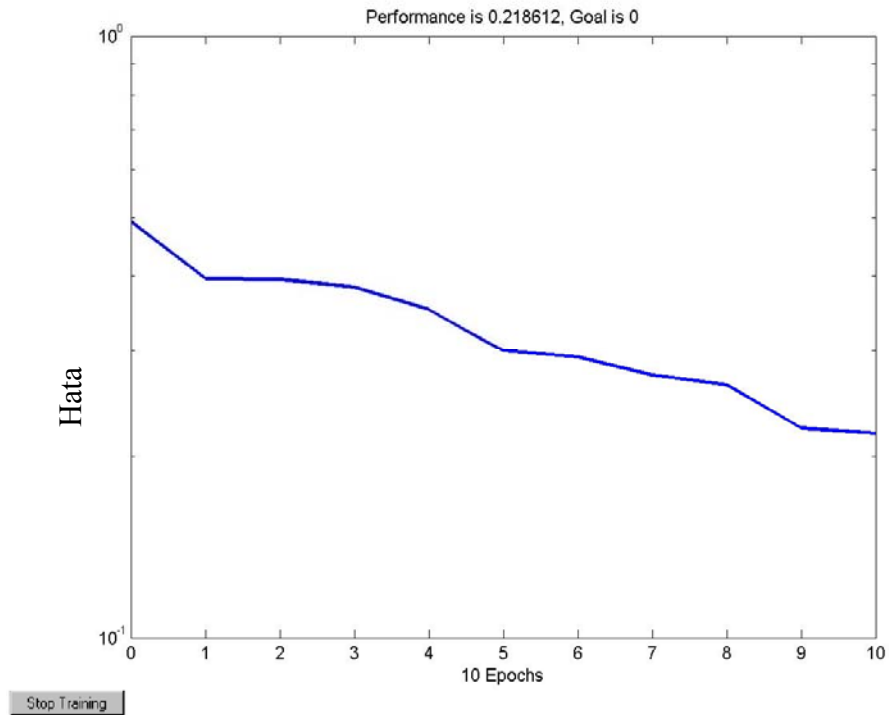
Şekil 4.50 Eğitimdeki hata değeri



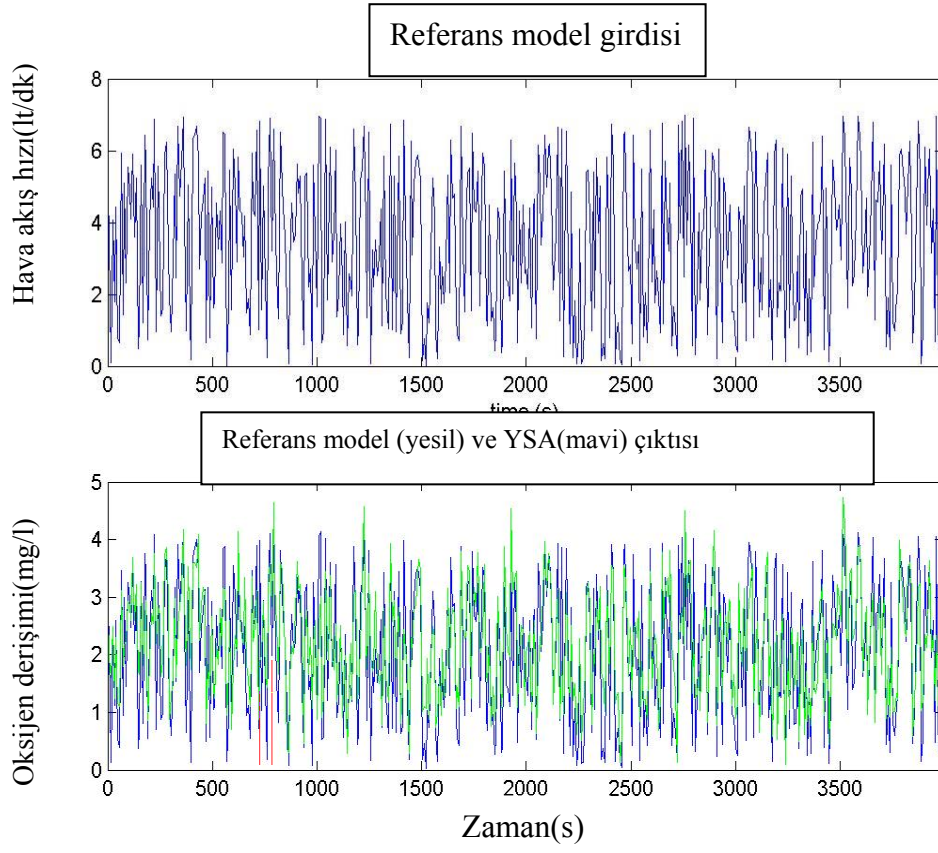
Şekil 4.51 Model girdisi ve çıktısı YSA çıktısı ve model çıktısı ile arasındaki hata



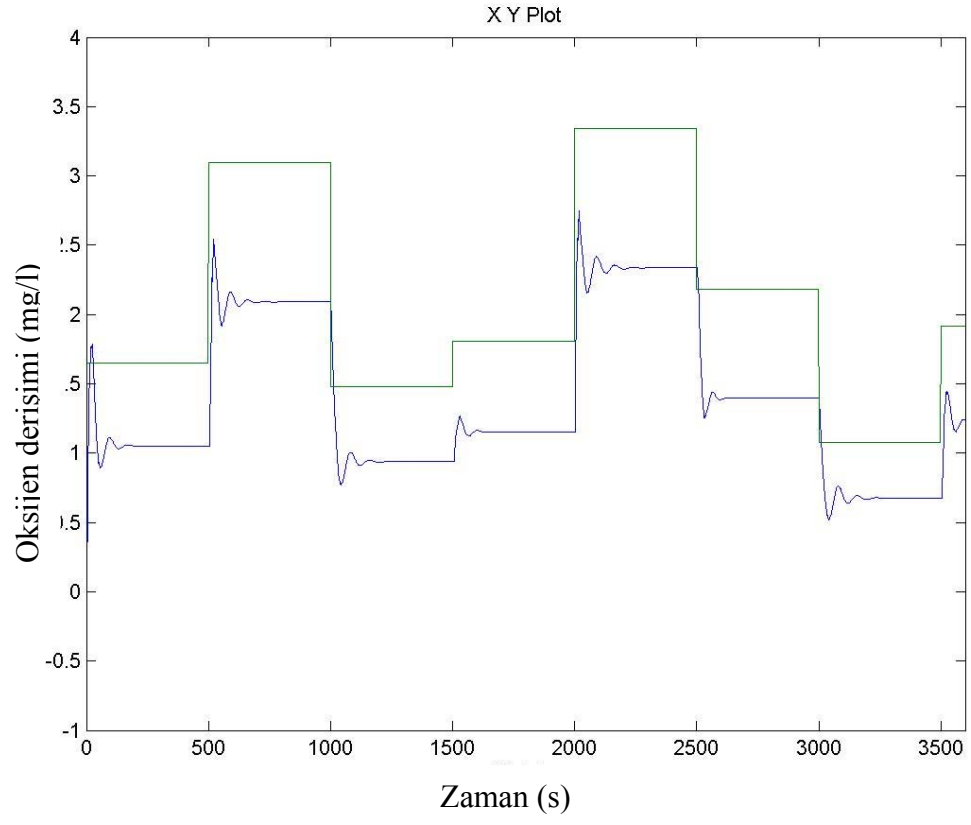
Şekil 4.52 Referans model girdi ve çıktısı



