**T.C.**
**ERCİYES ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**
**MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

# YAPAY SİNİR AĞLARI KULLANARAK KOT İŞLEME ROBOTLARINDA YÖRÜNGE ANALİZİ

**Hazırlayan**
**Kürşat YILDIRIM**

**Danışman**
**Doktor Öğretim Üyesi**
**Mehmet Bahadır ÇETİNKAYA**

**Yüksek Lisans Tezi**

**Haziran 2021**

**KAYSERİ**

**T.C.**
**ERCİYES ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**
**MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

# YAPAY SİNİR AĞLARI KULLANARAK KOT İŞLEME ROBOTLARINDA YÖRÜNGE ANALİZİ

**(Yüksek Lisans Tezi)**

**Hazırlayan**
**Kürşat YILDIRIM**

**Danışman**
**Doktor Öğretim Üyesi**
**Mehmet Bahadır ÇETİNKAYA**

**Haziran 2021**

**KAYSERİ**

# BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Kürşat YILDIRIM

**“Yapay Sinir Ağları Kullanarak Kot İşleme Robotlarında Yörünge Analizi”** adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi ’ne uygun olarak hazırlanmıştır.

| **Tezi Hazırlayan** | **Danışman** |
| --- | --- |
| Kürşat YILDIRIM | Doktor Öğretim Üyesi<br>Mehmet Bahadır ÇETİNKAYA |

**Mekatronik Mühendisliği ABD Başkanı**

Prof. Dr. Şahin YILDIRIM

# TEŞEKKÜR

Çalışmalarım boyunca bilgi, tecrübe ve önerileriyle beni aydınlatan, rehberlik eden, ilgi ve yardımlarını esirgemeyen sayın hocam Doktor Öğretim Üyesi Mehmet Bahadır ÇETİNKAYA ‘ya teşekkürlerimi sunarım.

Çalışmalarım sırasında karşılaştığım zorlukları aşmamda yardımlarından ve katkılarından dolayı mesai arkadaşlarıma teşekkür ederim.

Beni yüksek lisans yapmam için teşvik eden ve hayatım boyunca olduğu gibi yüksek lisans yolculuğumda beni sürekli destekleyen, başta annem Münevver YILDIRIM olmak üzere aileme ve arkadaşlarıma minnettarım.

Kürşat YILDIRIM

Haziran 2021, KAYSERİ

# YAPAY SİNİR AĞLARI KULLANARAK KOT İŞLEME ROBOTLARINDA YÖRÜNGE ANALİZİ

**Kürşat YILDIRIM**

**Erciyes Üniversitesi, Fen Bilimleri Enstitüsü**

**Yüksek Lisans Tezi, Nisan 2021**

**Danışman: Dr. Öğretim Üyesi Mehmet Bahadır ÇETİNKAYA**

## ÖZET

Son yıllarda, endüstriyel robotik sistemlerde analiz ve kontrol uygulamaları, daha çok yapay sinir ağları kullanılmaktadır. Özellikle, yapay sinir ağlarının adaptif yapısından dolayı, tercih sebebi olmaktadır. Robotik sistemlerin lineer olmayan ve kompleks yapısından dolayı, modelleme uygulamalarında, en optimum çözüm sonuçlarını bulmak mümkün olmaktadır.

Bu tez çalışmasında, 6 serbestlik dereceli, endüstriyel bir robot manipülatörün ters kinematik analizi, 4 farklı öğrenme algoritması kullanılarak gerçekleştirilmiştir. Burada kullanılan, yapay sinir ağı için kullanılan öğrenme algoritmaları; Geri Yayılım Öğrenme Algoritması, Random Geri Yayılım Öğrenme Algoritması, Hızlı Geri Yayılım Algoritması ve Delta-Bar-Delta Öğrenme Algoritması şeklinde kullanılmıştır. Sonuçlar, tablo ve grafiksel olarak detaylı irdelendiğinde, Hızlı Geri yayılım Öğrenme Algoritması ile eğitilen 3-10-6 tip yapay sinir ağı ile en iyi performansta tahminleme gerçekleştirmiştir. Netice itibarıyla, bu tip yapay sinir ağı yapısı gerçek zamanlı modellemede, robot kinematiği analizinde de kullanılabilir.

**Anahtar Kelimeler**: Endüstriyel robot manipülatör; Yapay sinir ağları; Öğrenme algoritmaları; Yörünge planlama ve analizi.

**TRAJECTORY ANLYSIS OF DENIM PROCESS ROBOTS USING ARTIFICIAL NEURAL NETWORKS**

**Erciyes University, Graduate School of Natural and Applied Sciences**

**Master Thesis, April 2021**

**Supervisor: Dr. Lecturer. Mehmet Bahadır ÇETİNKAYA**

# ABSTACT

In recent years, there are many applications on analysis and control applications in industrial robotic systems using artificial neural networks. Especially, it is preferred due to the adaptive nature of artificial neural networks. Due to the nonlinear and complex nature of robotic systems, it is possible to find the most optimum solution results in modeling applications.

In this thesis, the inverse kinematic analysis of a 6 degree of freedom industrial robot manipulator was carried out using 4 different learning algorithms. The learning algorithms used here for the artificial neural network are; Back Propagation Learning Algorithm, Random Back Propagation Learning Algorithm, Fast Back Propagation Algorithm and Delta-Bar-Delta Learning Algorithm were used. The results are shown and examined in detail in tables and graphs, the best performance estimation was realized with 3-10-6 type artificial neural network trained with Fast Backpropagation Learning Algorithm. Consequently, this type of artificial neural network structure can also be used in real-time modeling, robot kinematics analysis.

**Keywords:** Industrial robot manipulator**;** Artificial neural networks; Learning algorithms; Trajectory planing and analysis.

# İÇİNDEKİLER

## YAPAY SİNİR AĞLARI KULLANARAK KOT İŞLEME ROBOTLARINDA YÖRÜNGE ANALİZİ

BİLİMSEL ETİĞE UYGUNLUK……..……i

YÖNERGEYE UYGUNLUK……...……...ii

KABUL VE ONAY..……...…iii

TEŞEKKÜR……...…..iv

ÖZET…….…….....v

ABSTRACT……...…….vi

İÇİNDEKİLER…….....vii

KISALTMALAR……....x

SİMGELER……....xi

TABLOLAR LİSTESİ…….……..xiv

ŞEKİLLER LİSTESİ ……...…….…xv

GİRİŞ……...……...…1

**1. BÖLÜM**

**GENEL BİLGİLER VE LİTERATÜR ÇALIŞMASI**

**1.1. Temel Kavramlar……..……..3**

**1.2. Robot Kinematiği……....4**

**1.2.1. Eklem Değişkenlerinin Belirlenmesi……..…5**

**1.2.2. Yörünge Planlanması……..…8**

**1.2.3. Eklem Uzayında Yörüngenin Belirlenmesi ve Planlanması…….10**

**1.2.4. Kartezyen Uzayında Yörüngenin Analizi……....12**

**1.3. Literatür Taraması……..…..16**

## 2. BÖLÜM

## GEREÇ VE YÖNTEM

**2.1. Endüstriyel Robot Manipülatör…….……19**

**2.2. Tasarlanmış Yapay Sinir Ağı…….…….23**

**2.2.1. Geri Yayılım (GY) Öğrenme Algoritması……..27**

**2.2.2. Random Geri Yayılım (RGY) Öğrenme Algoritması……27**

**2.2.3. Hızlı Geri Yayılım (HGY) Öğrenme Algoritması……..28**

**2.2.4. Delta –Bar-Delta (DBD) Öğrenme Algoritması…….28**

**2.3. İleri Beslemeli Yapay Sinir Ağları (YSA)…….....29**

## 3. BÖLÜM

## BULGULAR

**3.1. Tasarlanmış Yapay Sinir ağları ile Tahminleme…….35**

## 4. BÖLÜM

## TARTIŞMA-SONUÇ ve ÖNERİLER

**4.1.Tartışma Sonuç……61**

**4.2. Öneriler ……..64**

**KAYNAKLAR……...……...66**

**EKLER……..70**

**EK 1. GY Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++Kodu……..70**

**EK 2. RGY) Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++Kodu……..73**

**EK 3. HGY Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++ Kodu……..78**

**EK 4. DBD Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++Kodu……81**

**EK 5. GY Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++Kodu……....86**

**EK 6. RGY Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu……91**

**EK 7. HGY Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu……...91**

**EK 8. DBD Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu……...100**

# KISALTMALAR

| Kısaltma | İngilizce Karşılığı | Türkçe Karşılığı |
|---|---|---|
| YSA | Artificial Neural Network | Yapay Sinir Ağları |
| GY | Backpropagation | Geri Yayılım |
| GYA | Backpropagation Algorithm | Geri Yayılım Algoritması |
| HGY | Fast Backpropagation | Hızlı Geri Yayılım |
| DBD | Delta –Bar-Delta | Delta –Bar-Delta |
| RGY | Random Backpropagation | Random Geri Yayılım |
| D-H | Denavit-Hartenberg | Denavit-Hartenberg |

# SİMGELER

| Simge | Karşılığı |
|---|---|
| X | x yönünde eksen |
| Y | y yönünde eksen |
| Z | z yönünde eksen |
| T | dönüşüm matrisi |
| N | eklem fonksiyon sayısı |
| **a** | eksenleri arasındaki dik uzaklık uzuv uzunluğu |
| **Ɵ** | iki bağ arasında oluşan eklem açısı |
| $t_0$ | başlangıç zamanı |
| $Ɵ_0$ | $t_0$ anındaki başlangıç konumu |
| **s** | polinom sabit katsayısı |
| $\theta_b$ | parabolic kısmın sonundaki konumu, |
| $t_b$ | yörünge orta zaman |
| $\theta_h$ | gerçekleşen yörüngenin tam ortası |
| $t_h$ | yarı zaman |
| $\ddot{\theta}$ | parabolik bölgedeki ivme |
| $\theta_b$ | parabolic kısmın sonundaki konum |
| $u_i$ | gizli katmanın çıkış katmanın çıkış sinyalleri |
| *f(.)* | basit doğrusal olmayan bir fonksiyon |
| η | öğrenme oranı |

| | |
|---|---|
| $\alpha$ | momentum terimi |
| $\Delta w_{ij}$ | ağırlık değişimi |
| $E(w)$ | ağırlığa bağlı hata eğimindeki değişiklik |
| $S(t)$ | ağırlığa göre hatanın türevidir |
| $\delta$ | geçmiş değerlerinin üstel ortalaması |
| $e_1$ | 1. mafsal açısı hatası |
| $e_2$ | 2. mafsal açısı hatası |
| $e_3$ | 3. mafsal açısı hatası |
| $e_4$ | 4. mafsal açısı hatası |
| $e_5$ | 5. mafsal açısı hatası |
| $e_6$ | 6. mafsal açısı hatası |
| $\Theta_1$ | 1.mafsal bağ açısı |
| $\Theta_2$ | 2.mafsal bağ açısı |
| $\Theta_3$ | 3.mafsal bağ açısı |
| $\Theta_4$ | 4.mafsal bağ açısı |
| $\Theta_5$ | 5.mafsal bağ açısı |
| $\Theta_6$ | 6.mafsal bağ açısı |

# TABLOLAR LİSTESİ

*Tablo 1.1*. Doğrusal yörüngeye ait konum, hız ve ivme denklemleri……..…..…15

*Tablo 2. 1*. Endüstriyel Robot manipülatörün teknik özellikleri ve parametreleri…..…21

T*ablo 2.2.* Endüstriyel robot manipülatörün D-H parametreleri……..…22

*Tablo 3.1*. Öğrenme Algoritmaları, YSA yapısı ve ortalama- maksimum hatalar…..…59

# ŞEKİLLER LİSTESİ

*Şekil 1.1*. Uzuv ve eksenleri numaralandırılması…….6

*Şekil 1.2*. Eksenlerin koordinat sistemleri ile gösterilişi……...6

*Şekil 1.3*. Eksenlerin arasındaki mesafe şematik gösterilişi…….7

*Şekil 1.4*. Uzuv ekseneleri doğrultusunda arasındaki mesafe $d_i$ şemantik gösterimi….7

*Şekil 1.5*. Eklem eksenleri arasındaki açı $\alpha_{i-1}$ gösterimi……8

*Şekil 1.6*. Eklem açıları $\theta_i$ gösterimi…….8

*Şekil 1.7*. Uç İşlevcinin Kartezyen ve Eklem uzayında takip ettiği yörünge (Bingül, 2009)……9

*Şekil 1.8*. Üçüncü dereceden kübik yörüngeye ait a) Konum, b) Hız ve c) İvme grafiği görünümü…….……12

*Şekil 1.9*. Doğrusal yörüngenin grafiksel görünümü…….……..13

*Şekil 1.10*. Parabolik kısımlarla oluşturulan doğrusal yörünge değişimi……13

*Şekil 1.11*. Parabolik kısımlar eklenmiş doğrusal yörüngeye ait a)Konum, b) Hız ,

c) İvme grafiksel değişimlerinin gösterimi……15

*Şekil 2.1.* Operatör esaslı jeans kot pantolon kimysal püskürtme işlemi görünümü…...19

*Şekil 2.2*. 6 Serbestlik dereceli endüstriyel robot ve dönme eksenleri…….....……...20

*Şekil 2.3.* Robot manipülator ve eksen takımlarının görünümü……..22

*Şekil 2.4.* İnsan beynindeki hücre yapısı ve bileşenleri.....24

*Şekil 2.5*. Temel Yapay Sinir Ağlarının yapısının şematik gösterilişi.....25

*Şekil 2.6*. Yapay Sinir Ağının Çalışma İşlemlerinin Blok diagramı.....26

*Şekil 2.7*. İleri beslemeli, 3-5-6 yapay sinir ağı modelinin şematik örünümü……30

*Şekil 2.8*. İleri beslemeli, 3-10-6 yapay sinir ağı modelinin şematik örünümü…….31

*Şekil 2.9*. **3-10-6** tipi yapay sinir ağı kullanılarak robot manipülatörün Modellenmesi…….32

*Şekil 2.10*. **3-5-6** tipiYapay sinir ağı kullanılarak robot manipülatörün Modellenmesi…….33

*Şekil 3.1.* 1.mafsal açısının **($\Theta_1$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimi……..35

*Şekil 3.2.* 2. mafsal açısının **($\Theta_2$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….…….36

*Şekil 3.3.* 3.mafsal açısının **($\Theta_3$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….…….37

*Şekil 3.4.* 4.mafsal açısının **($\Theta_4$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….……...38

*Şekil 3.5.* 5.mafsal açısının **($\Theta_5$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….……..39

*Şekil 3.16.* 6.mafsal açısının **($\Theta_6$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….……..40

*Şekil 3.7.* 1.mafsal açısının **($\Theta_1$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……41

*Şekil 3.8.* 2.mafsal açısının **($\Theta_2$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……42

*Şekil 3.9.* 3.mafsal açısının **($\Theta_3$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……43

*Şekil 3.10.* 4.mafsal açısının **($\Theta_4$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……44

*Şekil 3.11.* 5.mafsal açısının **($\Theta_5$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……45

*Şekil 3.12.* 6.mafsal açısının **($\Theta_6$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……46

*Şekil 3.13.* 1.mafsal açısının **($\Theta_1$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……47

*Şekil 3.14.* 2. mafsal açısının **($\Theta_2$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……48

*Şekil 3.15*. 3. mafsal açısının **($\Theta_3$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….49

*Şekil 3.16.* 4. mafsal açısının **($\Theta_4$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……....50

*Şekil 3.17.* 5. mafsal açısının **($\Theta_5$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……..…….…..51

*Şekil 3.18.* 6. mafsal açısının **($\Theta_6$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….……52

*Şekil 3.19.* 1. mafsal açısının **($\Theta_1$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……..……...53

*Şekil 3.20.* 2. mafsal açısının **($\Theta_2$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi…….54

*Şekil 3.21.* 3. mafsal açısının **($\Theta_3$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……..……..55

*Şekil 3.22.* 4. mafsal açısının **($\Theta_4$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……..……..56

*Şekil 3.23.* 5. mafsal açısının **($\Theta_5$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……57

*Şekil 3.24.* 6. mafsal açısının **($\Theta_6$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi……58

# GİRİŞ

Çalışma alanında belirli bir konuma sabitlenmiş tabana sahip robotlar, endüstriyel robot sistemlerinin geleneksel türünü oluşturur. Onlar mekanik bir kol, uygun eyleyici ve elektronik elemanlar aracılığıyla canlandırılan eklem ve uzuvların bir zincirini oluşturur. Bununla birlikte, endüstriyel robot sisteminin yapısı, mekanik bir altsistem, devindiriciler, algılama altsistemi ve kontrol sisteminden oluşur.