Şekil 4.53 Referans model ile sistem çıktısı arasındaki YSA hata değerleri



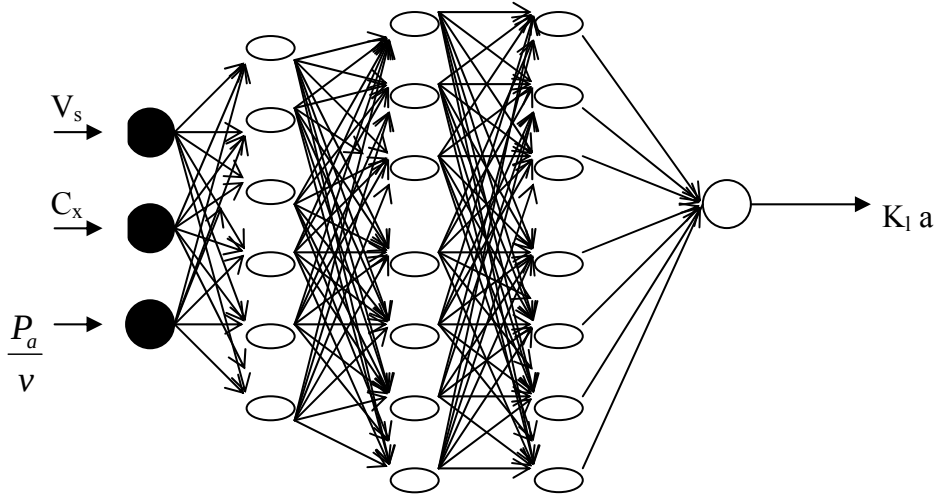
Şekil 4.54 Referans model girdi değeri, YSA ve referans model çıktısı



Şekil 4.55 Çalışmada kullanılan ARMAX modelinin oksijen kontrolü

4.3 K tle Aktarım Katsayısının Yapay Sinir Ađları Tarafından Tahmini

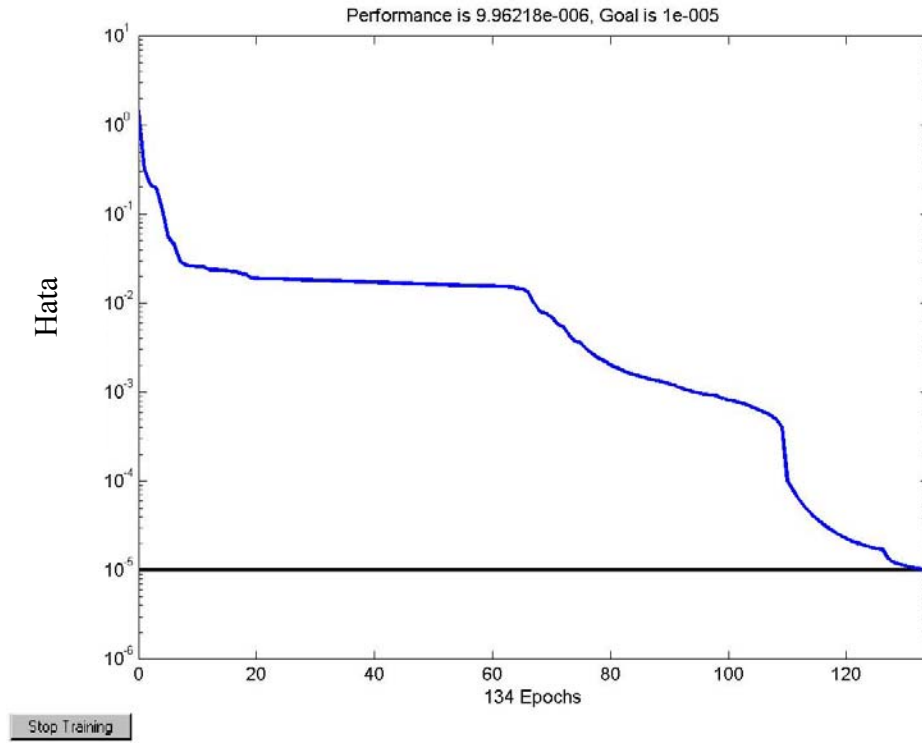
Oksijen k tle aktarım katsayısı biyoreakt r n tasarımı ve iřletme kořullarının belirlenmesinde kullanılan  ok  nemli bir parametredir ve bir  ok fakt r tarafından etkilenir bu y zden hesaplanması gereken bir parametredir. YSA kullanılarak k tle aktarımı tahmininde eřitlik 3.30 kullanılarak k tle aktarım katsayısında kullanılan veriler elde edilmiř (hava akıř hızı, g   katsayısı, mikroorganizma kuru ađırlıđı) ve elde edilen veriler kullanılarak YSA eđitilmiř ve test edilmiřtir. řekil 4.56'da kullanılan YSA yapısı g r lmektedir. Kullanılan bu yapay sinir ađı  ok katmanlı olup, birinci katmanda 6 n ron, ikinci ve    nc  katmanda 7 n ron, son katmanda tek n ron kullanılmıřtır. Birinci katmanda lineer, ikinci ve    nc  katmanda hiperbolik tanjant, son katmanda lineer transfer fonksiyonu kullanılmıřtır. Ađın eđitilmesinde geri besleme ve Lewenberg Marquart teknikleri kullanılmıřtır. řekil 4.57'den anlařılacađı  zere Lewenberg Marquart tekniđinde hata deđer 10⁻⁵ deđerine ulařılmıřtır, řekil 4.58'den geri besleme hata deđer 2*10⁻¹ olduđu g r lmektedir. Lewenberg Marquart y ntemi ile eđitilen ađın eđitimdeki hata deđer daha k   k bulunmaktadır. YSA yapısında ki giriř katmanı isimlendirilirken kullanılmaz. Eđitim i in 86 veri kullanılmıř, test i in 7 veri kullanılmıřtır. İki ayrı YSA oluřturulup birinci ađ Lewenberg Marquart, ikinci ađ geri besleme y ntemiyle eđitilmiřtir. Her iki ađdan da k tle aktarım katsayısı bulunmuřtur. Kullanılan yapay sinir ađlarından ve eřitlik 3.30 den elde edilen kıa  izelge 4.3'de verilmiřtir.  izelge 4.3'den de anlařılacađı  zere YSA ile hesaplanan k tle aktarım katsayısı eřitlik 3.30'dan hesaplanan k tle aktarım katsayısına yakın bulunmuřtur. Buradan g r ld đu  zere YSA k tle aktarım katsayısının hesaplanmasında kullanılabilir bir hesaplama y ntemidir.  alıřmada kullanılan veriler ekler kısmında verilmiřtir.



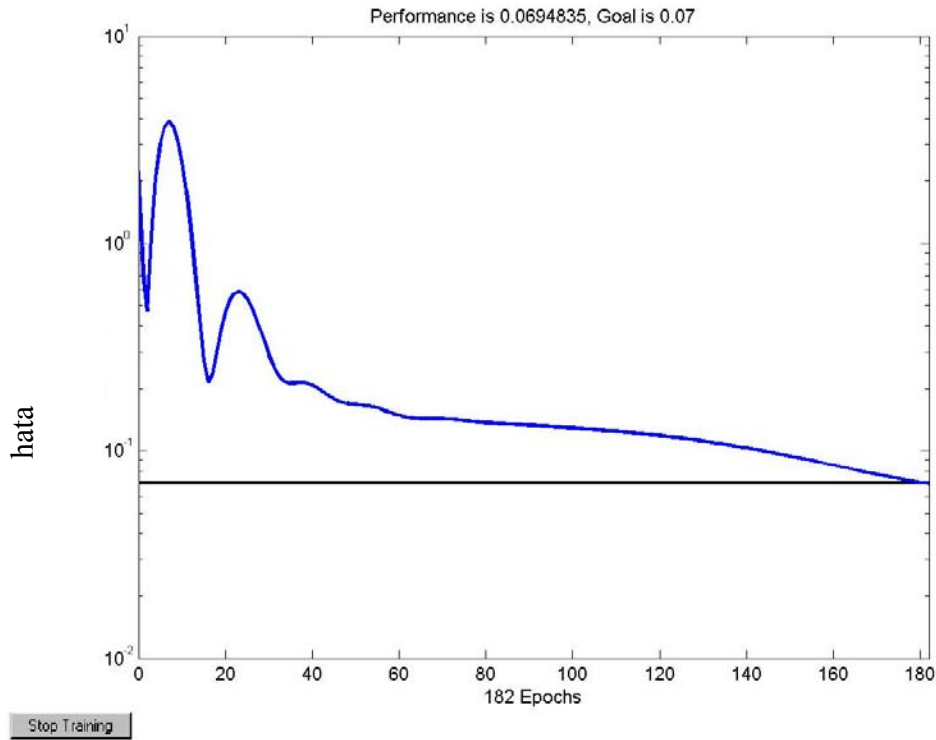
Şekil 4.56 Kütle aktarımın katsayısının tahmininde kullanılan YSA şematik gösterimi

Çizelge 4.3 Denklemden elde edilen k_{1a} ile yapay sinir ağından elde edilen k_{1a} 'ın karşılaştırılması

C_x ($g \cdot l^{-1}$ kuru ağırlık)	V_s ($m \cdot s^{-1}$)	$\frac{P_a}{v}$ ($w \cdot m^{-3}$)	Eşitlik (3.30) elde edilen k_{1a} (s^{-1})	Geri besleme algoritmayla eğitilen ağdan elde edilen k_{1a} (s^{-1})	Levenberg Marquardt algoritmasıyla eğitilen ağdan elde edilen k_{1a} (s^{-1})
127	3	435	1.93	1.97	1.92
130	3	450	1.89	1.98	1.88
132	5	460	2.43	2.43	2.31
133	5	465	2.42	2.43	2.31
135	5	475	2.39	2.52	2.26
136	4	480	2.12	2.10	2.11
148	2	540	1.38	1.38	1.25



Şekil 4.57 Levenberg Marquardt algoritması ile eğitilen ağın hatası



Şekil 4.58 Geri besleme algoritması ile eğitilen ağın hatası

5.TARTIŞMA VE SONUÇ

Yapılan araştırmalarda *S. Cerevisiae* mikroorganizmasının üretiminde sıcaklık, pH ve başlangıç derişiminin etkili olduđu görölmüştür. Bu yüzden farklı sıcaklık ve pH değerinde reaksiyon hızları tahmin edilmiştir. Ađın eğitilmesi için veriler deneysel veriler olup literatürden alınmıştır ve bulunan sonuçlar deneysel verilerle karşılaştırılmıştır. Tahmin edilen reaksiyon hızları, deneysel reaksiyon hızlarına yakın olduđu görölmüştür. Sıcaklığın mikroorganizma çoğalmasına etkisinde beş farklı sıcaklık değeri kullanılmıştır. Bu sıcaklıklar: 27, 30, 32, 35, 38 °C olup deneysel hızlar ile YSA kullanılarak tahmin edilen hızlar karşılaştırılmıştır. Ayrıca pH'ın mikroorganizma çoğalması etkisinde de beş farklı pH değeri kullanılmıştır. Çalışmadaki pH değerleri 3, 4, 5, 6, 7 değerlerindedir. Farklı pH değerinde tahmin edilerek bulunan hız ile deneysel verilerden bulunan hız karşılaştırılmıştır ve YSA hız tahmininde kullanılabileceđi görölmüştür.

Biyoreaktörde ki oksijen derişim kontrolü yine yapay sinir ađları kullanılarak yapılmış olup kontrolde üç farklı kontrol yöntemi kullanılmıştır. Her bir yöntem için oksijen kontrolü sağlanmaya çalışılmıştır. Sistemin oksijen derişimi kontrolü için kullanılan kontrol yöntemleri şunlardır: Tahmine dayalı kontrol, NARMA-L2, model referanslı kontroldür.

Sistem, hem deneysel veriler kullanılarak modellenmiş, hem literatürden alınan taşıyım olaylarına dayalı diferansiyel eşitlikler kullanılarak modellenmiştir. Deneysel olarak sistemin modellenmesi için sisteme PRBS etkisi verilmiştir, sistemin PRBS etkisinde ki çıktıları literatürden alınmıştır (Boyacıođlu 2002). Bu etki kullanılarak sistem ARMAX ile modellenmiştir. Her iki model için üç ayrı kontrol algoritmasıyla sistem kontrol edilmiştir. Mikroorganizmaların uygun sıcaklık ve pH gibi çevre koşulları altında çoğalmaları belirli koşullar altında çeşitli evrelerden oluşmaktadır. Literatürden alınan diferansiyel eşitlik üstsel faz evresi için çıkarılmış olup mikroorganizmanın tüm yaşam evrelerini içermemektedir. Bu yüzden sistem deneysel olarak modellenme yoluna gidilip mikroorganizmanın tüm yaşam evrelerini içeren model çıkarılması yoluna gidilmiştir. ARMAX model sistemin işletim koşulları değışmemek kaydı ile zaman

sınırlaması olmaksızın mayanın tüm üreme fazlarında kullanılırken, taşınım olaylarına dayalı diferansiyel model 4 saat boyunca sadece üstsel üreme fazı için sistemi temsil etmektedir.

Taşınım olaylarına dayalı diferansiyel denklemin doğruluğunu kontrol etmek için biyoreaktörün dinamik özellikleri deneysel ve teorik olarak incelenmiş teorik çalışma ile deneysel çalışmanın birbirine uygunluğu araştırılmıştır. Verilen işletim şartlarında teorik ve deneysel olarak bulunan oksijen derişimi karşılaştırılmıştır. Teorik ve deneysel oksijen derişimi profilinin zamana göre değişimi yaklaşık 4 saat boyunca yakın olduğu görülmüştür. Üstsel üreme fazı için kullanılabileceği sonucuna varılmıştır. Teorik ve deneysel oksijen derişiminin azda olsa farklı olmasının sebebi modelin geliştirilmesindeki varsayımlar ve deneysel hatalardan ileri geldiği kanısına varılmıştır.