Endüstriyel robotlar, teknolojik gelişmeler ile birlikte farklı endüstriyel uygulamalarda yaygın bir şekilde kullanılmaya başlanmıştır. Robot manipulator sistemleri elektronik, donanım, yazılım bileşenlerinden oluşan ve kontrol edilebilen elektro-mekanik sistemlerdir. Robot manipulatör sistemleri taşıma, montaj, sürekli yada punta kaynaklı birleştirme, boyama ve presleme gibi önemli endüstriyel uygulamalarda özellikle hız, hassasiyet ve kalite açısından yüksek performans sağladıkları için yaygın olarak tercih edilmektedir.

Endüstriyel robotlar, insan müdalesi ve etkisi olmadan endüstriyel işlemleri etkin ve verimli bir şekilde gerçekleştirebilme yeteneklerinden dolayı üretim süreçlerinin her aşamasında yaygın olarak kullanılmaktadır. Ayrıca, kimyasal ve kansorejenler içeren süreçlerde operatör destekli üretim modelinden robot manipulator destekli yapılara geçilmesiyle iş sağlığı ve güvenliği açısından önemli avantajlar sağlanmıştır.

Robot manipulator sistemlerinde karşılaşılan en önemli problemlerden birisi yüksek doğrulukta *Pozisyonlama* işleminin gerçekleştirilebilmesidir. Bu çalışma kapsamında analiz edilen kot kumaş taşlama sürecinde, kumaşın aşındırılacağı bölge belli olduğundan uç işlevcinin tam yörüngeyi en yüksek doğrulukla izlemesi gerekmektedir. Ayrıca, yörünge analizi yapılırken en önemli bozucu etkiler olan ortam havalandırması ve mekanik eksen kaçıklıkları da dikkate alınmalıdır.

Bu tez çalışmasında tekstil sektörüne yönelik robot manipulator sistemleri için yapay sinir ağlarına (YSA) dayalı pozisyonlama analizleri gerçekleştirilmiş ve en yüksek doğrulukta pozisyonlama için optimum bir YSA modeli geliştirilmeye çalışılmıştır. Altı (6) mafsallı bir robot manipülatörün yörünge analizi ters kinematik yaklaşımı kullanılarak teorik olarak hesaplanmış ve ilgili robot manipülatörün yapay sinir ağlarına dayalı ters kimenatik modeli oluşturulmuştur. Benzetim çalışmalarında ayrıca farklı YSA modelleri kullanılarak bu modellerin performans mukayeseleri gerçekleştirilmiştir. Elde edilen sonuçlardan ileri beslemeli, tek gizli katmanlı, çok hücreli (10 hücre) ve hızlı yayılım algoritmalı YSA modelinin kot kumaş taşlama sürecine yönelik en başarılı *Pozisyonlama* modelini oluşturduğu ve YSA tabanlı yaklaşımların robot manipülatörlerde yüksek doğrulukta *Pozisyonlama* elde edebilmek için etkin bir şekilde kullanılabileceği görülmüştür.

Bu tez çalışmasında, birinci bölümde robotik sistemler hakkında detaylı bilgiler ve tez konusu ile ilgili yapılan çalışmalar hakkında açıklamlar yapılmıştır. İkinci bölümde ise robot kinematiği, kullanılan robot tasarımı, eksen takımları ve yapay sinir ağlarında özellikle ağırlıkların değişimini sağlamada kullanılan öğrenme algoritmalrının teorisi detaylandırılmıştır. Üçüncü bölümde simülayon çalışmaları neticesinde farklı algoritmalar kullanılarak YSA model sonuçları verilerek detaylı açıklanmıştır. Dördüncü bölümde ise sonuçlar, iredeleme ve önerilere açıklanmaya çalışılmıştır.

# 1. BÖLÜM
# GENEL BİLGİLER VE LİTERATÜR ÇALIŞMALARI

## 1.1. Temel Kavramlar

Tipik bir manipulator robot sistemi üç temel modüler yapıdan oluşur. Bunlar sırası ile kumanda konsolu, manipulator robot ve control panosudur [1]. Kumanda konsolu, robot opratörü ile robot arasında iletişim ve control işlemlerine imkan sağlarlar. Kontrol panosu ise kumanda konsolundan gelen komutlar ile ilgili hareketlerin robot manipulator tarafından gereçekleştilmesini sağlamak üzere gereken elektriksel güç sinyallerini üreterekl robot manipülatöre iletir. İletilen bu güç sinyalleri robot yapısı içerisinde yer alan servo motorlar sayesinde harekete geçirirler.

Şekil 1.1'de gösterilen detaylardan da belirleneceği gibi, control panosu robot sisteminin bilgisayar sistemi, ilgili işletim sistemi, robot control-kumanda yazılımı, servo motor sürücüleri, control kartları ve güç-elektroniği kısımlarından oluşur. Diğer taraftan, robot manipülatör sistemi içinde ise servo motorlar, ekoderler, dişli redüktör mekanızmalar, mkaik robot yapı elemanları mevcuttur.

Bu simulasyon destekli çalışmada, özellikle tehlikeli ortamda kullanılan insan faktörünü devre dışı bırakarak, tamamen kimyasal olarak, kot kumaşından üretilen pantalon üzerine güskürtülen karışımın robot maipülatör tarafından yapılması durumunda, posiyonlama için optimum YSA modeli belirlenmeye çalışılmıştır.

## 1.2 . Robot Kinematiği

Geometrik bir sistemin yapısını ve hareketlerini inceleyen bilim dalına kinematik olarak ifade edilir. Bir sistemin belirli şartlar altında nasıl hareket ettiğini anlayabilmek için bu sistemin kuvvet atalet ve enerji gibi büyüklüleri yani, dinamiği hakkında belirgin oranda bilgi birikimine sahibi olmak gerekir. Robot manipülatörlerin ileri yön kinematiği, robot bağlarının konumları, hızları ve ivmeleri arasındaki ilişkiyle ilgilidirr. Bir açık kinematic zincir özelliğine sahip, seri bir robot manipülatör, ana çerçevesinden araç çerçevesine doğru birbirine prizmatik veya dönel eklemlerle tutturulmuş seri bağlardan oluşur. Eğer her ekleme bir koordinat sistemi yerleştirilirse, komşu iki eklem arasındaki ilişki bir i 1,T dönüşüm matrisi ile ifade edilir. İlk ekleme ait dönüşüm matrisi, ilk eklemle ana çerçeve arasında bir ilişki tanımlarken son ekleme ait dönüşüm matrisi ise uç işlevcisi ile son eklem arasında bir ilişki tanımlar. Arka arkaya sıralanan bu eklem dönüşüm matrislerinden yararla-narak ana çerçeveyle araç çerçevesi arasında bir ilişki tanımlanır. Bu ilişkiye ileri kinematik denir ve araç çerçevesinin yönetimini ve konumunu ana çerçeveye göre ifade eder ve aşağıdaki gibi gösterilir (Bingün, 2009; Kuşcu, 2015).

$$ {}^{0}_{N}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T \dots\dots\dots {}^{N-1}_{N}T \qquad (1.1) $$

Her bir eklem matrisi bir eklem değişkeninin fonksiyonudur. °T dönüşüm matrisi Ise N tane eklemin birer fonksiyonudur. Bir robotun uç işlevcisinin çalışma uzayında serbestçe hareket edebilm "için 6 serbestlik iş derecesine sahip olması yeterlidir.Eklem matrislerine ek olarak ana çerçeve ile istasyon çerçeve (istasyon 0T) ve uç işlevci ile araç çerçeve arasında (araç-çerçeve NT) matrisleri tanımlanır. Araç çerçeve ile istasyon çerçe-ve arasındaki toplam dönüşüm ise aşağıdaki gibi ifade editir

*Araç$^{istasyon}$ $_{çerçeve}$ T=Istasyon 0T°N.Tara, çerçeve*

Yukarıdaki eşitlikte yer alan ana çerçeve ile istasyon çerçeve ve uç işlevci ile araç çerçeve arasındaki dönüşümler sabit olduğundan bu dönüşümler °NT ileri kinematik matrisinin içerisinde yer alır.

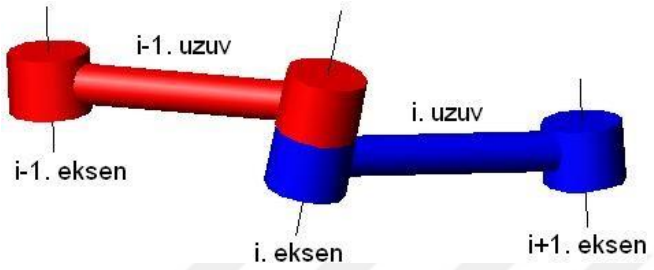
### 1.2.1. Eklem Değişkenlerinin Belirlenmesi

Robotlarin eklem değişkenlerinin belirlemek için bir çok kinematik yöntem geliştirilmiştir. Kinematik problemlerin çözümü, Kartezyen üç boyutlu ve Kartonyum [6] dört boyutlu olmak üzere iki farklı uzayda gerçekleştirilir. Kartezyen uzayda üstel yöntem, Pieper-Roth yöntemi baş vurulan yöntemlerden bazılarıdır. Fakat çoğunlukla bir robotun eklem değişkenlerinin belirlenmesinde en fazla tercih edilen yöntem kısaca D-H olarak gösterilen Denavit-Hartenberg yöntemidir.

Denavit-Hartenberg yönteminde dört ana değişken kullanılarak robot kinematiği gkanlır. Bu değişkenler, iki eksen arasındaki bağ uzunluğu, (*i-1*) ile *i* eksenleri arasındaki bağ açısı, üst üste çakışan bağlar arasındaki eklem kaçıklığı, ve iki bağ arasında oluşan eklem açısı θ 'dır. Bu dört genel D-H değişkeni denir. Bu değişkenleri belirlemek için, öncelikle şekilde de gösterildiği gibi robotun dönme eksenleri belirlenir ve dönme eksenleri bağlardan bir fazla olacak şekilde numaralandırılır.

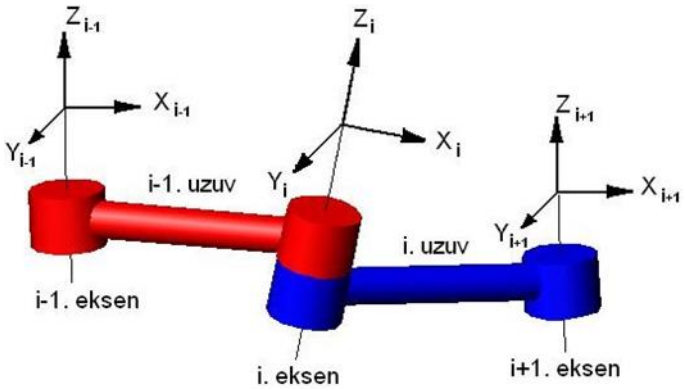
Öncelikle eklem eksenleri dönme veya kayma yönleri belirlenir ve bu eksene paralel bir doğru çizilir 2. Bu işlem gerçekleştirilirken eklem eksenleri, döner eksenler için dönme yönü Z, prizmatik eklemler için kayma yönü Z ekseni olarak belirlenir. 3. Z eksenine dik ve kol boyunca olan bağ (uzuv) uzunluğu X ekseni olarak kabul edilir. 4. Z ve X eksenleri belirlendikten sonra sağ el kuralÍna göre Y ekseni bulunur. 5. Eğer arka arkaya gelen 2 eklemin dönme veya kayma yönleri aynÍ ise Z ekseni belirlendikten sonra kol boyunca X ekseni belirlenir. Son olarak sağ el kuralina göre Y ekseni belirlenir. 6. 0 ve 1. eksenler üst üste kabul edilebilir. 7. Bir seri robotun eklemine koordinat sistemleri yerleótirilirken 1. eksenin dönme yönü Z ekseni olarak belirlendikten sonra genellikle bu eksene X eksenince döndürüldüğünde komóu iki Z ekseni çakışacak şekilde bir X ekseni yerleştirilir

Bu değişkenleri belirlemek için öncelikli olarak şekilden görüldüğü gibi robotun dönme eksenleri belirlenir ve dönme eksenleri uzuvlardan bir fazla olacak şekilde numaralandırılır.



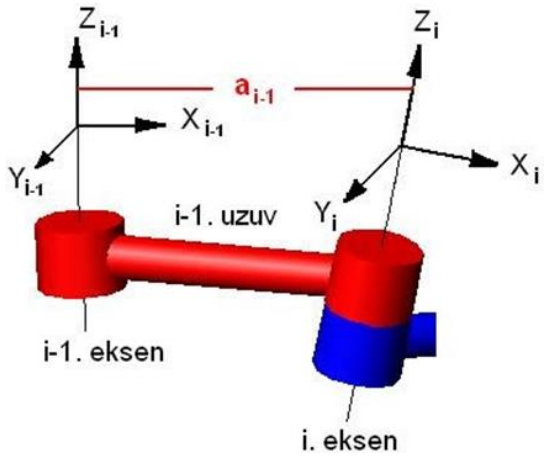

Şekil 1.1. Uzuv ve eksenleri numaralandırılması

Daha sonra bu eksenlerin her birine bir koordinat sistemi yerleótirilir ve uzuv dönme ekseni aşağıdaki şekilde görüldüğü gibi koordinat sisteminin Z ekseni olarak kabul edilir.



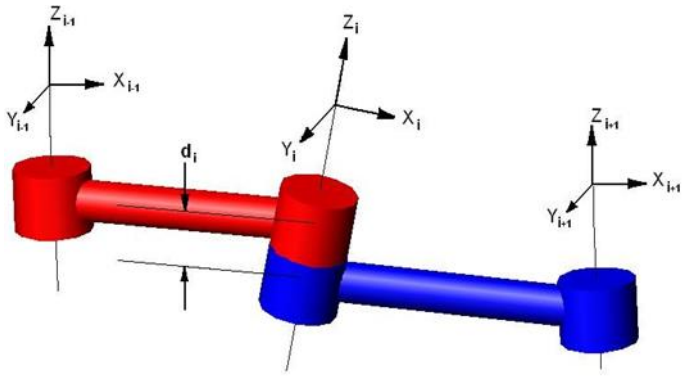

Şekil 1.2. Eksenlerin koordinat sistemleri ile gösterilişi

Aşağıdaki şekildeki gibi x $_{i-1}$ yönünde uzanan $Z_{i-1}$ ile Z $_i$ eksenleri arasındaki dik uzaklığa $a_{i-1}$ uzuv uzunluğu denir.




Şekil 1.3. Eksenlerin arasındaki mesafe şematik gösterilişi

Aşağıdaki şekildeki gibi $Z_i$ yönünde uzanan $X_{i-1}$ ile $X_i$ eksenleri arasındaki dik uzaklığa $d_i$ eklem kaçıklığı denir.



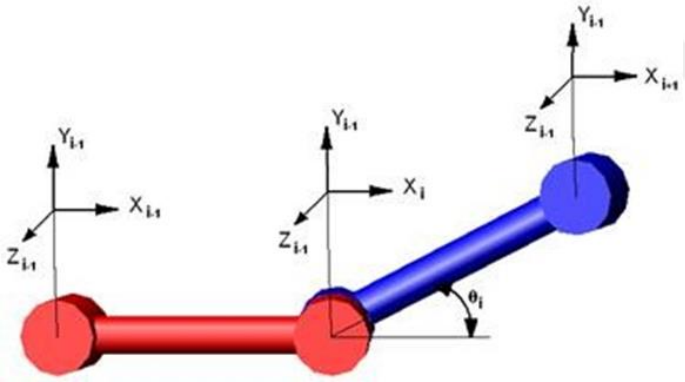

Şekil 1.4. Uzuv ekseneleri doğrultusunda arasındaki mesafe $d_i$ şemantik gösterimi

$Z_{i-1}$ ile $Z_i$ dönme eksenleri arasında oluşan açı şekildeki gibi $a_{i-1}$ bağ açısıdır.




Şekil 1.5. Eklem eksenleri arasındaki açı $\alpha_{i-1}$ gösterimi

$\theta_i$ eklem açısı ise $a_{i-1}$ ile $a_i$ bağları arasında X ekseni boyunca ölçülen açıdır.

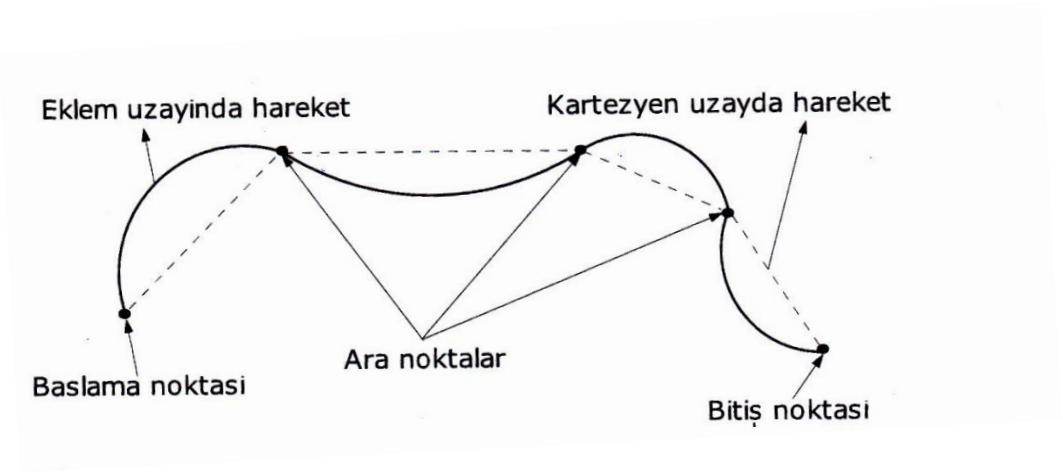



Şekil 1.6. Eklem açıları $\theta_i$ gösterimi

### 1.2.2. Yörünge Planlanması

Uç işlevcinin bulunduğu konumdan istenen konuma gelmesini sağlamak, bir robot manipülatörün en temel problemlerinden biridir. Robot uç işlevcisinin bu hareketi gereçkleştirme esnasında yönelimi ve konumu ana çerçeveye göre değişir. Gerçekte, uç işlevcinin başlanguıç ve bitiş noktaları da birer ara nokta sayılır. Uç işlevcisi bu noktalardan geçerekn, zaman bağlı olarak değişen belli biz hıza ve ivmeye sahiptir. Bu değişken ivmeli hareket titreşime neden olabilir. Endüstriyel robot manipülatörün

titreşimden yalıtılmış, çalışma uzayındaki herehangi bir cisme çarpması söz konusu olmadan, eyleyicilerin maksimum sınırlarını zorlamadan kontrollü ve yumuşak bir şekilde hareket edebilmesi için yörünge planlaması gerekir. Robotik sistemler iki nokta arasında doğrusal, dairsel, sinüzoidal veya değişik polinomlar şekillerinde takip etmesi sağlanabilir. Bu yörünge şekilleri zamana bağlı olarak eklem açılarının veya kartezyen koordinat sitemlerinin ötelemelerinin birer fonksiyonudur. Her ara nokta, ilk önce ana çerçeveye göre uç işlevcinin konumu ve yönelimi cinsinden tanımlanır. Daha sonra, her bir ara nokta, ters kinematic çözümümün uygulanmasıyla eklem açılarına dönüştürülür. Bu iki nokta arası da seçilen bir yörünge planlama yöntemiyle n adet noktaya bölünerek gerçekleştirilir. Ara noktalar bütün eklemler tarafından aynı zamanda geçilmesine rağmen her bir eklemin alacağı mesafe aynı olmayacağından, hız ve ivmeleri farklılık gösterebilecektirler. Bu nedenle, bir eklemin hız ve ivmesi diğerlerinden bağımsızdır. Netice itibarıyla, robotic sistemlerdeki eyleyiciler bu açılarla çalıştığı zaman arzu edilen yönelim ve konuma ulaşılabilir. Robot manipülatörler için kartezeyen ve eklem uzayı olmak üzere iki farklı yörünge planlaması yapılır. Şekil 1.7.'de bir uç işlevcinin kartezyen ve eklem uzayında takip ettiği yörüngenin başlangıç, bitiş ve ara noktaları verilmektedir.




Şekil 1.7. Uç İşlevcinin Kartezyen ve Eklem uzayında takip ettiği yörünge (Bingül, 2009)

### 1.2.3. Eklem Uzayında Yörüngenin Belirlenmesi ve Planlanması

Eklem uzayında yörünge planlaması yapılırken üç veya daha yüksek dereceli polinom fonksiyonlar kullanılır. Örnek olarak; uç işlevcisi, başlangıç noktasından arzu edilen noktaya belli bir zaman aralığında gitsin. İlk olarak uç işlevcinin başlangıç ve hedef noktalarının konumu ve yönelimi eklem açıları cinsinden ters kinematic yöntemiyle hesaplanır. Uç işlevcinin $t_0$ anındaki başlangıç konumu $\Theta(0)= \Theta_0$ ve $t_f$ anınsaki hedef konumu $\Theta(t_f)=\Theta_f$ olsun. Bu durumda, $t_0$ ile $t_f$ arası üçüncü dereceden bir polinom kullanılarak n tane noktaya bölünerek, bu iki koşula ek olarak başlangıç ve bitiş hızları $\dot{\theta}(0) = 0$ ve $\dot{\theta}(t_f) = 0$ eklenir (Craig, 1989).

$$\Theta(0)=\Theta_0 \tag{1.2}$$

$$\Theta(t_f)=\Theta_f$$

$$\dot{\theta}(0)= 0$$

$$\dot{\theta}(t_f)= 0$$

Yukarıdaki dört koşul, üçüncü dereceden bir polinom katsayılarını bulmak için yeterlidir. Bu koşullar, zamana bağlı kübik bir yörünge oluşturan 3.dereceden polinom olarak denklem 67.5'deki gibi ifade edilebilir.

$$\Theta(t)= s_0+s_1t+s_2t^2+s_3t^3 \tag{1.3}$$

Bu yörüngedeki eklem hızları ve ivmeleri denklem 1.3'ün 1. ve 2. dereceden türevlerei alınarak bulunur.

$$\dot{\theta}(t)=s_1+2s_2t+3s_3t^2 \tag{1.4}$$

$$\ddot{\theta}(t)=2s_2+6s_3t \tag{1.5}$$

Denklem 2.1, denklem 2.5'te yerine konursa, aşağıdaki denklem ifadeleri elde edilir.

$$\Theta(0)= s_0+s_10+s_20^2+s_30^3 \tag{1.6}$$

$s_0=\Theta(0)$ veya $s_0=\Theta_0$

$\Theta(t_f)=\Theta_f$

$\Theta_f = s_0+s_1t_f+s_2 t_f^2+s_3 t_f^3$ (1.7)

Denklem 2.3 ve denklem 2.4, denklem 2.6'da yerine konulduğunda denklem şu şekilde ifade edilir.

$\dot{\theta}(0)=s_1+2s_20+3s_30^2$ (1.8)

$s_1 = 0$

$\dot{\theta}(t_f)=s_1+2s_2t_f+3s_3t_f^2$

Polinom katsayıları yukarıda elde edilen denklemlerden faydalanarak bulunur. Bilindiği gibi $s_0=\Theta_0$ ve $s_1=0$ sırasıyla denklem 1.6 ve denklem 1.8'de bulunmuştu. Diğer iki katsayıdan $s_2$ denklem 1.9'da $\dot{\theta}(t_f)= 0$ ve $s_1=0$ değerleri yerine konarak bulunur.

$\dot{\theta}(t_f)=s_1+2s_2t_f+3s_3t_f^2 =0$

$0= 0+2s_2t_f+3s_3t_f^2$ (1.9)

$2s_2t_f = -3s_3t_f^2$ ise $s_3=-\frac{2s_2}{3t_f}$ veya $s_2= -\frac{3s_3\ t_f}{2}$ $s_3=-\frac{2s_2}{3t_f}$ ifadesi denklem 1.10'da yerine konulursa $s_2$ katsayısı aşağıdaki gibi bulunur ($s_0=0$ ve $s_1=0$) .

$\Theta_f = s_0+s_1t_f+s_2 t_f^2+s_3 t_f^3$

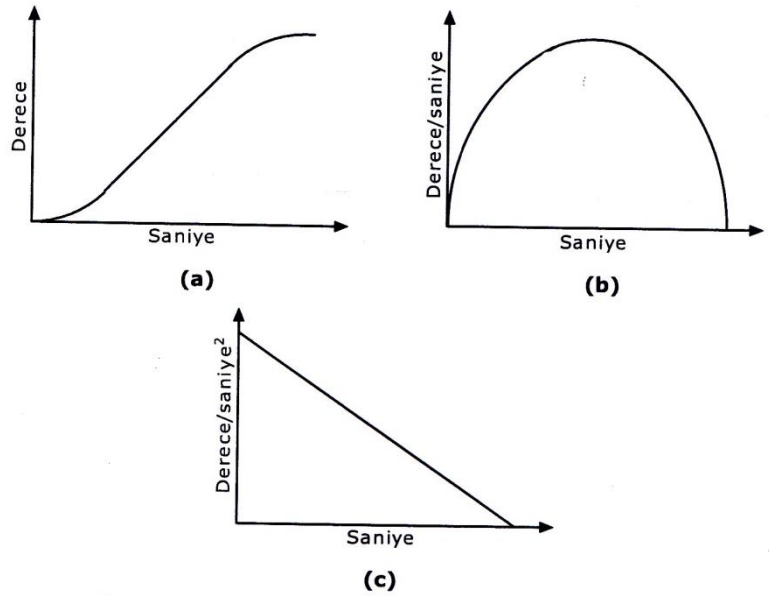
$\Theta_f = \Theta_0+0\ t_f+s_2 t_f^2-\frac{2s_2}{3t_f} t_f^3$

$\Theta_f = \Theta_0+s_2 t_f^2-\frac{2}{3} s_2t_f^2$

$s_3= -\frac{2}{t_f^3}(\Theta_f - \Theta_0)$ (1.10)

Netice itibarıyla bulunan katsayılar aşağıdaki şekilde ifade edilebilir.

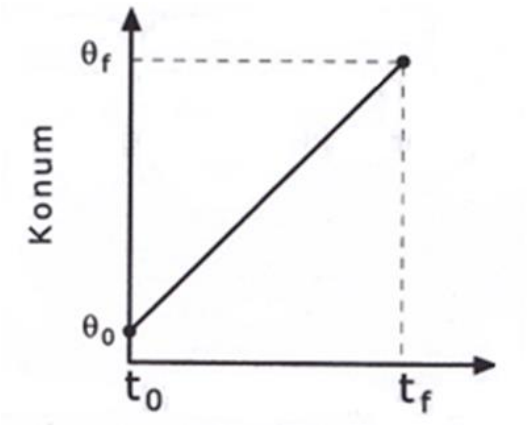
$s_0=\Theta_0$, $s_1= 0$, $s_2=\frac{3}{t_f^2}(\Theta_f - \Theta_0)$, $s_3= -\frac{2}{t_f^3}(\Theta_f - \Theta_0)$ bu katsayılardan faydalanarak, başlangıç ve bitiş hızları sıfır olan 3.dereceden kübik bir yörüngeye ait konum, hız ve ivme değişimi Şekil 1.8'deki gibi gösterilebilir.




Şekil 1.8. Üçüncü dereceden kübik yörüngeye ait a) Konum, b) Hız ve c) İvme grafiği görünümü

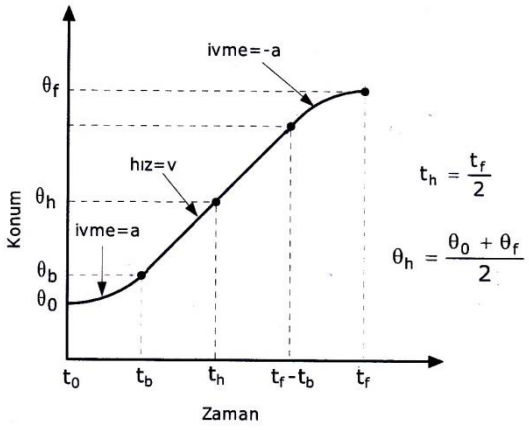
### 1.2.4. Kartezyen Uzayında Yörüngenin Analizi

Endüstriyel robot manipülatörlerinin izlediği yörünge yöntemlerinden bir olan diğer yöntem de, parabolic kısımlşarın yerleştirildiği doğrusal yörüngedir. Bu yöntemde, n serbestlik derecesine sahip robot kol eklemlerinin tamamı bir doğrusal yörünge takip etme durumu olsa bile, uş işlevci iki nokta arasını doğrusal geçmesi söz konusu değildir. Şekil x.x.'de görülen doğrusal yörüngede başlangıç ve bitişine parabolik kısımlar eklenerek konum ve hızın sürekliliği sağlanır.




Şekil 1.9. Doğrusal yörüngenin grafiksel görünümü

Ayrıca, parabolic bir yörüngede sabit ivmeli bir hareket kullanılarak, hızın pürüzsüz bir şekilde sürekli olması sağlanır. Şekil 1.11.'de parabolic kısımlar eklenmiş doğrusal yörünge görülmektedir.




Şekil 1.10. Parabolik kısımlarla oluşturulan doğrusal yörünge değişimi

Yukarıdaki Şekildeki $t_0$-$t_b$ ile $t_f$ arası doğrusal yörüngenin başlangıcına ve bitişine eklenmiş parabolik kısımları, $t_b$ ile $t_f$-$t_0$ arası ise doğrusal bölgeyi göstermektedir. Yine aynı Şekilde, parabolic kısımlar eklenerek kalan bölümler simetrik olacak şekilde tasarlanır. Bunun için parabolic kısımlarda gerçekleşen zamanların bir birine eşit olmasının yanında ivmelerde aynı ve sabit olmalıdır. Aynı zamanda, süerekli bir hız elde edbilmek için parabolic kısımların sonundaki hızla doğrusal bölümdeki hızın bir birine eşit olması gerekir. Bu durumda her bir parabolic kısımdaki ivme aşağıdaki denklemle açıklanbilir.

$\ddot{\theta}\ t_b = \frac{\theta_h - \theta_b}{t_h - t_b}$ (1.11)

Denklemde, $\theta_b$ parabolic kısmın sonundaki konumu, $t_b$ zamanı, $\theta_h$ ise gerçekleşen yörüngenin tam ortasını, $t_h$ ise yarı zamanı ve son olarak $\ddot{\theta}$ parabolik bölgedeki ivmeyi göstermektedir. $\theta_b$ parabolic kısmın sonundaki konum aşağıdaki gibi elde edilir.

$\theta_b = \theta_0 + \frac{1}{2}\ \ddot{\theta}\ t_b^2$ (1.12)

Denlem (2.13) düzenlendiğinde aşağıdaki ifade elde edilir.

$\ddot{\theta}\ t_b(t_h - t_b) = \theta_h - \theta_b$ (1.13)

Yörüngenin tümünde gerçekleşen zam t olduğundan $t_h = \frac{t}{2}$ 'dir. Denklem (1.13)'de yerine konulursa ve $\theta_h$ yerine $\frac{\theta_f - \theta_o}{2}$ yazıldığında, aşağıdaki denklem elde edilir.