Sistemin oksijen kontrolü için ayar değişkeni hava akış hızı seçilmiştir ve hava akış hızı artırılıp, azaltılarak sistemdeki oksijen miktarı ayarlanmaya çalışılmıştır. Ortamda oksijen miktarı belirli bir değerin altına düştüğünde mikroorganizma etil alkol üretmektedir. Mikroorganizmanın maya olarak kullanılması sırasında etil alkol üretmesi istenmez, bu yüzden oksijen kontrolü yapılmıştır. Sisteme fazla miktarda hava göndermek ise sistemin pompalama maliyetini artırmakta. Tahmine dayalı kontrolde bir amaç fonksiyonu önerilmiş bu amaç fonksiyonunu minimum yapan hava akış hızı bulunarak kontrol sağlanmıştır. NARMA-L2 kontrol yönteminde ise doğrusal olmayan dinamik sistemin doğrusallaştırılması ile sistemin kontrolü sağlanmıştır. Model referanslı kontrol de ARMAX denkleminin kontrolünün önerilen referans modeli uygun olmadığından kontrol tam sağlanamamıştır. Kontrolde sistemin referans modeli olarak bir çok referans model denenmiş olup uygun referans model tam olarak bulunamamış, bu yüzden başarılı bir şekilde kontrol sağlanamamıştır. Sistemin referans modeli birinci derece transfer fonksiyonu seçilmiştir. Model referanslı kontrol de taşınım olaylarına dayalı modelin oksijen kontrolünde ise önerilen referans model uygun olup kontrol tam olarak sağlanmıştır.

Biyoreaktörün tasarımında ve işletme koşullarının belirlenmesinde oksijen aktarımı çok önemli bir parametredir. Oksijen aktarımı birçok parametre tarafından etkilenir

(biyoreaktörün şekli ve tipi, ortam bileşimi, ortam derişimi, ortamdaki mikroorganizmanın yapısı, gaz dağıtıcı tipi, sıcaklık, vizkozite). Bu yüzden biyoreaktörde kütle aktarım katsayısı hesaplanması gerekmektedir. Karıştırmalı biyoreaktörde oksijen kütle aktarım katsayısının bulunmasında, YSA'nın uygulanabilirliği gösterilmiştir. Literatürde ki bağıntıyı kullanarak (hava akış hızı, güç katsayısı, mikroorganizma kuru ağırlığı) kütle aktarım katsayıları elde edilmiştir. Bu değerler yapay sinir ağıının eğitiminde kullanılmıştır. Bu eğitilen ağ kullanılarak kütle aktarım katsayısını tahmin edilmiş ve tahmin edilen kütle aktarım katsayısını, literatürden elde edilen kütle aktarım katsayısı ile karşılaştırılmıştır. Ağıın eğitimi için iki ayrı algoritma kullanılmıştır bu algoritmalar şunlardır: Geri besleme ve Lewenberg Marquart algoritması. Lewenberg Marquart algoritması geri beslemeye göre daha hızlı eğitilmiş ve daha düşük hata değerine ulaşılmıştır.

Yukarıda bahsedildiğı gibi bu çalışma üç aşamadan oluşmaktadır: Mikroorganizma Reaksiyon hızlarının tahmini, Kütle aktarım katsayılarının YSA ile tahmini, Çözünmüş oksijen derişimin YSA ile kontrolü. Yapay sinir ağı, diğer klasik yöntemlerle kıyaslandığında aşağıdaki avantajları içermektedir.

- Sonuçların yüksek güvenilirlikte olması sebebi ile geleneksel yöntemlere bir alternatif olarak kullanılabilir.
- Ayrıca model oluşturulması sıradan bazı varsayımlar yapılyorken bu çalışmada deneysel verilerden yararlanıldığı için modelin oluşturulması sırasındaki varsayımlardan doğan hataları içermez.
- Yapay sinir ağıları hızlı olması, basit yapıda olmaları, modellenmesi zor problemlere kabul edilebilen sonuçlar önermesi,
- tahmin edilen değişken hakkında (reaksiyon hızı, oksijen derişimi, $k_L a$) uzun tecrübeye sahip olunmasını gerektirmemesi,

- zamanla ölçüm sayısını en aza indirerek, bu konuda zaman ve para kaybını engellemesi,
- yüksek güvenilirlikte sonuçlar üretmesi, hangi faktörün etkili olduğunu izlememizi sağlayarak, iyileştirme çalışmalarının odak noktasını belirlemede yardımcı olabilir.

Literatürdeki çalışmalardan da anlaşılacağı üzere YSA ile bulunan veriler klasik yöntemlerle bulunan verilere göre daha yüksek doğrulukta yani deneysel veriler daha yakın olduğu görülmüştür. Deneysel sonuç ile yapay sinir ağlarının bulduğu sonuçların birbirlerine çok yakın olduğunu görülmüştür. Buradan yola çıkılarak yapay sinir ağları rahatlıkla reaksiyon hızının belirlenmesinde, sistemin kontrolünde ve kütle aktarımının hesaplanmasında kullanılabilir. Bir kere eğitilen ağ sonuçların doğruluğu test edilerek uygun sonuçlar verdi ise bu ağ defalarca kullanılabilir.

Bütün bunlar göz önüne alındığında, YSA düşük maliyetle uygulanabilecek yöntemdir. Ağı tasarlayan kişinin, modelinin tahmininde doğru tahminler yapabilen bir yapay sinir ağı geliştirdiğinde, elde ettiği ağ yapısına uygun pratik yazılımlar geliştirerek reaktör tasarımında gerekli olan verileri tasarım aşamasında kolaylıkla elde edebilir. Bu nedenle YSA kullanıldığı alanlarda önemli bir kolaylık sağlamıştır.

KAYNAKLAR

- Akay, B. 1998 . Karıştırılmalı kesikli bir biyoreaktörde optimum sıcaklığın doğrusal olmayan model öngörmeli kontrolü. Doktora tezi. Ankara üniversitesi, 257s., Ankara.
- Baş, D. and Boyacı, İ.H. 2006. Modeling and Optimization IV: Inverstigation of Reaction Kinetics and Kinetic Constans Using a Program in Which Artificial Neural Network (ANN) was İntegrated. Journal of Food Engineering, 79, 1152-1158.
- Boyacıoğlu, H. 2002. Kesikli biyoreaktörde çözünmüş oksijen derişiminin genelleştirilmiş minimum varyans kontrolü. Yüksek lisans tezi. Ankara Üniversitesi, 178s.,Ankara.
- Cardello, R. and San, K. 1987. The Design of control for batch bioreactors. Biotechnology and bioengineering. 32, 519-526.
- Demuth, D. and Beale, M. 2001. Matlab Neural Network Toolbox. Version 4, 120s Mathworks.
- Eğrisöğüt, A. 2002. Çimento Ham Madde karışım Prosesinin Yapay sinir ağıları ile Matematiksel modellenmesi, Yüksek Lisans Tezi, Gazi Üni. Fen Bilimleri Enstitüsü, Ankara
- Erdoğan, H., Gülal, E., Akpınar, B. ve Ata, E. 2005, Dinamik sistemlerin tanımlanması, Ankara, Harita ve kadastro mühendisler odası 10. bilimsel kurultayı
- Garcia-Ochoa, F. and Gomez Castrol, E. 2001, Estimation of oxygen mass transfer coefficient in stirred tank reactors using artifical neural networks. Enzyme and microbial technology, 28, 560-569.