$\frac{\ddot{\theta} t_b^2}{2} - \frac{\ddot{\theta}}{2}\ t\ t_b + \frac{\theta_f - \theta_o}{2} - \theta_0 = 0$ (1.14)

Denklem tekrar düzenlendiğinde aşağıdaki şekilde ifade edilir.

$\ddot{\theta}\ t_b^2 - \ddot{\theta}\ t\ t_b + \theta_f - \theta_0 = 0$ (1.15)

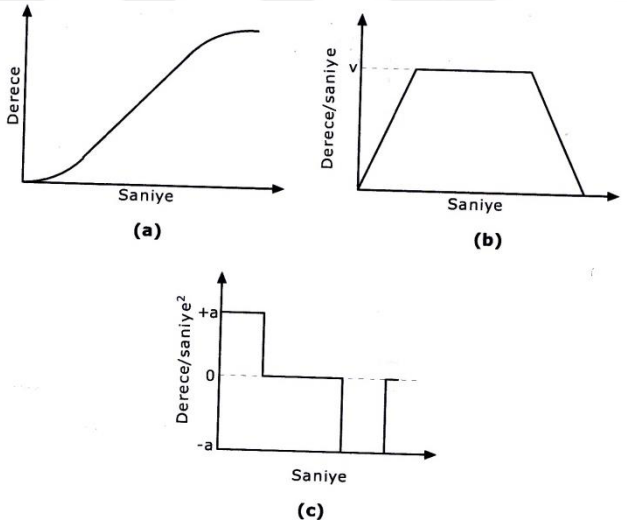
elde edilir. $\theta_f$ , $\theta_0$ ve t değişkenlerinin verilmesi ve $\ddot{\theta}$ ile $t_b$ değişkenlerinden birinin seçilmesiyle birçok yörünge tasarlanabilir. Genellikle $\ddot{\theta}$ seçilerek $t_b$ hesaplanır. Bu durumda $t_b$ aşağıdaki gibi olur. İvmenin büyük seçilmesi parabolic kısımların küçülmesine neden olacağı unutulmamalıdır.

$t_b = \frac{t}{2} - \frac{\sqrt{t^2 - 4\left(\frac{\theta_f - \theta_o}{\ddot{\theta}}\right)}}{2} = \frac{t}{2} - \frac{\sqrt{\ddot{\theta}^2 t^2 - 4\ \ddot{\theta}\ (\theta_f - \theta_o}}{2\ \ddot{\theta}}$ (1.16)

Tablo 1.1. Doğrusal yörüngeye ait konum, hız ve ivme denklemleri

| Zaman (t) | $\theta(t)$ | $\dot{\theta}(t)$ | $\ddot{\theta}(t)$ |
|---|---|---|---|
| $0 \leq t \leq t_b$ | $\theta_0 = \frac{a}{2}t^2$ | a t | a |
| $t_b \leq t \leq t_f - t_b$ | $vt + \frac{\theta_f + \theta_0 - vt_f}{2}$ | $v$ | 0 |
| $t_f - t_b \leq t \leq t_f$ | $\theta_f - \frac{a\, t_f^2}{2} + a\, t_f t - \frac{a}{2}t^2$ | $at_f - at$ | $-a$ |

Tablo 2.1'de gösterilen denklemlere ek olarak $t_0 = 0$ , $t_f > 0$ $t_b > 0$ kabulleriyle $\frac{\theta_f - \theta_o}{v} < t_f \leq \frac{2(\theta_f - \theta_o}{v}$ ve $v^2 \leq a(\theta_f - \theta_0$ sınırlamalarının sağlanması gerekir.




*Şekil 1.11.* Parabolik kısımlar eklenmiş doğrusal yörüngeye ait a)Konum, b) Hız ,

c) İvme grafiksel değişimlerinin gösterimi

## 1.3. Literatür Taraması

Robotlar son yıllarda endüstriyel uygulamalarda ve üretim alanında problem çözümünde çok yaygın olarak, bir çok üstünlüğe sahip olduklarından vazgeçilmez terci nedeni olmuşturlar. Bu alanda yapılan bilimsel çalışmaları detaylandırmak istersek. Saad (1994) ve arkadaşları, uyarlanabilir ve yapay sinir ağlı kontrolcü tasarımı ve robot manipülatörlerde uygulanması konusunda araştırma çalışmaları yapmıştırlar. Briot (2008a) ve arkadaşları, toplama ve yerleştirme uygulamaları için yeni bir tamamen ayrılmış 3 serbestlik dereceli çevirmeli paralel robot tasarımı ve gerçekleştirilmesi üzerine bilimsel bir çalışma gerçekleştirmişlerdir. Yine Briot (2008b) ve arkadaşları, genel yapıda 3-RPR düzlemsel paralel robotların kendinden hareketlerinin analizi ve çözümüne yönelik yaklaşımlar çalışmasını gerçekleştirmiştirler. Liu (2008) ve arkadaşları ise paralel Kinematik ile iki mafsallı takım kafası özelliğine sahip robot yapısının, oryantasyon yeteneğinin belirlenmesi, hata analizi ve boyut optimizasyonu üzerine çalışmalar yapmışlardır. Bununla birlikte, endüstriyel bir robotun kuvvet kontrolü için yeni bir optimal formüller çalışması Birgas (2008) ve arkadaşları tarafından gerçekleştirilmiştir. Diğer taraftan tıp alanında yapılan çalışmalar ise, Lessard ve arkadaşları tarafından, yeni bir tıbbi paralel robot tasarımı ve statik dengeleme optimizasyonu alanında yapılmıştır. Özellikle düzlemsel ve çözümü, 3 boyutlu analize göre kolay olabilecek bir çalışma ise, Briot ve Bonev (2007) tarafından, 3 serbestlik dereceli düzlemsel bir robotun doğruluk analizi konusunda gerçekleştirilmiştir. Bu çalışmada, yerel maksimum yönelim ve bölgede meydana gelen konum hatalarının ayrıntılı bir çalışmasını detaylı olarak sunulmuştur. Burada, girişlerde hatalara maruz kalan 3-DOF düzlemsel paralel robotlar. Yeterince uzak olduğunda tekillikler, yerel maksimum yönelim ve konum hataları yalnızca en az iki girdi zarar gördüğünde meydana gelebilir durumu maksimum hata durumuna göre incelenmiştir. Yine benzer bir çalışmanın farklı yaklaşımı, 3 serbestlik dereceli düzlemsel paralel robotların, doğruluk analizinde geometrik çözümü konusunda yapılmıştır.

Tombul ve Sarıtaş (2003) tarafından gerçekleştirilen bilimsel yayında, hazır üretilmiş beş eksenli bir Edubot robot üzerinde, ters kinematik hesaplamalar ve yörünge planlaması üzerinde araştırmalar yapılmıştır. Burada robot ait ters kinematik

hesaplamaları ve yörünge planlaması MATLAB 5.02 kullanılarak gerçekleştirilmiştir. Ayrıca, başlangıç ve hedef noktaları arasında, robotun geçmesi istenen noktalar dikkate alınarak, yörünge hesaplamaları tekrar yapılmıştır. Neticede, elde edilen sonuçlar Edubot robot üzerinde denenmiştir. Karahan (2007)'ın yüksek lisans tezi çalışmasında, Staubli RX-60 Robot'un dinamik modeli, Lagrange-Euler yöntemi kullanılarak elde edilerek, dinamik model ayrık parametreler cinsinden ifade edilerek bilinen parametreleri içeren matris ve bilinmeyen parametreleri içeren sistem parametre vektörü elde edildi. Bu vektörden, robotun dinamik modeli üzerinde herhangi bir etkiye sahip olmayan parametreler çıkarılarak, robotun bilinmeyen atalet parametreleri elde edilmiştir. Bıçakcı (2009)'ın yüksek lisans tez çalışmasında ise, insan şeklindeki robotun elektronik devre tasarımı ve imalatı yapılmıştır. Bununla birlikte, robotun matematiksel ifadeleri çıkarılıp, ilgili kontrol sisteminin yazılımı gerçekleştirilmiştir. Robotun anatomisi, insanın anatomisini andırmaktadır. Robotun toplam 23 serbestlik derecesi vardır ve her eklem bir DC motor tarafından sürülmektedir. Eklemler arası konumları ölçmek için manyetik artımsal encoder ve servo potansiyometre kullanılmıştır. Hamelin (2008) ve grup araştırmacıları, gerçek bir çevre ile etkileşen empedans kontrollü robotun döngü içinde donanım simülasyonu, konusunda araştırma yapmıştırlar. Bonev (2008) ve arkadaşları ise, kübik yapıda olmayan 3-RPR düzlemsel paralel robotlarda, paralel singüler durumlara geçmeden, montaj modlarını değiştirme, üzerine bir araştırma ve geliştirme bilimsel çalışmaları gerçekleştirmiştirler. Bigraz (2007) ve grubu da, bir KUKA tipi endüstriyel robot manipulator için gerçek zamanlı uygulama yeni bir formulasyon geliştirilmesi ve uygulaması üzerine çalışmalar yapmıştırlar. Figielski (2007) ve arkadaşları, optimal çalışma alanı kullanımı ile 2 serbestlik dereceli düzlemsel paralel endüstriyel robot manipülatörün geliştirlmesi konusunda çalışmalar gerçekleştirmiştirler. Lessard (2006) ve arkadaşları da, tıbbi 3 boyutlu ultrason görüntüleme için paralel robotun tasarımı üzerine deneysel uygulamalı bir çalışma yapmıştırlar. Bunun yanında, yine Lessard (2007) ve arkadaşları tıbbi 3 boyutlu ultrason görüntüleme için paralel robotun optimum statik dengelemesi üzerine deneysel uygulamalı bir çalışma ile geliştirilmiş uygulama çalışmaları yapmıştırlar. Bigras (1997) ve arkadaşları, esnek robot manipülatörlerin belirli çalışma alanında uyarlanabilir yörünge izleme belirlenmesi ve uygulanması konusunda araştırma çalışmaları gerçekleştirmiştirler. Bonev (1999a) ve arkadaşları, 6 serbestlik dereceli paralel robot manipülatörün oryantasyon çalışma alanı analizi konulu çalışma

yapmıştırlar. Yine Bonev (1999b) ve araştırma grubu, üç ekstra sensör kullanan paralel manipülatörlerin doğrudan kinematiğinin basit yeni bir çözüm tekniği ile kapalı form yaklaşımı konusunda araştırma konularını sunmuşturlar. Bununla birlikte, bu yapılan çalışmalar dışında Bonev (2000) ve çalışma grubu, 6-RUS tipi paralel manipülatörlerin sabit yönlü çalışma alanının hesaplanması için geliştirilen geometrik algoritma ve uygulaması konusunda aştırma yapmıştırlar.

Robot manipülatörlerin tekil (singüler) durumlarının giderilmesin yönelik bilimsel çalışmalar da yapılmıştır. Benzer şekilde, Bonev ve Gosselin (2001), döner eklemli düzlemsel paralel manipülatörlerin tekil durumluk odakları üzerine araştırma çalışmalarını bir uluslararası konferansta sunmuşturlar. Lupien (2001) ve arkadaşları ise farklı bir konuda, zorlu ortamlarda hareket eden, yürüyen bir robot için çok yönlü tasarım platform geliştirilmesi konusunda çalışmıştırlar. Zlatanov (2002) ve çalışma grubu, robot manipülator yapısına sahip, paralel mekanizmaların kısıt tekilliklerinin araştırılması konusunda çalışmalar gerçekleştirmiştirler. Bonev ve Gosselin (2002), farklı bir yaklaşım tekniğiyle, sabit yönlendirmeli çalışma alanı ve özel 6-RUS bir paralel robot manipülatörün tekillik yüzeylerinin hesaplanması için geometrik algoritmaların geliştirilmesi konusunda çalışmıştırlar.

Robot manipülatörlerin, kontrolcü tasarımı ve analizi konusunda da bazı çalışmalar yapılmıştır, Lessard (2004) ve araştırma grubu, LMI optimizasyonunu kullanarak tıbbi uygulama için endüstriyel Robotlarda gürbüz konum / kuvvet kontrolcü tasarımı ve uygulanması konusunda araştırma yapmıştırlar. Janvier (2005) ve arkadaşları, tıbbi 3 boyutlu ultra-sound görüntüleme için bir robotun optimum performansının değerlendirilmesi üzerine araştırma çalışmalarını uluslararası bir konferansta sunmuşturlar. Chablat (2006) ve grubu ise özel olarak tasarlanmış 3 eklemli RPR düzlemsel paralel robotun kendinden hareketleri konusunda çalışmalar yapmıştırlar. Poire (2006) ve arkadaşları ise, daha çok simülasyon destekli, grafik simülasyon ortamından robotik çerçeveye kontrol kanunlarının geçiş işlemleri temel başlığında araştırma yapmıştırlar.

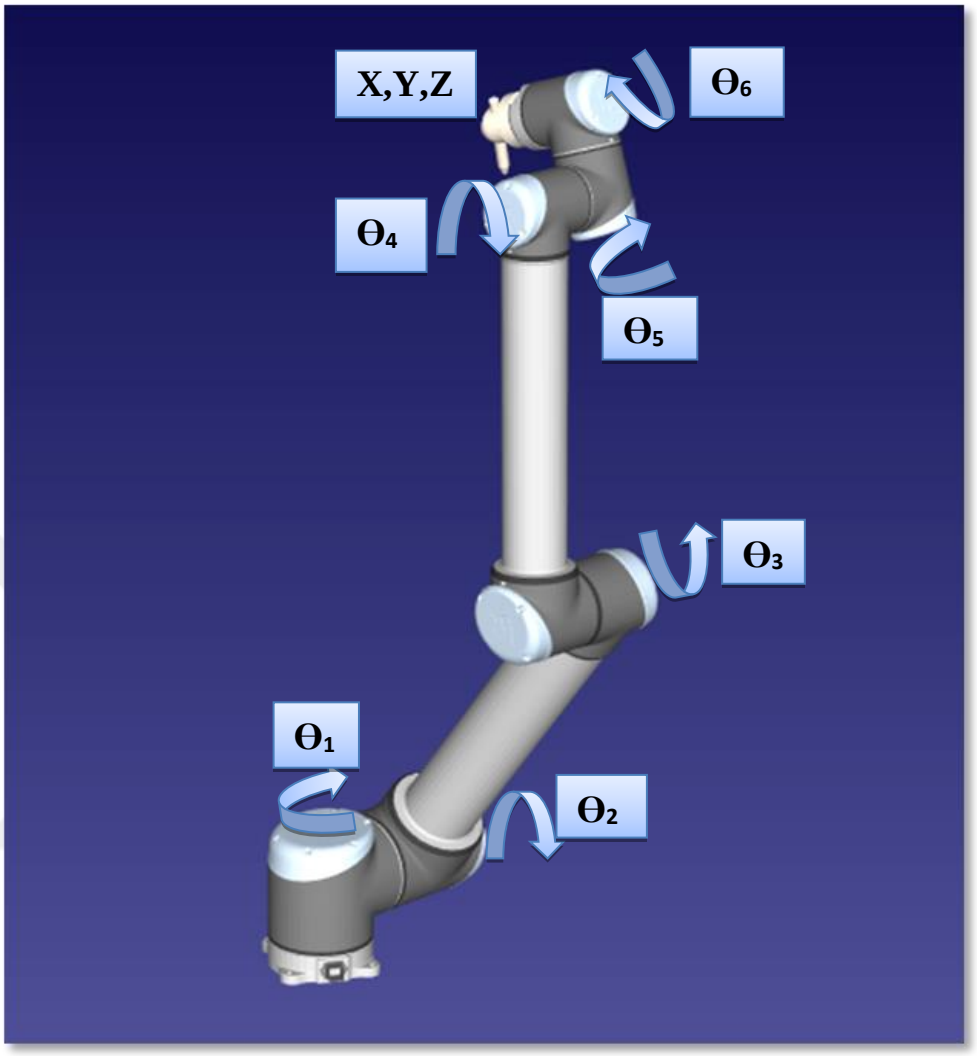
# 2. BÖLÜM
# GEREÇ VE YÖNTEM

## 2.1. Endüstriyel Robot Manipülatör

Klasik operator esaslı jeans pantolanlara kimysal püskürtme işlemlerinde, sağlık problemleri, kumaşta renk tonu farklılıkları gibi istenmeyen durumlar oluşmaktayadı (Şekil 2.1.) Bu bölümde yapılan simulasyon çalışmalarında 6 serbestlik dereceli endüstriyel bir robot özelliğine sahip, robot manipülatörün en optimum yörüngeyi izleyerek kot jeans pantalonlara kimyasal püskürterek renk değişimini sağlaması amaçlanmıştır. Belirgin oranda, robot uç işlevcisi tarafından yörünge hataları olması durumunda ürün olan kot pantalonların renk değişiminde alan bölgesi renk farklılıkları oluşmaktadır. Bu durumda ürünün ıskartaya ayrılması söz konusu olabilmektedir. Bu simülasyon çalışmasında kinematic analizi gerçekleştirilen 6 serbestlik dereceli endüstriyel robotunm tasarlanmış hali Şekil 2.2. de gösterildiği gibidir.



*Şekil 2.1.* Operatör esaslı jeans kot pantolon kimyasal püskürtme işlemi görünümü




*Şekil 2.2*. 6 Serbestlik dereceli endüstriyel robot ve dönme eksenleri

Ayrıca Tablo 2.1.' de verildiği gibi robot hassasiyet ve hız olarak belirgin oranda üretim hızına ve hassasiyetine katkı sağlayacak özelliktedir. Bununla birlikte, robot manipülatörün tüm eklemlerinde +/- 360° özelliğe sahip olması çalışma uzayında, ulaşabileceği noktalar maksimum seviyededir. Robotik sistemin gürültüsüz çalışması da bir üretim hattında onlarca robot kullanılması durumunda, bozucu etken olarak sistemin performanısınısa etkilemektedir.

Tablo 2. 1. Endüstriyel Robot manipülatörün teknik özellikleri ve parametreleri (https://www.universal-robots.com/tr/urunler/ur5-robot/

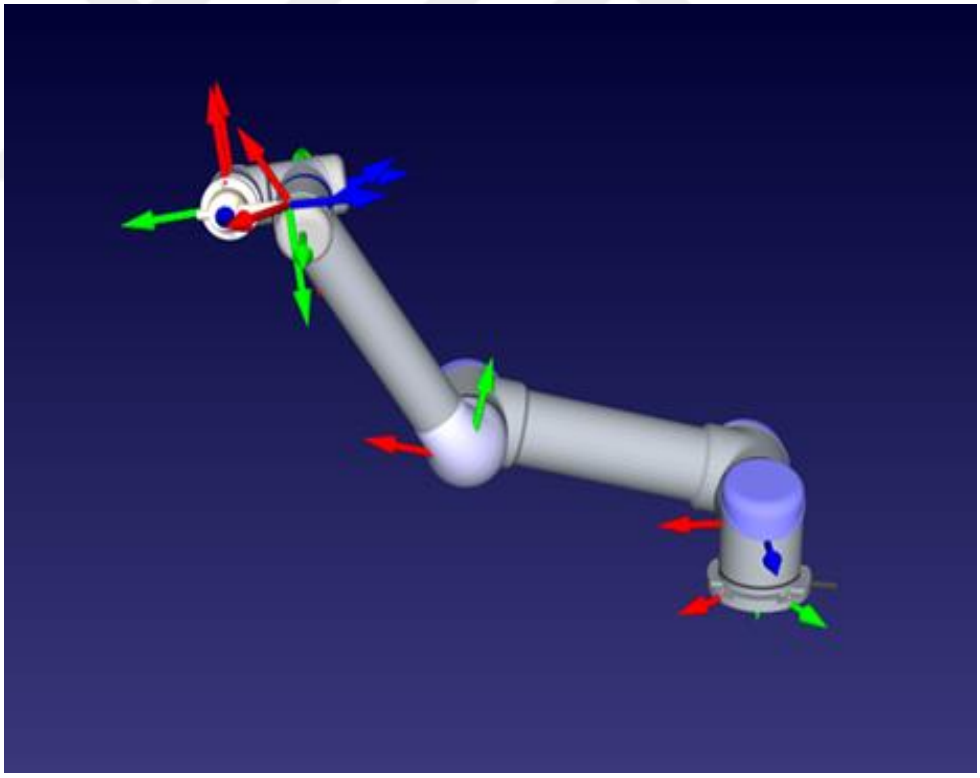
| | |
|---|---|
| ***Ağırlık*** | 18,4 KG |
| ***Taşıma Kapasitesi*** | 5 KG |
| ***Menzil*** | 850 mm. |
| ***Eklem Dönüş Menzilleri*** | Tüm eklemlerde +/- 360° |
| ***Hız*** | Eklem - saniyede maksimum 180°<br>Araç - saniyede yaklaşık 1 m |
| ***Tekrarlama*** | +/- 0,1 mm |
| ***Kapladığı Yer*** | Ø 149 mm |
| ***Eksen Sayısı*** | 6 dönen eklem |
| ***Kontrol Kutusu Boyutu*** | 475 mm x 423 mm x 268 mm (G x Y x D) |
| ***I/O Bağlantı Noktaları*** | 18 dijital giriş, 18 dijital çıkış, 4 analog giriş, 2 analog çıkış |
| ***I/O Güç Kaynağı*** | Kontrol kutusunda 24 V 2A ve araçta 12 V/24 V 600 mA |
| ***İletişim*** | TCP/IP — Ethernet yuvaları<br>Modbus TCP |
| ***Programlama*** | 12 inç dokunmatik ekranda poliskop grafik kullanıcı arayüzü |
| ***Gürültü*** | Göreceli olarak gürültüsüz |
| ***IP Sınıflandırması*** | : IP54 |
| ***Güç Tüketimi*** | Normal bir program kullanıldığında yaklaşık 200 W |
| ***İşbirliği İşlemi*** | EN ISO 13849:2008 PL d ve EN ISO 10218-1:2011'in 5.4.3 fıkrasına uygun olarak test edilmiştir |
| ***Malzemeler*** | Alüminyum, PP plastik |
| ***Sıcaklık*** | Robot 0-50°C arasında bir sıcaklıkta çalışabilir |
| ***Güç Kaynağı*** | 100-240 VAC, 50-60 Hz |

Şekil 2.3'de endüstriyel robot manipülatörün tasarlanmış hali ile birlikte koordinat eksenlerinin gösterilimi detaylandırımıştır. Şekilden ve Tablo 2.1'de verildiği gibi robotun her mafsalının +/- 360° eklem dönme açılarına sahip olmasından dolayı hareket kabiliyeti ve esnekliği çok iyi düzeydedir.

Robot manipülatörün kinematic parametrelerini gösterir Denavit-Hartenberg (D-H) parametreleri detaylı olarak Tablo 2.2' de verilmiştir. Tablo 2.2'den da görüleceği gibi robot manipülarörün her eklemi döner mafsal özelliğindedir. Herhangi bir eklem kaçıklığı ve 2. komşu eksen açısı yoktur. Bundan dolayı bu yapının özellikle tam poziyonlamada çok iyi düzeyde avantaj sağlayacağı belirgindir. Burada gövde, omuz, dirsek ve bilek eklemleri olarak toplam 6 adet mafsaldan oluşmaktadır.

Tablo 2.2. Endüstriyel robot manipülatörün D-H parametreleri

| *Eksen No* | *2 Komşu Eksen Açısı* | *Uzuv Uzunluğu* | *Eklem Kaçıklığı* | *Eklem Açısı* | *Eklem Değişkeni* |
|---|---|---|---|---|---|
| *i* | $\alpha_{i-1}$ | $a_{i-1}$ | $d_i$ | $\Theta_i$ | $d_i$ *veya* $\Theta_i$ |
| 1 | *0* | $L_1$ | 0 | $\Theta_1$ | $\Theta_1$ |
| 2 | *0* | $L_2$ | 0 | $\Theta_2$ | $\Theta_2$ |
| 3 | *0* | $L_3$ | 0 | $\Theta_3$ | $\Theta_3$ |
| 4 | *0* | $L_4$ | 0 | $\Theta_4$ | $\Theta_4$ |
| 5 | *0* | $L_5$ | 0 | $\Theta_5$ | $\Theta_5$ |
| 6 | *0* | $L_6$ | 0 | $\Theta_6$ | $\Theta_6$ |



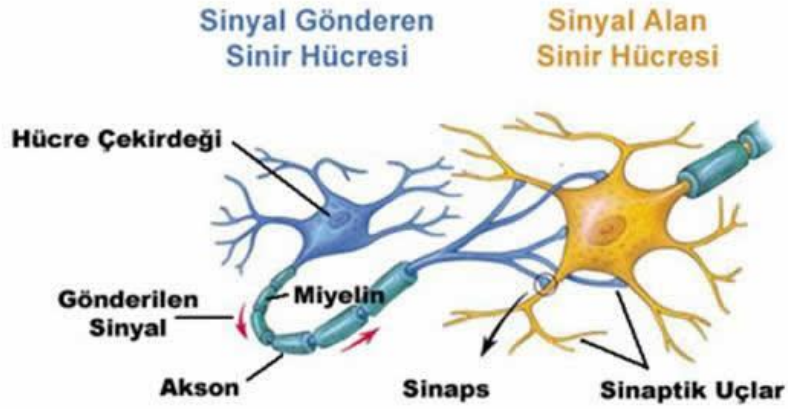
*Şekil 2.3.* Robot manipülator ve eksen takımlarının görünümü

## 2.2. Tasarlanmış Yapay Sinir Ağı

Yapay Sinir Ağları kısaca YSA olarak adlandırırsak; insan beyninin çalışma yapısını taklit ederek geliştirilen yapay zeka tekniklerinden sadece biridir [Haykin,2008]. YSA biyolojik sinir ağlarının modellenmesi olduğu için, öncelikle biyolojik sinir sisteminin yapısına bakmak en uygunu olur. Biyolojik sisnir sisteminin temel yapısını nöronların (hücre) yapısı dört ana bölümden oluşmaktadır; dentrit, çekirdek, akson ve bağlanıları. Dentritler, daha çok haberleşme ağı olarak nöronlardan veya duyu organlarından gelen sinyalleri çekirdeğe iletmektedir. Çekirdek, dentritler tarafından gelen sinyalleri toplar ve diğer hücrelere aksonlar vasıtasıyla iletir. Bu işlemlerin benzetim işlemleri YSA yapısıyla gerçekleştirilmektedir. Şekil 2.4'de insan beyni sinir ağı yapısı ve işlemleri detaylandırılmıştır.Aktivasyon fonksiyonu kullanılarak, hücreye gelen toplam girdiyi işlenerek hücrenin bu girişe karşı üreteceği çıkış sinyali belirlenir. Aktivasyon fonksiyonu olarak, doğrusal, adım, hiperbolik sigmoid ve hiperbolik tanjant gibi fonksiyonlar kullanılır. Yapay sinir ağları, biyolojik sinir sistemlerinde gözlenen prensiplere dayanan bir yapıya sahip doğrusal olmayan haritalama sistemleridir. Şekil 2.4. 'de görülebildiği gibi oldukça basitleştirilmiş terimlerle, tipik bir gerçek nöron, sınırlı bir alanda diğer birçok nörondan sinyal toplayan dallı bir dendritik ağaca sahiptir; toplanan sinyalleri entegre eden ve bir yanıt sinyali üreten (metabolik fonksiyonları yöneten) bir hücre gövdesi; ve yanıtı, diğer birçok nöronun dendritik ağaçları ile temas yoluyla dağıtan uzun bir dallanma aksonu. Her bir nöronun yanıtı, girdilerinin nispeten basit doğrusal olmayan bir fonksiyonudur ve büyük ölçüde girişlerinden gelen bağlantıların gücü ile belirlenir. Münferit birimlerin göreli sadeliğine rağmen, birçok nöron içeren sistemler karmaşık ve ilginç davranışlar üretebilir.

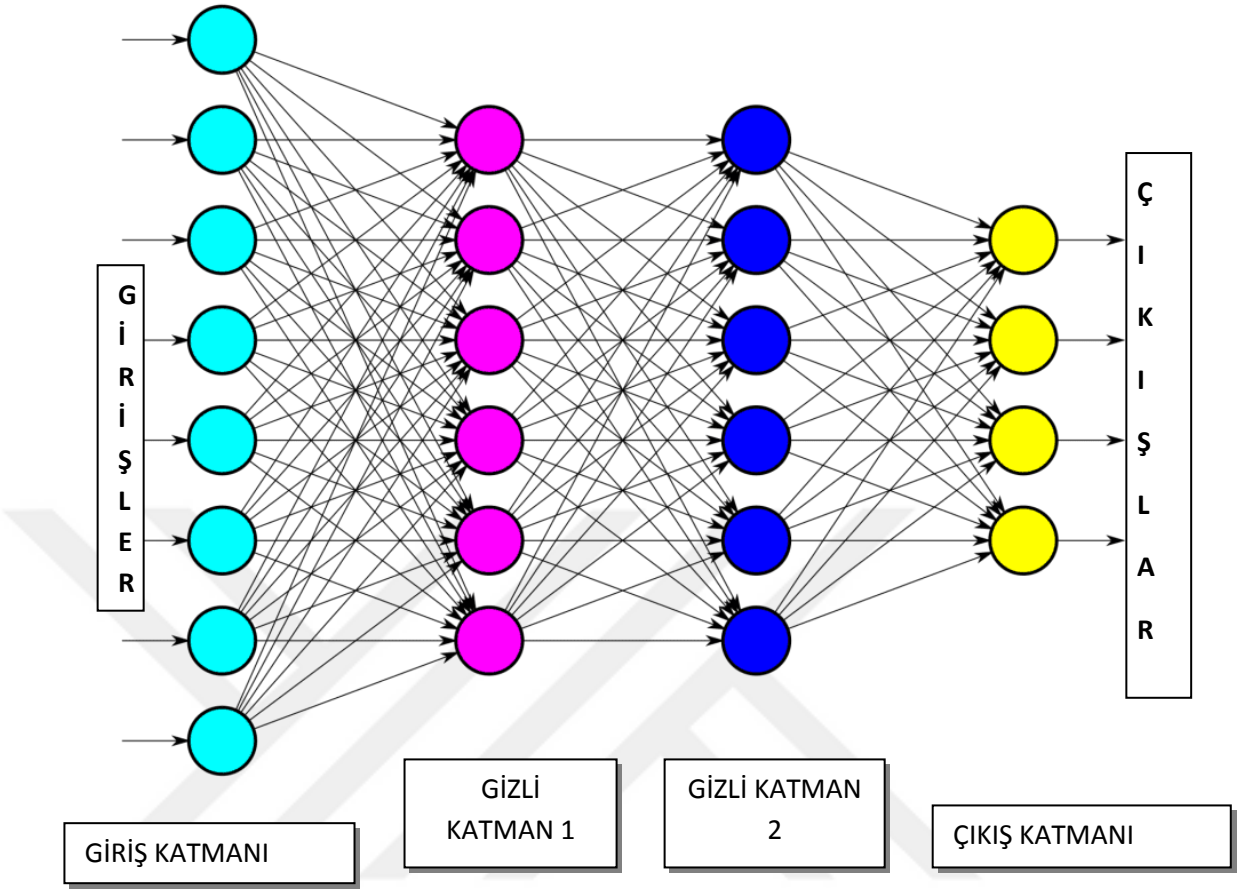
YSA bu fikirlere dayanmaktadır. En genel terimlerle, bir sinir ağı, ağırlıklı bağlantılarla bağlanmış çok sayıda basit işlemciden oluşur. Benzetme yoluyla, işlem düğümleri 'nöronlar' olarak adlandırılabilir. Her düğüm çıkışı, yalnızca dahili olarak depolanan veya ağırlıklı bağlantılar yoluyla gelen düğümde yerel olarak bulunan bilgilere bağlıdır. Her birim diğer birçok düğümden girdi alır ve çıktısını henüz diğer düğümlere iletir. Tek başına, tek bir işleme elemanı çok güçlü değildir;

girişlerinin basit doğrusal olmayan bir fonksiyonu olan tek bir sayısal değere sahip bir skaler çıkış üretir. Sistemin gücü birçok ünitenin birleşiminden uygun bir şekilde ortaya çıkar.