- Galaction,A.L., Cascaval, D., Oniscu, C. and Turnea, M. 2004. Prediction of oxygen mass transfer coefficients in stirred bioreactors for bacteria, yeasts and fungus broths . Biochemical engineering journal 20, 85-94.
- Günlü, G. 2004. Yapay sinir ağları ile yüz tanıma, yüksek lisans tezi, Gazi üniversitesi, 112s, Ankara.
- Hang, C.C., Ge,S.S. and Zhang,T. 1998. Nonlinear adaptive control using neural networks and its application to CSTR system. Journal of process control. 9, 313-323.
- Horn, J. 2001. Trajectory tracking of a batch polymerization reactor based on input-output-linearization of a neural process model. Computer and chemical engineering. 25, 1561-1567.
- Kasparian, V. and Batur, C. 1998. Model reference based neural network adaptive Controller. Isa Transaction, 37, 21-31.
- Keleşoğlu, Ö ve Fırat, A. 2006. Tuğla duvardaki ve tesisattaki ısı kaybının yapay sinir ağları ile belirlenmesi, Fırat üniv. Fen ve müh. Bil. Der. 18(1), 133-141
- Molga, E. and Cherbanski, R. 1999. Hybrid First Principle Neural Network Approach to Modelling of the Liquid Reaction System. Chemical Engineering Science,54, 2467-2473.
- Mjalli, F. S. and Al-Asheh, S.2005. Neural Network Based FeedBack Linearizasyon Versus Model Predictive ve Control Of Continuous Alcolic Fermentasyon Proses. Chem. Eng. Technol,28, 1191-1200.
- Özkan, O., Kınacı, C. ve Sağıroğlu, Ş. 2006. Çözünmüş Oksijen derişiminin Yapay sinir ağları ile belirlenmesi: Kızılırmak örneđi. İtü dergisi Mühendislik, 5, 30-38

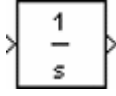
- Terzi, Ö. 2006. Yapay Sinir Ağları Metodu ile Eğridir Gölü Su Sıcaklığının Tahmini. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 10, 297-302
- Ünver, S. 1996. Maya Üretilen Bir Fermentörün Dinamik Matris Kontrolü, Yüksek lisans tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, 110s., Ankara.
- Van can, j. L., Hubert, A.B, Scherpen, M.A and Verbruggen, H.B. 1998. Control of nonlinear chemical processes using neural models and feedback linearization. Computers chem. Eng.22 ,1113-1127.
- Yu, D. L. and Gomm, J.B. 2003.İmpementation of Neural Network Predictive Control to Multivariable Chemical Reactor. Control Engineering Practise,11, 1315-1323.
- Mete, T., Boyacıoğlu, H., Ertunc, S., Akay, B., Özkan, G., Hapoğlu H. and Albaz, M. Definition of Bioreaction Rate With Artifical Neural Network, Somer Symposium Series, 136,2007

EKLER

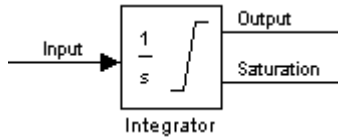
EKLER.....	113
EK 1 Governin Eşitliğinin Similasyonu.....	114
EK 2 Kullanılan Similasyonda ki Bloklarının Görevleri.....	115
EK 3 ARMAX'tan Elde Edilen Modelin Similasyonu.....	117
EK 4 ARMAX Katsayılarının Bulunmasında Kullanılan Matlab Programı	118
EK 5 Kütle Aktarımı Katsayısının Hesaplanması için Oluşturulan Matlab Programı.....	121
EK 6 Farklı Sıcaklıklarda Yapay Sinir Ağları ile Reaksiyon Hızı Tahmininde Kullanılan Matlab Programı.....	122
EK 7 Farklı pH Değerlerinde Yapay Sinir Ağları İle Reaksiyon Hızı Tahmini....	123
EK 8 Farklı sıcaklıklarda yapay sinir ağları ile reaksiyon hızı tahmininde kullanılan matlab programında kullanılan hizgiris.xls, hizcikis.xls verileri.....	124
EK 9 Farklı Ph Değerlerinde Yapay Sinir Ağları İle Reaksiyon Hızı Tahmini Matlab Programında Kullanılan Hiz2giris.Xls Ve Hiz2giris.Xls Verileri.....	126
EK 10 Kütle Aktarımı Katsayısının Hesaplanması İçin Oluşturulan Matlab Programındaki Giris.Xls ve Cikis.Xls Verileri.....	127
EK 11 Kütle Aktarımı, Farklı Sıcaklık ve pH Değerinde Kullanılan Test Verileri.....	128

The diagram illustrates a control system for a chemical process. It features several constant blocks (Constant9, Constant4, Constant5, Constant7, Constant8, Constant10), gain blocks (Gain, Gain1, Gain2), integrator blocks (1/s), product blocks (Product1 through Product9), summing junctions, and a scope (Random Reference). The system is controlled by a feedback loop with a reference input 'oksijen derisimi' and a disturbance input 'hava akis kuz'.

EK 2 Kullanılan Similasyonda ki Bloklarının Görevleri:

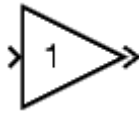


: İntegrator bloğu, giriş değerinin integralini alır

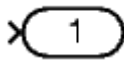


: Sınırlı integrarilini alır. Üç durum söz konusudur bunlar;

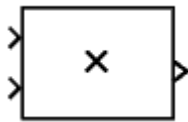
- a) Eğer çıkış değeri doygunluk değerinde aşağı ise değer doygunluk değerine eşit,
- b) Eğer çıkış değeri doygunluk üst limiti ve alt limiti arasında ise değeri integralin çıkış değeridir,
- c) Eğer çıkış değeri doygunluk limitine eşit ve fazla ise doygunluk limiti çıkış olarak alınır.



: Giriş değerini belirli bir sabit değer ile çarpılmasını sağlar.
Çarpılan bu değer scalar, vektörel veya matris olabilir.



: Sistemde çıktıyı temsil eder.



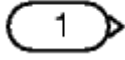
: Bu blok giriş değerinin çarpılmasını veya bölünmesini sağlar



: Bu blok similasyon zamanını gösterir.



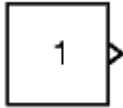
: Bu blok kendisine gelen giriş değerinin toplar veya çıkarır.



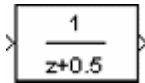
: Giriş değer bloğudur verilen aralıkta bu değer sürekli otomatik olarak değişerek sistemin çıktısına etki eder. Kontrolde ayar değişkenine karşılık gelmektedir.



: Belirli aralıkta rasgele değerler üretir.

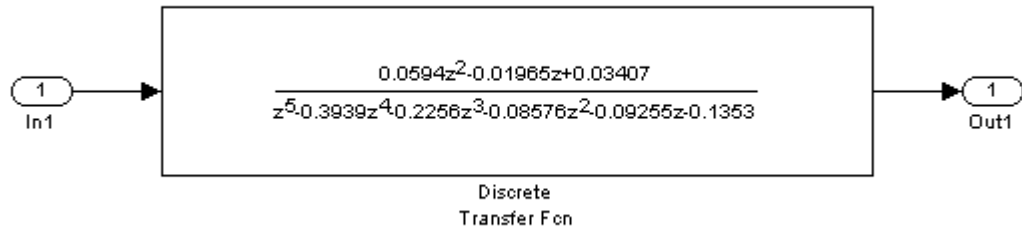


: Similasyona sabit değerlerin girişini sağlar



: Bu blok sistemin Discrete transfer fonksiyonun yazılmasını sağlar.