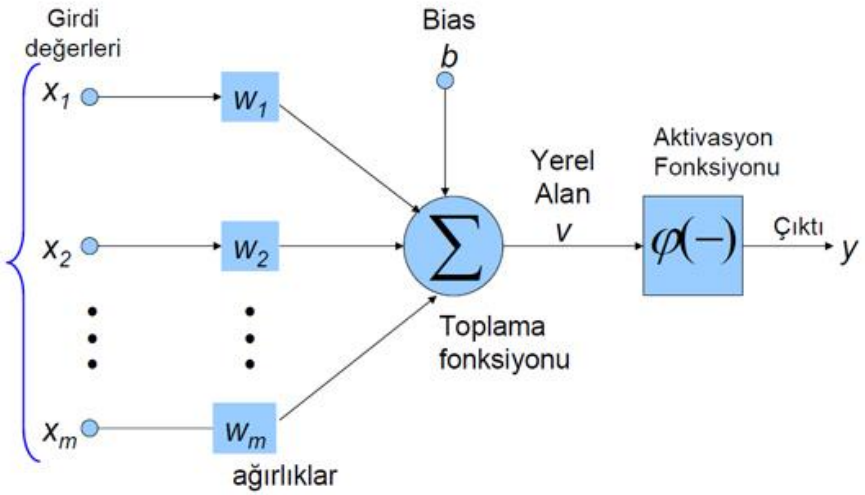

*Şekil 2.4.* İnsan beynindeki hücre yapısı ve bileşenleri

Genel yapıda YSA'lar giriş katmanı, gizli katmaları ve çıkış katmalarından oluşur. Fakat, modellecek yada control edilecek sistemin komplekslik yapısına göre gizli katman sayısı ve nöron sayıları ona göre tercih edilir. Şekil 2.5'de bu yapının gizli katman sayısı iki olan YSA 'nın yapısı şematik olarak gösterilmiştir. Burada giriş katmanı hücrelerindeki nöronlar daha çok lineer yapıda olurlar. Gizli ve Çıkış katmanlarındaki hücreler daha çok lineer olmayan işlemci fonksiyonlarından oluşurlar.




*Şekil 2.5*. Temel Yapay Sinir Ağlarının yapısının şematik gösterilişi

Bir ağ, bağlantı topolojisini ve bağlantı ağırlıklarının değerlerini değiştirerek farklı işlevleri yerine getirmek için uzmanlaşmıştır. Karmaşık fonksiyonlar, ünitelerin uygun ağırlıklarla birleştirilmesiyle uygulanabilir. Aslında, uygun bir yapıya ve seçilen ağırlığa sahip yeterince büyük bir ağın, belirli geniş kısıtlamaları karşılayan herhangi bir işleve keyfi bir doğrulukla yaklaşabileceği gösterilmiştir (Şekil 2.6).




*Şekil 2.6*. Yapay Sinir Ağının Çalışma İşlemlerinin Blok diagramı

Genellikle, işlemci birimleri aşağıdaki denklemdeki gibi cevaba sahiptir;

$$y = f\left(\sum_i u_i\right) \tag{1.16}$$

burada $u_i$ gizli katmanın çıkış katmanına çıkış sinyalleridir, *f(.)* sigmoid veya lojistik fonksiyon gibi basit doğrusal olmayan bir fonksiyondur. Bu birim, girdilerinin ağırlıklı bir doğrusal kombinasyonunu hesaplar ve skaler bir çıktı üretmek için bunu doğrusal olmayanlıktan geçirir. Genel olarak, sınırlı doğrusal olmayan, doğrusal olmayan bir işlevdir; lojistik fonksiyon ortak bir seçimdir.

Bu model, elbette, gerçek sinir sistemlerinin büyük ölçüde basitleştirilmiş bir yaklaşımıdır. Amaç, biyolojinin getirdiği fiziksel kısıtlamalar hakkında çok fazla değişiklik yapmadan gerçek ağların bilgi işleme fonksiyonlarında önemli olan temel özellikleri yakalamaktır.

### 2.2.1. Geri Yayılım (GY) Öğrenme Algoritması

GY, YSA'lar için en yaygın kullanılan eğitim algoritmasıdır. Ağırlıklar aşağıdaki gibi ayarlanır;

$$\Delta w_{ij}(t) = -\eta \quad \frac{\delta E(t)}{\delta w_{ij}(t)} + \alpha \Delta w_{ij}(t-1) \tag{2.1}$$

Burada η öğrenme oranı ve α momentum terimidir. Diğere taraftan, çevrimiçi GY, YSA'ya desen sunulduktan sonra ağırlıkları günceller. Her GY döneminden önce rastgele seçilen girdi düzeni sırasına sahip çevrimiçi GY. Bu öğrenme sürecini stokastik hale getirir ve çoğu durumda tercih edilir.

### 2.2.2. Random Geri Yayılım (RGY) Öğrenme Algoritması

Bu ağırlık güncellemelerinin büyüklüklerine değil yerel gradyanların işaretine dayandığı uyarlanabilir bir öğrenme oranı yöntemidir. Her ağırlığın, $\Delta w_{ij}$, kendi adım boyutu veya güncelleme değeri vardır, $\Delta w_{ij}$, t zamanına göre değişiklik gösterir:

$$\Delta_{ij} \begin{cases} \eta^{+} \ \Delta_{ij}(t-1), if \ \frac{\delta E(t-1)}{w_{ij}(t-1)} \frac{\delta E(t)}{w_{ij}(t)} > 0 \\ \eta^{-} \ \Delta_{ij}(t-1), if \ \frac{\delta E(t-1)}{w_{ij}(t-1)} \frac{\delta E(t)}{w_{ij}(t)} < 0 \\ \Delta_{ij}(t-1), \ else \end{cases} \tag{2.2}$$

Burada $0 < \eta^{-} < 1 < \eta^{+}$. Ağırlıklar aşağıdaki denkleme göre ayarlanır:

$$\Delta_{ij}\begin{cases} -\Delta_{ij}(t), if\ \frac{\delta E(t)}{w_{ij}(t)} > 0 \\ +\Delta_{ij}(t), if\ \frac{\delta E(t)}{w_{ij}(t)} < 0 \\ 0,\ else \end{cases} \quad (2.3)$$

### 2.3.3. Hızlı Geri Yayılım (HGY) Öğrenme Algoritması

HGY, aşağıdaki varsayımlara dayanan başka bir eğitim yöntemidir, her ağırlık için *E(w)* yukarı doğru açılan bir parabolle yaklaştırılabilir ve ağırlığı için *E (w)* eğimindeki değişiklik, aynı anda değişen diğer ağırlıklardan etkilenmez zaman. Ağırlıkları değiştirme ve güncelleme kuralı aşağıdaki eşitlikle açıklanabilir;

$$\Delta w(t) = \frac{S(t)}{S(t-1)-S(t)}\Delta w(t-1) - \eta\, S(t) \quad (2.4)$$

Burada $S(t) = \frac{\delta E(t)}{\delta w(t)}$ , Pay, ağırlığa göre hatanın türevidir ve $\frac{S(t-1)-S(t)}{\Delta w(t-1)}$ , ikinci türevin sonlu bir fark yaklaşımıdır. Tek boyutlu bir fonksiyonu (x) en aza indirmek için bu yaklaşık Newton yöntemi birlikte: $(x)$: $\Delta x = \frac{-f'(x)}{f''(x)}$. Sonsuz bir geri adımdan veya geri doğru yokuş yukarı adımdan kaçınmak için, maksimum büyüme faktörü parametresi **μ** tanıtıldı. Ağırlık değişikliğinin önceki ağırlık değişiminin **μ** katından daha büyük olmasına izin verilmesi söz konusu değildir. Ayrıca *QP*, soruna uyması için seçilmesi gereken sabit bir öğrenme parametresi olan **η**'a sahiptir.

### 2.3.4. Delta –Bar-Delta (DBD) Öğrenme Algoritması

Her ağırlığın kendi öğrenme oranına sahip olduğu uyarlanabilir bir öğrenme oranı yöntemi şeklidir. Öğrenme oranları, gradyan işaretine göre güncellenir, ardışık yinelemelerde işaretleri değiştirmez, ardından adım boyutu doğrusal olarak artar. Gradyan işaretleri değiştirirse, öğrenme hızı katlanarak azalır. Bazı durumlarda bu

yöntem, uyarlanabilir olmayan yöntemlerden çok daha hızlı öğreniyor gibi görünmektedir. Öğrenme oranları **η** (t) aşağıdaki şekilde ifade edilir;

$$\Delta_{ij}\begin{cases}\kappa & if\ \ \delta'(t-1)\,\delta(t) > 0\\ -\phi\boldsymbol{\eta}(\mathbf{t}) & if\ \delta'(t-1)\,\delta(t) < 0\\ 0 & else\end{cases} \tag{2.5}$$

Burada δ(t) = $\frac{\delta E}{\delta w}$ *t* zamanında ve δ geçmiş değerlerinin üstel ortalaması olarak belirelenir. *δ'(t)=(1-θ) δ(t)+ θ δ'(t-1)*

## 2.3. İleri Beslemeli Yapay Sinir Ağları (YSA)

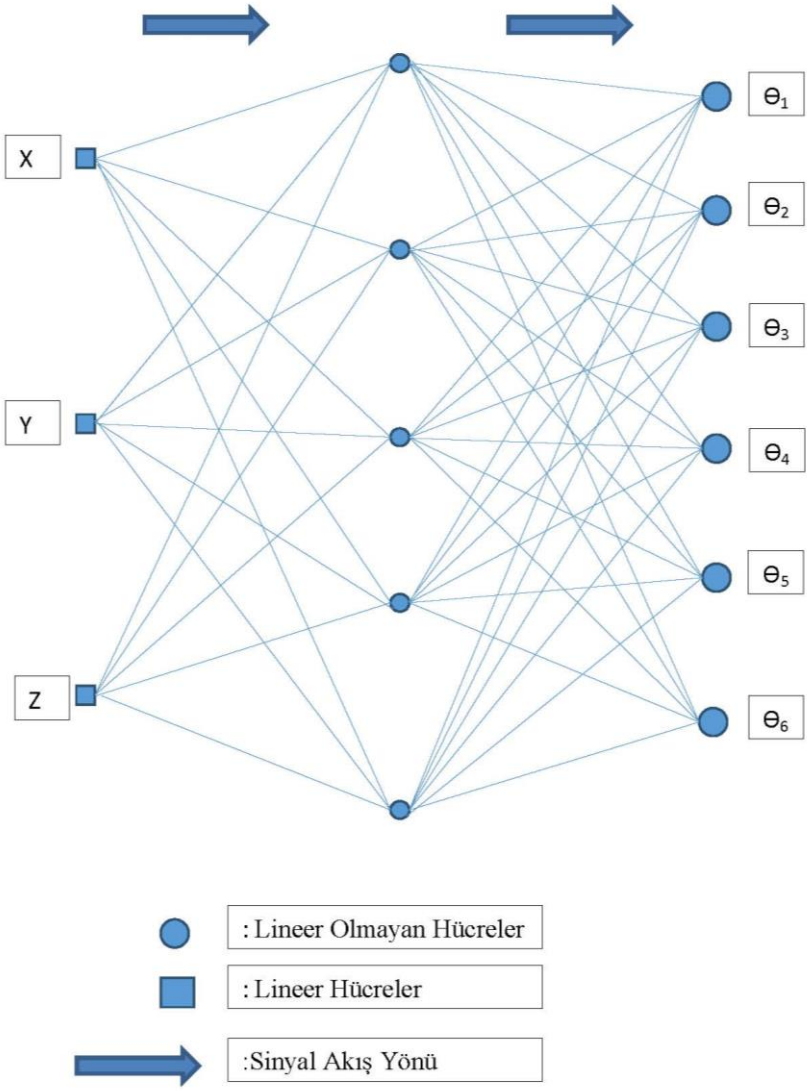
İleri beslemeli sinir ağlarında yapay nöronlar (düğümler veya işlem birimleri olarak da adlandırılır) ileri beslemeli bir şekilde düzenlenir (genellikle katmanlar şeklinde, yani her bir nöron dış ortamdan ve / veya diğer nöronlardan bir girdi alabilir, ancak geri bildirim yoktur. Standart bir ileri beslemeli sinir ağı basit işlem birimlerinden oluşur (dinamik elemanlar olmadan) Bir ileri beslemeli ağ, bazı girdi düzenlerine yanıt olarak bir çıktı paterni hesaplar Eğitildikten sonra (sabit bağlantı ağırlıkları ile) belirli bir girdi paternine çıktı yanıtı verilir Bu, ileri beslemeli sinir ağının gerçek bir dinamik göstermediği ve bu tür ağlarda herhangi bir kararlılık sorunu bulunmadığı anlamına gelir. İleri beslemeli ağlar için dinamikler genellikle tek bir anlık doğrusal olmayan haritalamaya basitleştirilir.

Robotik manipulator sisteminin uç işlevcisinin (end-effector) belirlenen yörüngesi yardımıyla her bir mafsalın gönme açılarının analizi için 2 tip yapay sinir ağı modeli kullanılmıştır, bu modellerde sadece gizli katmada kullnılan hücre sayıları değiştirilerek analiz gerçekleştirilmiştir. Birinci modelled hğcre sayısı olarak 5 hücre kullanılarak analiz, farklı aktivasyon fonkiyonları ile gerçekleştirilmiştir (Şekil 2.7.). Şekil 2.7'de de görüleceği gibi yapay sinir ağı modeli 3 katmandan oluşmaktadır, bu katmlara giriş katmanı, gizli katman ve çıkış katmanı şeklindedir. Gizli katmanda lineerolamyan 5 hücre kullanılmıştır. Bir önceki bölümde detaylı olarak açıklandığı gibi dört farklı Öğrenme Algoritması kullanılmıştır. Bu kullanılan algoritmalar;

- Geri Yayılım (GY) Öğrenme Algoritması;
- Random Geri Yayılım (EGY) Öğrenme Algoritması
- Hızlı Geri Yayılım (HGY) Öğrenme Algoritması
- Delta –Bar-Delta (DBD) Öğrenme Algoritması

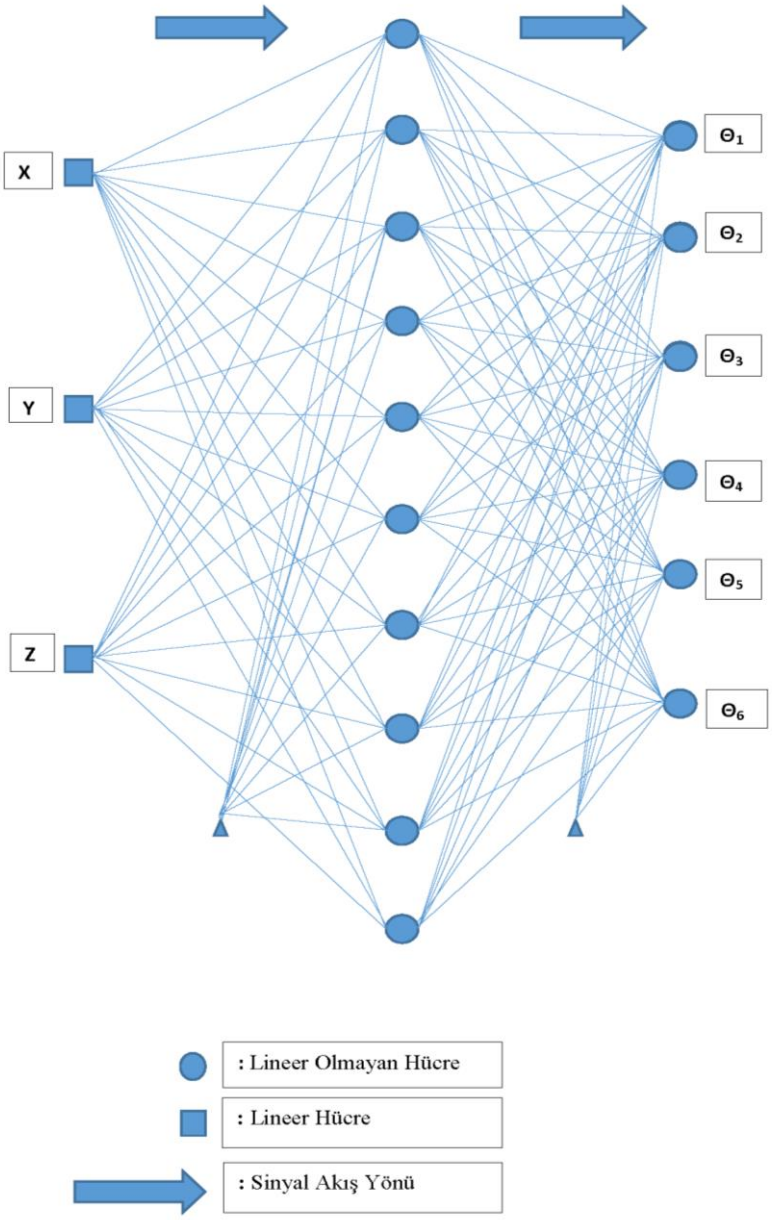
Şeklindedir.