EK 3 ARMAX'tan Elde Edilen Modelin Similasyonu



ARMAX'tan elde edilen model:

$$A*y=B*u$$

$$Y(t)-0.3939y(t-1)-0.2256y(t-2)-0.08576y(t-3)-0.09255y(t-4)-0.1353y(t-5)=0.0594y(t)-0.01965y(t-1)+0.03407y(t-2)$$

EK 4 ARMAX Katsayılarının Bulunmasında Kullanılan Matlab Programı

```
%-----  
t=A(:,1)  
y=A(:,2)  
u=A(:,3)  
  
plot(t,y)  
axis([0 2100 0 6])  
  
pause  
plot(t,u)  
pause  
axis([0 2100 -2 5])  
pause  
subplot(2,1,1)  
plot(t,y);xlabel('Zaman s');ylabel('cöz');title('P.R.B.S ETKİSİNDEKİ SİSTEM  
ÇIKTISI')  
subplot(2,1,2)  
plot(t,u);xlabel('Zaman s');ylabel('Pompa Akış Hızı');title('P.R.B.S  
ETKİSİNDEKİ SİSTEM ÇIKTISI');pause  
%axis([0 350 2 10])  
pause  
%ze=dtrend(ze);  
ze=[y(1:1068) u(1:1068)];  
zz=[y(1:1068) u(1:1068)];  
  
%---  
%ze=dtrend(ze);  
  
%zz=dtrend(zz);  
%idplot(ze,1:300);title('DTREND SİSTEM ÇIKTISI');pause
```

```

v=arxstruc(ze,zz,struc(2,2,1:10))
%[nn,Vm]=selstruc(v,'AIC')

%v=arxstruc(ze,zz,struc(1:10,1:10,0:10))
pause
nn=selstruc(v,'AIC ')
pause
th4=arx(ze,[5 3 0])
%th4=arx(ze,nn)

pause
xp=(th2zp(th4))
%pause
%th2 = arx(ze,[1 5 1])
compare(zz,th4,inf,1:1068);title('SİSTEM ÇIKTISI İLE ARX MODEL ÇIKTISININ
KARŞILAŞTIRILMASI');pause
e=resid(ze,th4),pause
plot(e),title('Artık'),pause
%ee=rand(1001,1)
%yy=idsim([u ee],th4),title('gul')
%x=[yy u]
%plot(t,yy),title('gürültülü sistem çıktısı'),pause
%compare(ze,yy,inf,1:300);title('transfer gürültüzz arx')

am2=armax(ze,[5 3 0 0])
pause
compare(ze,am2,inf,1:400);title('SİSTEM ÇIKTISI İLE ARMAX MODEL
ÇIKTISININ KARŞILAŞTIRILMASI');pause
%pause;compare(am2,th4,inf,1:300);title('ARMAX İLE ARX MODEL ÇIKTISININ
KARŞILAŞTIRILMASI');pause

```

```

%am2=arimax(x,[1 1 1 1])
pause
xpam2=th2zp(am2),
zpplot([xp xpam2]),title('arx armax kutupları'),pause

e1=resid(ze,am2),pause
plot([e e1]),title('arx ve armax artıkları'),pause
%yham2=idsim(y,am2)
%yharx=idsim(y,th4)
%subplot(211),plot([y yham2]),
%subplot(212),plot([y yharx]),

thm2=rplr(ze,[1 5 1 0 0 1],'ff',0.98)
plot(thm2),pause
thm3=rplr(ze,[1 5 0 0 0 1],'ff',0.98)
plot(thm3),pause

```

EK 5 Kütle Sktarımı Katsayısının Hesaplanması İçin Oluşturulan Matlab Programı

```
p=transpose(xlsread('giris.xls'));
t=transpose(xlsread('cikis.xls'));
net=newff(minmax(p),[6,7,7,1],{'purelin','tansig','tansig','purelin'},'traingdm');
% Levenberg-Marquardt algoritması için 'traingdm' yerine 'trainlm' yazılır.
net.trainparam.show=10;
net.trainparam.lr=0.05;
net.trainparam.mc=0.9;
net.trainparam.epochs=2000;
net.trainparam.goal=5*10^-5; % Levenberg-Marquardt algoritması için hata değeri
% net.trainparam.goal=0.07; geri besleme hata hedef değeri için yüzde işareti kaldırılıp
% yerine net.trainparam.goal=5*10^-5 önüne yüzde işareti konur.
[p_normal,minp,maxp,t_normal,mint,maxt]=premnmx(p,t);
net=train(net,p_normal,t_normal);
p_test=transpose(xlsread('giristest.xls'));
t_test=transpose(xlsread('cikistest.xls'));
p_testnormal=tramnmx(p_test,minp,maxp);
a_normal=sim(net,p_test);
a=postmnmx(a_normal,mint,maxt);
deneme=transpose(xlsread('giristest.xls'));
dnormal=tramnmx(deneme,minp,maxp);
similli=sim(net,dnormal);
cikis=postmnmx(similli,mint,maxt)
```

EK 6 Farklı Sıcaklıklarda Yapay Sinir Ağları İle Reaksiyon Hızı Tahmininde Kullanılan Matlab Programı

```
p=transpose(xlsread('hizgiris.xls'));
t=transpose(xlsread('hizcikis.xls'));
net=newff(minmax(p),[10,12,1],{'purelin','tansig','tansig'},'trainlm');
net.trainparam.show=10;
net.trainparam.lr=0.05;
net.trainparam.mc=0.9;
net.trainparam.epochs=2000;
net.trainparam.goal=0.005;
[p_normal,minp,maxp,t_normal,mint,maxt]=premnmx(p,t);
net=train(net,p_normal,t_normal);
p_test=transpose(xlsread('hizgiristest.xls'));
t_test=transpose(xlsread('hizcikistest.xls'));
p_testnormal=tramnmx(p_test,minp,maxp);
a_normal=sim(net,p_test);
a=postmnmx(a_normal,mint,maxt);
deneme=transpose(xlsread('hizgiristest.xls'));
dnormal=tramnmx(deneme,minp,maxp);
similli=sim(net,dnormal);
cikis=postmnmx(similli,mint,maxt)
```

EK 7 Farklı pH Değerlerinde Yapay Sinir Ağları İle Reaksiyon Hızı Tahmini Matlab Programı

```
p=transpose(xlsread('hiz2giris.xls'));
t=transpose(xlsread('hiz2cikis.xls'));
net=newff(minmax(p),[12,13,1],{'purelin','tansig','tansig'},'trainlm');
net.trainparam.show=10;
net.trainparam.lr=0.05;
net.trainparam.mc=0.9;
net.trainparam.epochs=2000;
net.trainparam.goal=0.001;
[p_normal,minp,maxp,t_normal,mint,maxt]=premnmx(p,t);
net=train(net,p_normal,t_normal);
p_test=transpose(xlsread('hiz2girisest.xls'));
t_test=transpose(xlsread('hiz2cikistest.xls'));
p_testnormal=tramnmx(p_test,minp,maxp);
a_normal=sim(net,p_test);
a=postmnmx(a_normal,mint,maxt);
deneme=transpose(xlsread('hiz2girisest.xls'));
dnormal=tramnmx(deneme,minp,maxp);
similli=sim(net,dnormal);
cikis=postmnmx(similli,mint,maxt)
```


EK 8 Farklı Sıcaklıklarda Yapay Sinir Ağları İle Reaksiyon Hızı Tahmininde Kullanılan Matlab Programında Kullanılan “hizgiris.xls”, “hizcikis.xls” Verileri

Veri no	Sıcaklık (°c)	Başlangıç derişimi (g/l)	Derişim (g/l)	Deneysel hız (g*l ⁻¹ /s)	Veri no	Sıcaklık (°c)	Başlangıç derişimi (g/l)	Derişim (g/l)	Deneysel Hız (g*l ⁻¹ /s)
1	27	1	1	0,5066	46	35	1	2,4	1,0962
2	27	1	0,99	0,5303	47	35	1	3,1	1,0104
3	27	1	1	0,554	48	35	1	4	0,9246
4	27	1	1,1	0,5777	49	35	1	4,9	0,8388
5	27	1	1,2	0,6014	50	35	1	5,1	0,753
6	32	1,1	1,1	1,436	51	27	1	4	0,7436
7	32	1,1	1	1,5032	52	27	1	4,2	0,7673
8	32	1,1	1	1,5704	53	27	1	4,8	0,791
9	32	1,1	1,3	1,6376	54	27	1	5,2	0,8147
10	32	1,1	1,8	1,7048	55	27	1	5,4	0,8384
11	30	1	1	1,2371	56	38	1	3,1	0,5875
12	30	1	0,8	1,2843	57	38	1	3,2	0,5933
13	30	1	1	1,3315	58	38	1	4,1	0,5991
14	30	1	1,2	1,3787	59	38	1	4,3	0,6049
15	30	1	1,3	1,4259	60	38	1	4,5	0,6107
16	35	1	1	1,5252	61	35	1	5,5	0,6672
17	35	1	0,7	1,4394	62	35	1	6	0,5814
18	35	1	0,8	1,3536	63	35	1	6,3	0,4956
19	35	1	1,1	1,2678	64	35	1	6,4	0,4049
20	35	1	1,5	1,182	65	35	1	6,5	0,324
21	38	0,7	0,7	0,5295	66	32	1,1	6	2,108
22	38	0,7	0,5	0,5353	67	32	1,1	6,3	2,1752
23	38	0,7	0,7	0,5411	68	32	1,1	6,5	2,2424
24	38	0,7	0,8	0,5469	69	32	1,1	6,9	2,3096
25	38	0,7	1	0,5527	70	32	1,1	7	2,3768
26	27	1	1,3	0,6251	71	30	1	5,4	1,7091
27	27	1	1,6	0,6488	72	30	1	5,8	1,7563
28	27	1	2	0,6725	73	30	1	6,1	1,8035
29	27	1	2,8	0,6962	74	30	1	6,5	1,8507
30	27	1	3,2	0,7199	75	30	1	6,8	1,8979
31	30	1	2	1,4731	76	27	1	5,8	0,8621
32	30	1	2,3	1,5203	77	32	1,1	6,9	2,444
33	30	1	3,6	1,5675	78	30	1	6,7	1,9451
34	30	1	4	1,6147	79	35	1	6,4	0,2382
35	30	1	4,5	1,6619	80	38	0,7	4,6	0,6165
36	38	0,7	1,1	0,5585	81	32	1,1	7	2,5122
37	38	0,7	1,2	0,5643	82	35	1	6,1	0,1524
38	38	0,7	1,8	0,5701	83	38	0,7	4,8	0,6223
39	38	0,7	2,1	0,5759	84	30	1	6,7	1,9923
40	38	0,7	2,8	0,5817	85	27	1	6,1	0,8858
41	32	1,1	2,5	1,772	86	30	1	6,6	2,03995
42	32	1,1	3,2	1,8392	87	35	1	6,2	0,666

43	32	1,1	4	1,9064	88	32	1,1	7	2,5784
44	32	1,1	4,8	1,9736	89	27	1	6,2	0,9095
45	32	1,1	5,3	2,0408	90	38	0,7	4,81	0,6281

**EK 9 Farklı pH Değerlerinde Yapay Sinir Ağları ile
Reaksiyon Hızı Tahmini Matlab Programında Kullanılan
“hiz2giris.xls” ve “hiz2giris.xls” Verileri**

Veri no	pH	Derişim (g/l)	Başlangıç Derişim i(g/l)	Deneysel hız (g ⁻¹ l ⁻¹ /s)	Veri no	pH	Derişim (g/l)	Başlangıç Derişimi (g/l)	Deneysel hız (g ⁻¹ l ⁻¹ /s)
1	3	0,4	0,4	0,529	44	2	1,3	0,8	2,8314
2	3	0,5	0,4	0,3086	45	3	1,4	0,4	3,9418
3	3	0,6	0,4	0,2052	46	7	1,16	0,8	1,16
4	3	0,7	0,4	0,2329	47	7	1,22	0,8	1,22
5	3	0,8	0,4	0,4058	48	7	2,9	0,8	1,199
6	4	0,9	0,8	-1,082	49	7	3,9	0,8	1,09
7	4	1	0,8	-0,1257	50	7	4,7	0,8	0,9249
8	4	1,1	0,8	0,5739	51	4	3,3	0,8	0,4955
9	4	1,3	0,8	1,046	52	4	3,35	0,8	0,2217
10	4	1,5	0,8	1,32	53	4	3,36	0,8	-0,0116
11	5	0,9	0,8	-1,082	54	4	3,39	0,8	-0,1749
12	5	1	0,8	-0,1257	55	4	3,4	0,8	-0,2385
13	5	1,2	0,8	0,5739	56	6	5,1	0,8	0,5757
14	5	1,4	0,8	1,04	57	6	5,3	0,8	0,3223
15	5	1,9	0,8	1,322	58	6	5,31	0,8	0,0844
16	6	0,9	0,8	-1,0568	59	6	5,2	0,8	-0,1152
17	6	1	0,8	-0,2106	60	6	5,11	0,8	-0,2539
18	6	1,1	0,8	0,4429	61	3	1,8	0,4	5,282
19	6	1,3	0,8	0,8666	62	3	2,1	0,4	6,8661
20	6	1,6	0,8	1,1436	63	3	2,3	0,4	8,7082
21	7	0,9	0,8	-1,2011	64	3	2,8	0,4	10,8224
22	7	1	0,8	-0,4014	65	3	3	0,4	13,2228
23	7	1,1	0,8	0,2174	66	5	5,5	0,8	0,4955
24	7	1,2	0,8	0,6726	67	5	5,6	0,8	0,2217
25	7	1,3	0,8	0,9813	68	5	5,61	0,8	-0,0116
26	5	2,3	0,8	1,43	69	5	5,51	0,8	-0,1749
27	5	2,1	0,8	1,399	70	5	5,52	0,8	-0,2385
28	5	4,1	0,8	1,26	71	7	4,9	0,8	0,7134
29	5	4,9	0,8	1,04	72	7	5,1	0,8	0,475
30	5	5,2	0,8	0,779	73	7	5,11	0,8	0,2269
31	6	2,1	0,8	1,2763	74	7	5,01	0,8	-0,01388
32	6	2,8	0,8	1,2877	75	7	5,02	0,8	-0,2302
33	6	3,4	0,8	1,2					
34	6	4,3	0,8	1,03					
35	6	4,9	0,8	0,8277					
36	4	1,9	0,8	1,429					
37	4	2,5	0,8	1,3999					
38	4	2,8	0,8	1,2677					
39	4	3,1	0,8	1,0452					
40	4	3,2	0,8	0,7798					
41	3	0,9	0,4	0,738					
42	3	0,98	0,4	1,2436					
43	3	1,1	0,4	1,9367					