Bununla birlikte, yapay sinir ağı modelinin gizli katmanında kullanılan hüzre sayısını artırarak performans özelliğini denenmesi gerçekleştirilmiştir. Burada hücere sayısı 10 olarak kullanılmıştır. Bu yapının şematik görünümü Şekil 2.8' de detaylı verilmiştir.



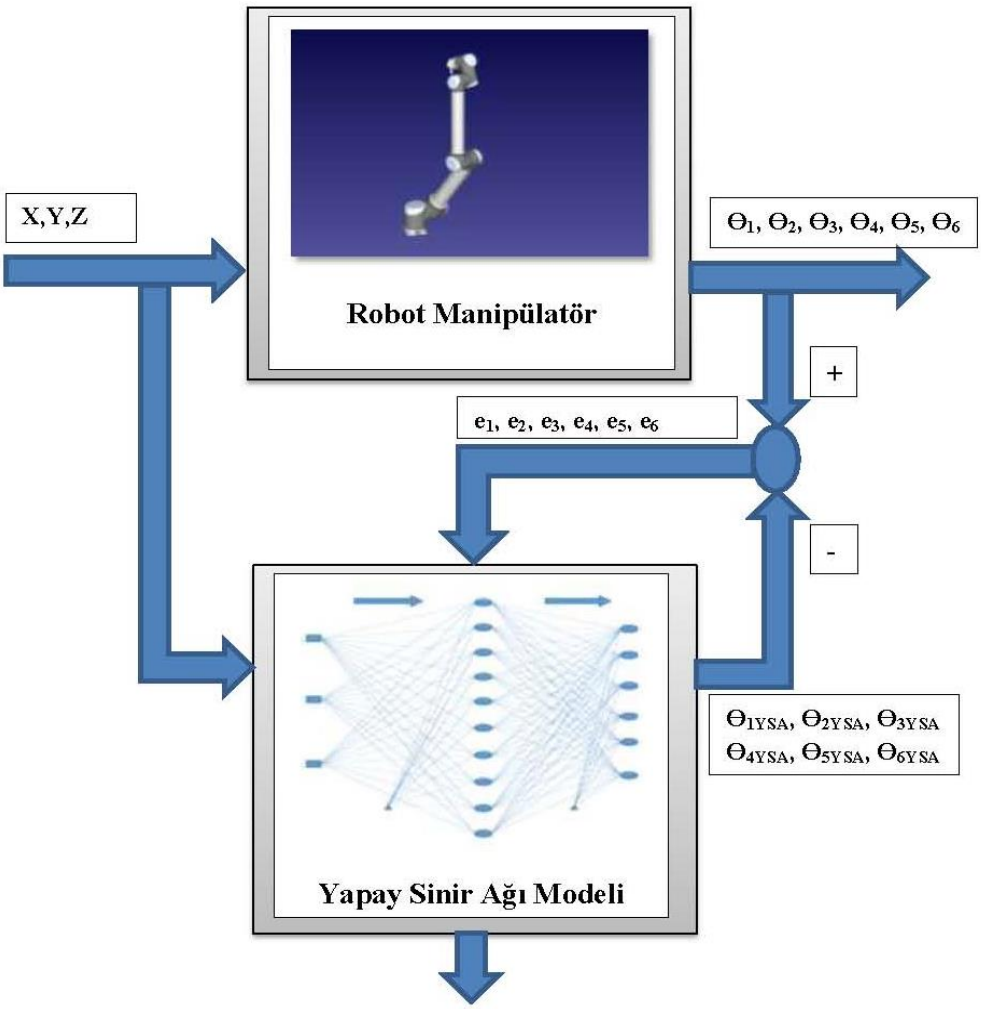

*Şekil 2.7*. İleri beslemeli, 3-5-6 yapay sinir ağı modelinin şematik görünümü

Bununla birlikte, yapay sinir ağı modelinin gizli katmanında kullanılan hücre sayısını artırarak performans özelliğini denenmesi gerçekleştirilmiştir. Burada hücere sayısı 10 adet lineer olmayan aktivasyon fonksiyonundan  katman olarak kullanılmıştır. Bu yapının, 3 giriş hücreli, 10 gizli katman hücreli ve 6 çıkış katmanı hücreli, şematik görünümü Şekil 2.8' de gösterildiği gibidir.



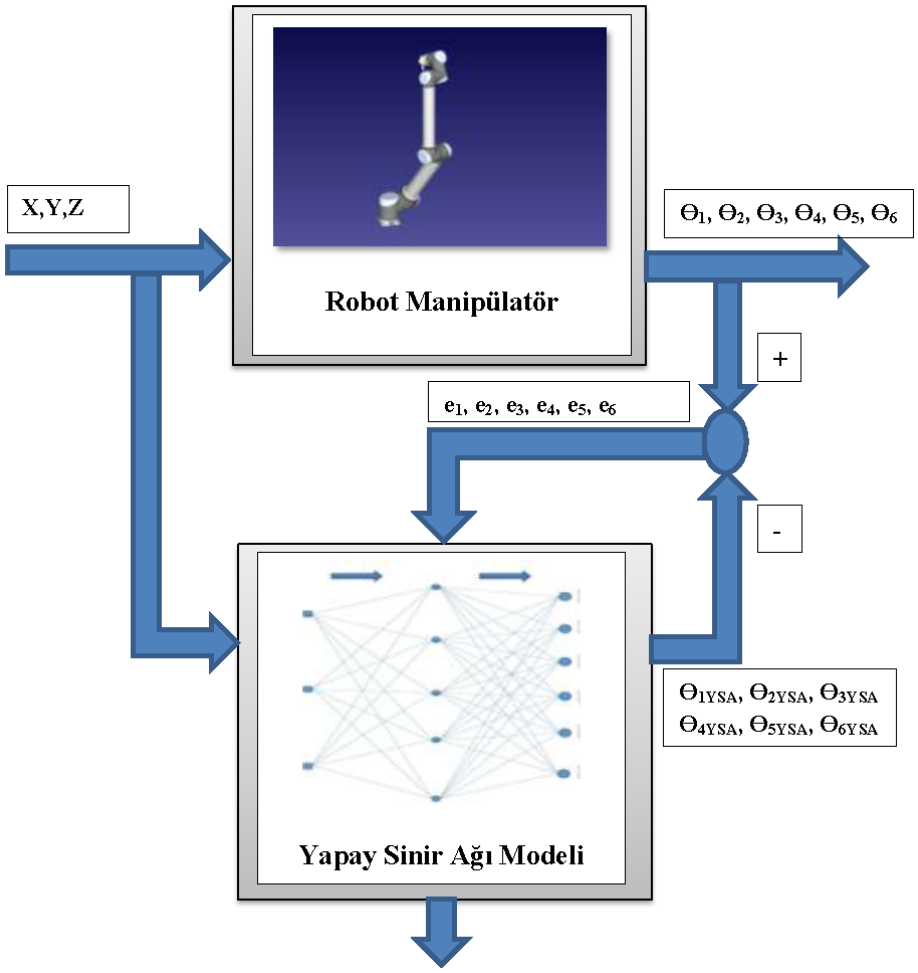

*Şekil 2.8*. İleri beslemeli, 3-10-6  yapay sinir ağı modelinin şematik görünümü

Yapay sinir ağı esaslı, robot manipülatörün özellikle belirlenen yörünge yardımıyla, maksallarda oluşabilecek açısal değişimlerin belirlenmesi için gerçekleştilen analizin yapısı Şekil 2.9'de blok diagram halinde verilmiştir. Burada, yapay sinir ağı farklı öğrenme algoritmaları kullanılarak, analiz gerçekleştilmiştir. Şekilden de görüleceği gibi gerçek açısal değerler ile yapay sinir ağı sonucu lede edilen açısal değişimlerin farkı olan hatalar yardımıyla öğrenme işlemi gerçekleştilmiştir. Ayrıca, daha iyi performans analizi etkisini göremek ve mukayse etmek için gizli katmanda 10 date lineer olmayan aktivasyon özelliğine sahip hücrelerin kullanılması tercih edilmiştir. Burada yapay sinir ağı yapısının 3-10-6 şeklinde olduğu görülmektedir.




*Şekil 2.9.* **3-10-6** tipi yapay sinir ağı kullanılarak robot manipülatörün modellenmesi

Bunun yanında, modelleme etkisinin daha hızlı olmasını ve önceki yapıya göre mukayese edebilmek için, Şekil 2.10'de görüldüğü gibi, gizli katmandaki hücre sayısı daha az kullanılmıştır. Şekil 2.9'de ki yapıya benzer, fakat yapay sinir ağı yapısı biraz daha basitleştilmiştir. Bilindiği gibi, öğrenme esnasında gizli katmadaki hücre sayısı artırıldıkça, buna oranla öğrenme hızı ters orantılı olarak azalmaktadır. Özellikle gerçek zamanlı robot manipulator üzerinde uygulanması durumunda hızlı cevap çok önem arz etmektedir. Yörünge izlerken mafsal açılarının hızlı cevabı, kumaş taşlama esasında binlerce ürün geçmesi durumunda enerji sarfiyatı ve hız öne plana çıkmaktadır.




*Şekil 2.10.* **3-5-6** tipiYapay sinir ağı kullanılarak robot manipülatörün modellenmesi

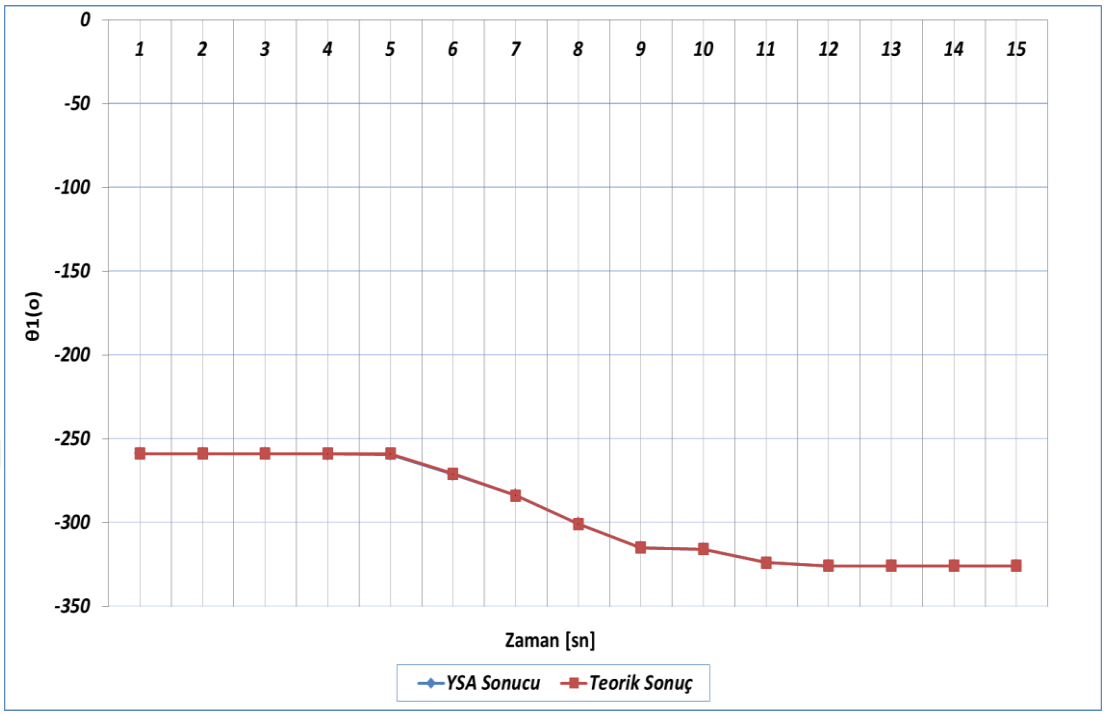
# 3. BÖLÜM

# BULGULAR

Bu bölümde belirlenen endüstriyel jeans kimyasall püskürtme işleminde kullanılan robot manipülatörün, ters kinematic analizinde açısal konum değişiminin tahminlenmesi ileri beslemeli YSA yapısı kullanılarak, simülasyon çalışması gerçekleştilmiştir. Ayrıca, en optimum YSA modelini belirleme ve mukayese açısından değerlendirme için farklı yapıda dört öğrenme algoritması kullanıraka YSA modeli test edilmiştir.

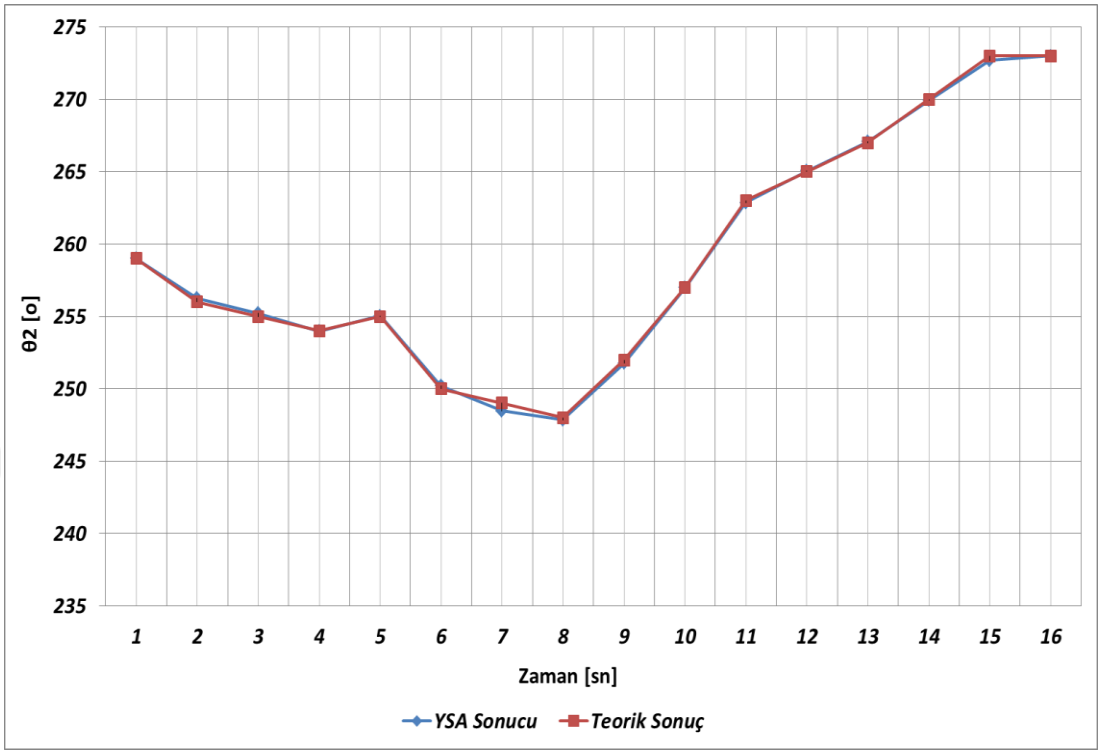
## 3.1. Tasarlanmış Yapay Sinir ağları ile Tahminleme

Bu bölümde özellikle, bölüm 2'de verilen Şekil 2.5.'deki blok diagram yapısı kullanılarak, simülasyon çalışmaları ve analizleri gerçekleştirilmiştir. Blok diyagramından da görüleceği gibi, yapay sinir ağı yapısı 3 giriş hücresi (robot uç işlevcinin poziyonu, X,Y,Z) 10 gizli katman hücresi ve 6 çıkış katmanı hücresinden, robot manipulator dönme açısal değişimleri, ($\Theta_1$ , $\Theta_2$ ,$\Theta_3$ , $\Theta_4$ , $\Theta_5$ , $\Theta_6$) oluşmaktadır. Şekil 3.1.'de 1 nolu Mafsalın açısal grafiksel değişimi detaylı verilmiştir. Grafikten de görüleceği gibi, teorik açısal değişimi ile delta-bar-delta öğrenme algoritması esaslı olarak, Yapay Sinir Ağları (YSA) sonucu birbirini izlemektedir. 5. ve 9. Saniyede ani açısal değişim olmasına rağmen, YSA tahminlemesi çok iyi sonuç vermiştir.