EK 10 Kütle Aktarımı Katsayısının Hesaplanması için Oluşturulan Matlab

Programındaki “giris.xls” ve “cikis.xls” Verileri

Veri no	C _x (g/l kuru ağırlık)	Hava akış hızı(m/s)	Güç katsayısı (w*m ⁻³)	k _{1a} (s ⁻¹)	Veri no	C _x (g/l kuru ağırlık)	Hava akış hızı(m/s)	Güç katsayısı (w*m ⁻³)	k _{1a} (s ⁻¹)
1	138	4	490	2,09	44	79	4	195	3,33
2	134	4	470	2,14	45	64	2	120	2,8
3	140	5	500	2,32	46	68	2	35	2,65
4	150	1	550	0,96	47	77	1	185	1,66
5	144	3	520	1,74	48	76	2	180	2,41
6	149	1	545	0,96	49	72	5	160	4,04
7	141	5	505	2,31	50	67	4	135	3,84
8	143	4	515	2,03	51	66	3	130	3,36
9	96	4	280	2,82	52	63	3	115	3,5
10	93	1	265	1,42	53	62	5	110	4,62
11	92	2	260	2,05	54	61	4	105	4,18
12	130	3	450	1,89	55	98	2	290	1,94
13	125	1	425	1,11	56	97	4	285	2,8
14	120	1	400	1,15	57	81	4	205	3,25
15	110	2	350	1,76	58	82	3	210	2,78
16	105	3	325	2,26	59	83	2	215	2,23
17	102	4	310	2,68	60	89	5	245	3,37
18	129	2	445	5,19	61	49	2	45	3,64
19	124	2	420	1,6	62	54	5	70	5,27
20	119	1	395	1,16	63	55	4	75	4,61
21	128	1	440	1,09	64	56	2	80	3,17
22	114	4	370	2,45	65	57	3	85	3,84
23	101	3	305	2,33	66	58	4	90	4,38
24	108	2	340	1,79	67	59	2	95	3,02
25	107	1	335	1,26	68	60	1	100	2,08
26	106	2	330	1,82	69	70	1	150	1,81
27	126	2	430	1,58	70	75	5	175	3,9
28	121	3	405	2,014	71	80	5	200	3,69
29	131	4	455	2,18	72	84	2	220	2,21
30	123	3	415	1,98	73	86	3	230	2,67
31	139	5	495	2,33	74	87	4	235	3,06
32	45	1	25	2,835	75	47	2	35	3,82
33	50	1	50	2,497	76	49	2	45	3,64
34	47	2	35	3,82	77	52	3	60	4,21
35	53	2	65	3,35	78	54	5	70	5,27
36	70	1	150	1,81	79	51	2	55	3,49
37	65	1	125	1,93	80	64	2	120	2,8
38	46	1	30	2,75	81	69	3	145	3,23
39	48	3	40	4,59	82	99	1	295	1,35
40	78	2	190	2,35	83	94	3	270	2,48
41	71	2	155	2,55	84	90	5	250	3,34
42	74	3	170	3,03	85	85	1	225	1,53
43	73	4	165	3,56	85	82	3	210	2,78

EK 11 Kütle Aktarımı, Farklı Sıcaklık ve pH Değerinde Kullanılan Test Verileri

Farklı sıcaklık için test verileri ve YSA verileri

Sıcaklık($^{\circ}\text{C}$)	Başlangıç derişimi(g/l)	Derişim (g/l)	Deneysel hız ($\text{g}^{\circ}\text{I}^{-1}/\text{s}$)	YSA tarafından hesaplanan hız ($\text{g}^{\circ}\text{I}^{-1}/\text{s}$)
27	1	1,4385	0,6562	0,6651
30	1	1,14285	0,7968	0,7296
38	0,7	4,2024	0,6372	0,6347
27	1	3,5022	1,0288	0,9755
32	1,1	7,2655	0,2575	0,3368
32	1,1	1,6239	0,8846	0,8357
32	1,1	5,6417	1,095	1,166
27	1	5,2253	0,8409	0,8211
32	1,1	6,684	0,7691	0,7649

Farklı pH için test verileri ve YSA verileri

pH	Derişim(g/l)	Başlangıç derişimi(g/l)	Deneysel hız ($\text{g}^{\circ}\text{I}^{-1}/\text{s}$)	hesaplanan hız ($\text{g}^{\circ}\text{I}^{-1}/\text{s}$)
3	0,7	0,4	0,2329	0,3282
3	1,8	0,4	5,282	5,3704
3	3	0,4	13,2228	12,7532
4	3,36	0,8	0	0,2444
5	2,8	0,8	1,43	1,4377
5	5,5	0,8	0,4955	0,2049
6	4,3	0,8	1,03	1,1384
6	5,31	0,8	0,0844	0,2183
7	1,3	0,8	0,9813	0,8433
7	2,9	0,8	1,199	1,1476

Kütle aktarım katsayısı içinn test verileri ve YSA verileri

$C_x(\text{g/l})$ kuru ağırlık)	Hava akış hızı(m/s)	Güç katsayısı($\text{W}^{\circ}\text{m}^{-3}$)	k_a (s^{-1})	Lewenberg Marquard algoritması ile eğitilen ağdan elde edilen k_a (s^{-1})	Geri beslemeli ile ağdan elde edilen k_a (s^{-1})
127	3	435	1,93	1,92	1,97
130	3	450	1,89	1,88	1,98
132	5	460	2,43	2,31	2,43
133	5	465	2,42	2,31	2,43
135	5	475	2,39	2,26	2,52
136	4	480	2,12	2,11	2,10
148	2	540	1,38	1,25	1,38

ÖZGEÇMİŞ

Adı Soyadı : Tufan METE
Doğum Yeri : Gelibolu
Doğum Tarihi : 04.10.1982
Medeni hali : Bekar
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Gelibolu Lisesi (2000)

Lisans : Ankara Üniversitesi Mühendislik Fakültesi Kimya Mühendisliği
Bölümü (2005)

Yüksek Lisans: Ankara Üniversitesi Fen bilimleri Enstitüsü Kimya Mühendisliği
Anabilim Dalı

Yayınları (SCI ve diğer)

Mete, T., Boyacıoğlu, H., Ertunc, S., Akay, B., Özkan, G., Hapoğlu H. and Alpbaz, M.
Definition of Bioreaction Rate With Artifical Neural Network, Somer
Symposium Series, 136,2007