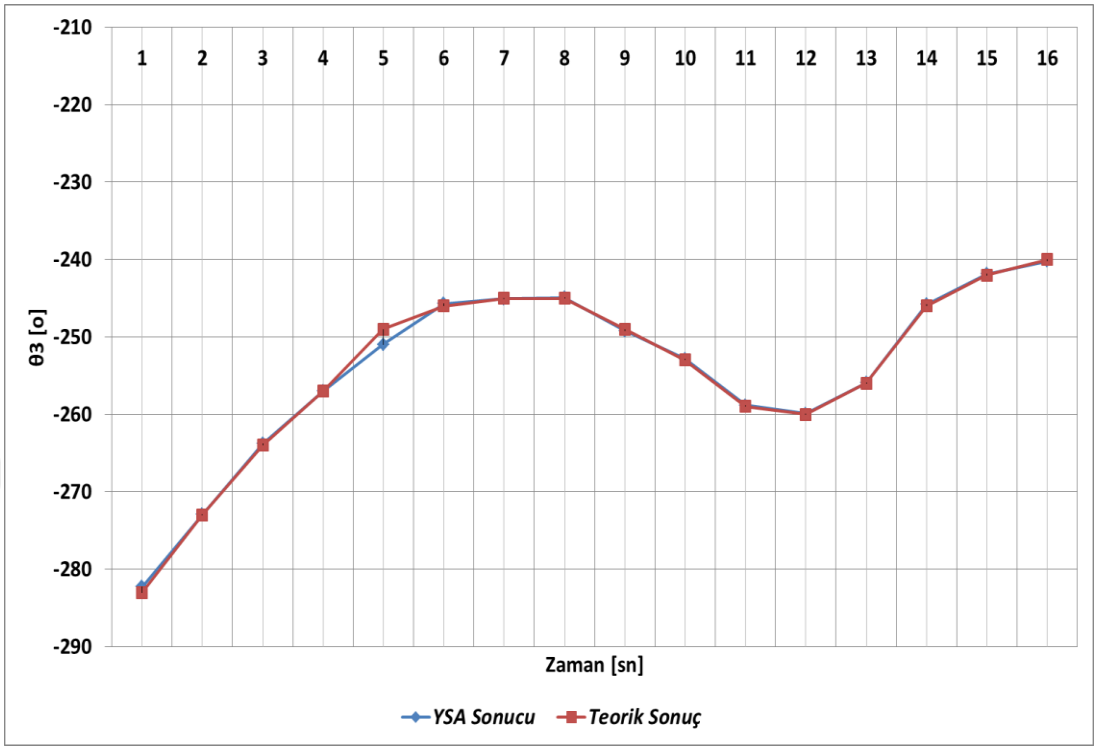

*Şekil 3.1.* 1.mafsal açısının **($\Theta_1$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Benzer şekilde delta-bar-delta öğrenme algoritması kullanarak, 2. mafsal açısının YSA esaslı tahminlemesi gerçekleştirilmiştir. Robot gövdesininin açısal değişiminden sonraki omuz açısı olarak kullanıldığından, Şekil 3.2.'da görüldüğü gibi ani açısal değişimlerin etkisinden dolayı 2.ve 4. saniyelerde, 6.ve 8. Saniyelerde benzer şekilde 14. Ve 16. Saniyelerde belirgin oranda hataların olduğu görülmektedir. Bu durum da göstermektedir ki, açısal olarak kimyasal püskürtülecek kumaş konumnda seri kimyasallamada ürün konum farklılığına sebeiyet verecektir. Bununu neticesinde, 2. Mafsal açısı için tahminleme yetersizliği söz konusudur.



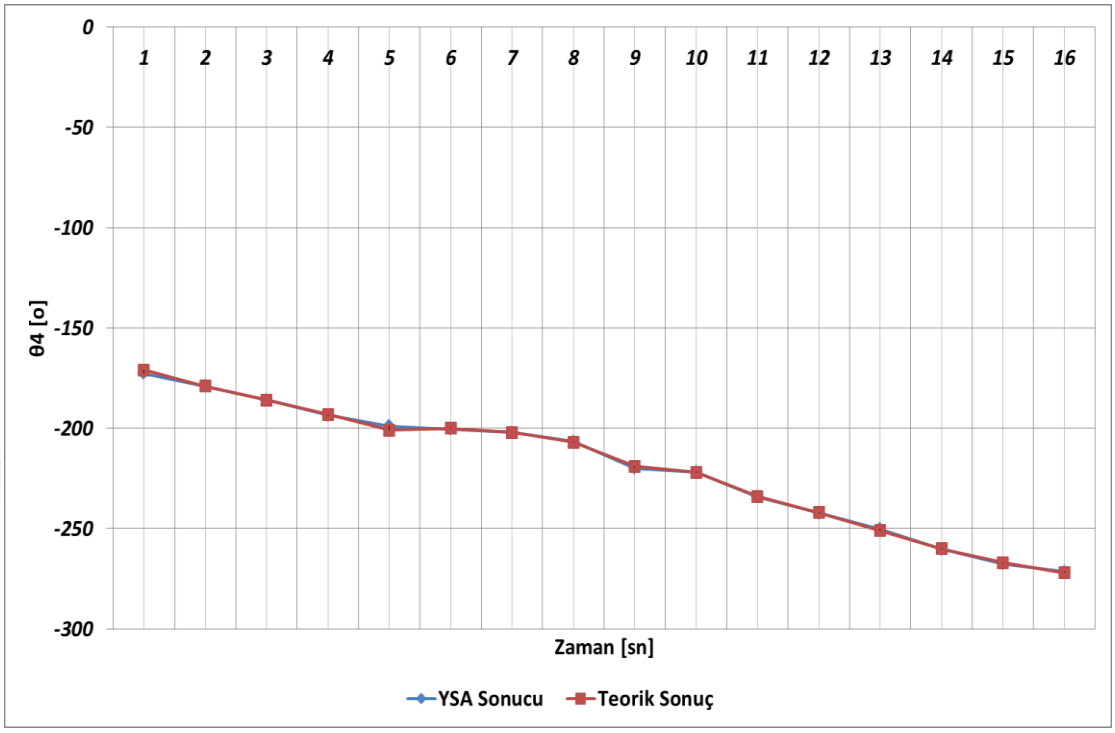

*Şekil 3.2.* 2. mafsal açısının **($\Theta_2$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot manipülatörün, insan yapısındaki dirsek görevini yerine getiren ve delta-bar-delta öğrenme algoritması kullanarak, 3. mafsal açısının YSA esaslı tahminlemesi gerçekleştirilmiştir. Robot gövdesininin açısal değişiminden sonraki omuz açısından sonraki dönem açısı yapısı olarak kullanıldığından, Şekil 3.3.'da görüldüğü gibi, belirlenen sonuçların yardımıyla, ani açısal değişimlerin etkisinden dolayı 1. saniyede, 4.ve 6. saniyelerde benzer şekilde 14. ve 16. saniyelerde belirgin oranda hataların olduğu görülmektedir. Bu durum da göstermektedir ki, 2.açısal değişimdeki hata etkileri ile birlite, açısal olarak kimyasal püskürtülecek kumaş konumnda seri kimyasallamada ürün konum farklılığına sebebiyet verecektir. Bununu neticesinde, 3. Mafsal açısı için tahminleme yetersizliği olarak vurgulanabilir. Diğer taraftan, ani açısal değişimlerde oluşan hataların yanında, bu zaman aralılları dışında çok iyi performans gösterdiği görülmektedir.



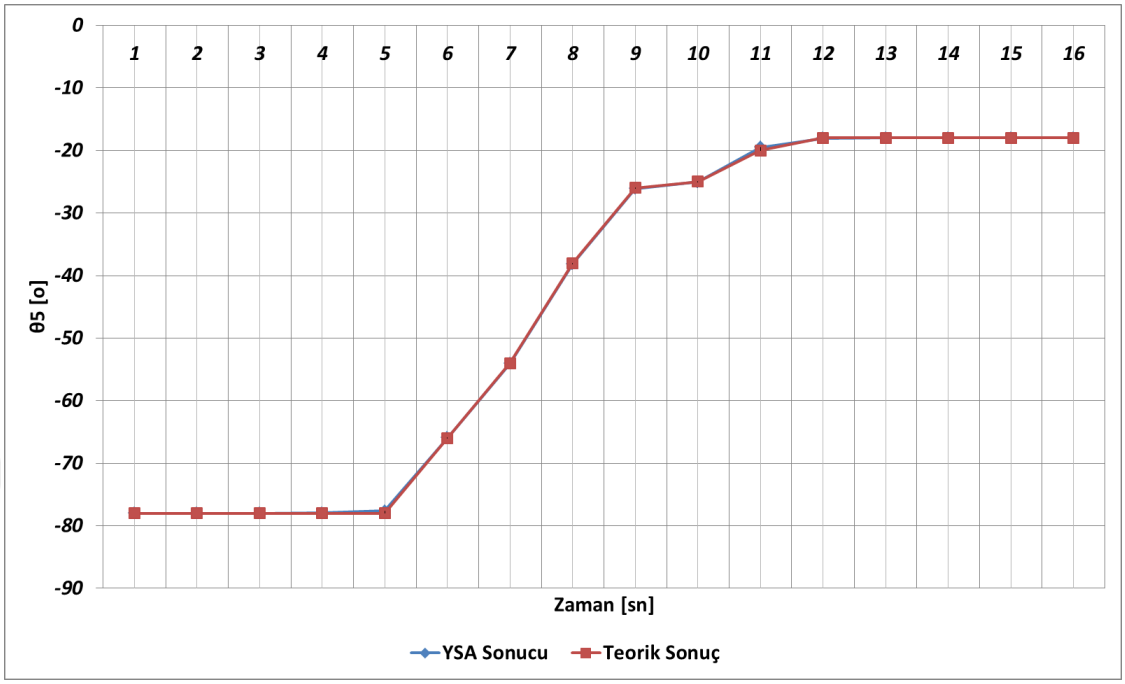

*Şekil 3.3.* 3.mafsal açısının **($\Theta_3$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Bu aşamadaki simülasyon çalışması yaklaşımında, robot manipülatörün 4. Açısal mafsalının, delta-bar-delta öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.4'de görüldüğü gibi, robot gripperin dönme açısının değişimi detayladırılmıştır. Ani açısal değişimlerin etkisinin olmamasına rağmen, 4. ve 6. saniyelerde, 8.ve 10. saniyelerde benzer şekilde 12. ve 15. saniyelerde belli oranlarda, teorik ve YSA yaklaşım arasında hataların olduğu görülmektedir. Ani dönme etkisinden dolayı, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada tercih olarak Kabul edilmemektedir.



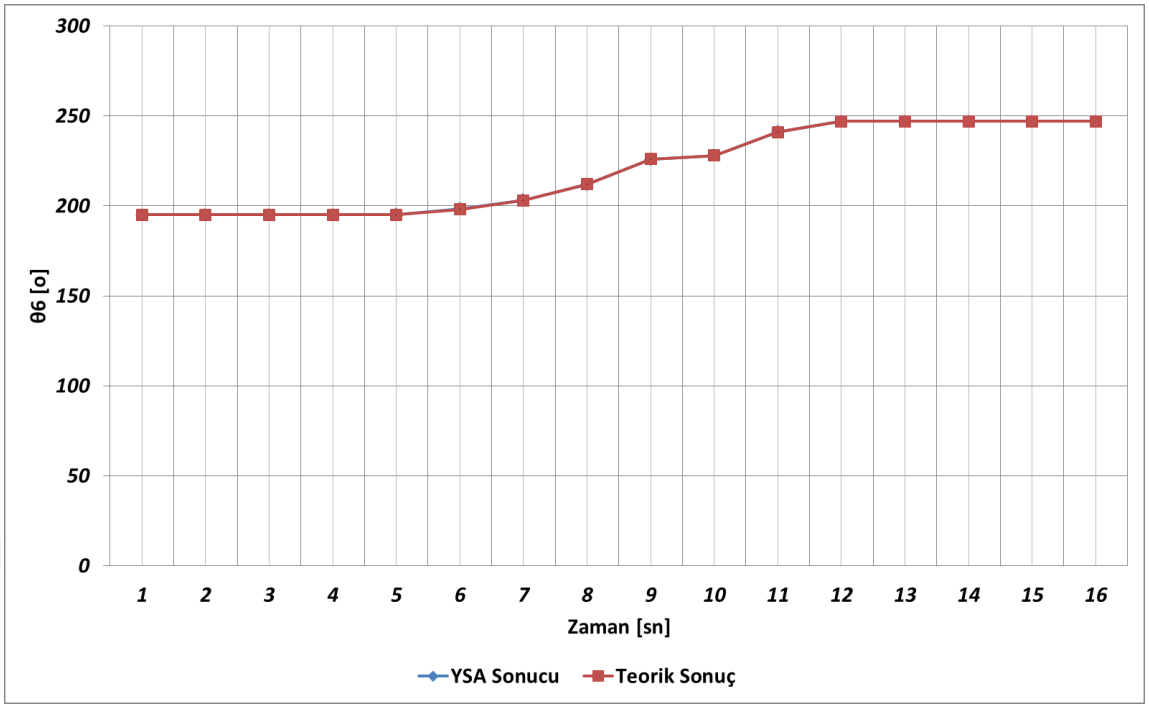

*Şekil 3.4.* 4.mafsal açısının **($\Theta_4$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Özellikle robot manipülatörün pozisyonalma önemli yapıya sahip, dönme açısı olan 5. mafsalda, küçük tahminleme hataları bile kabul edilebilir düzeyde değildir. Buradaki yaklaşımında, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 5. açısal mafsalının, yine aynı algoritma olan, delta-bar-delta öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.5'de görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak verilmiştir. Ani açısal değişimlerin etkisinden dolayı, 4. ve 5. saniyelerde, benzer şekilde 10.ve 12. saniyelerde belli oranlarda, teorik ve YSA yaklaşım arasında hataların olduğu görülmektedir. Ani dönme etkisinden dolayı, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada çok tercih edilen bir durum olarak dikkate alınmamaktadır.



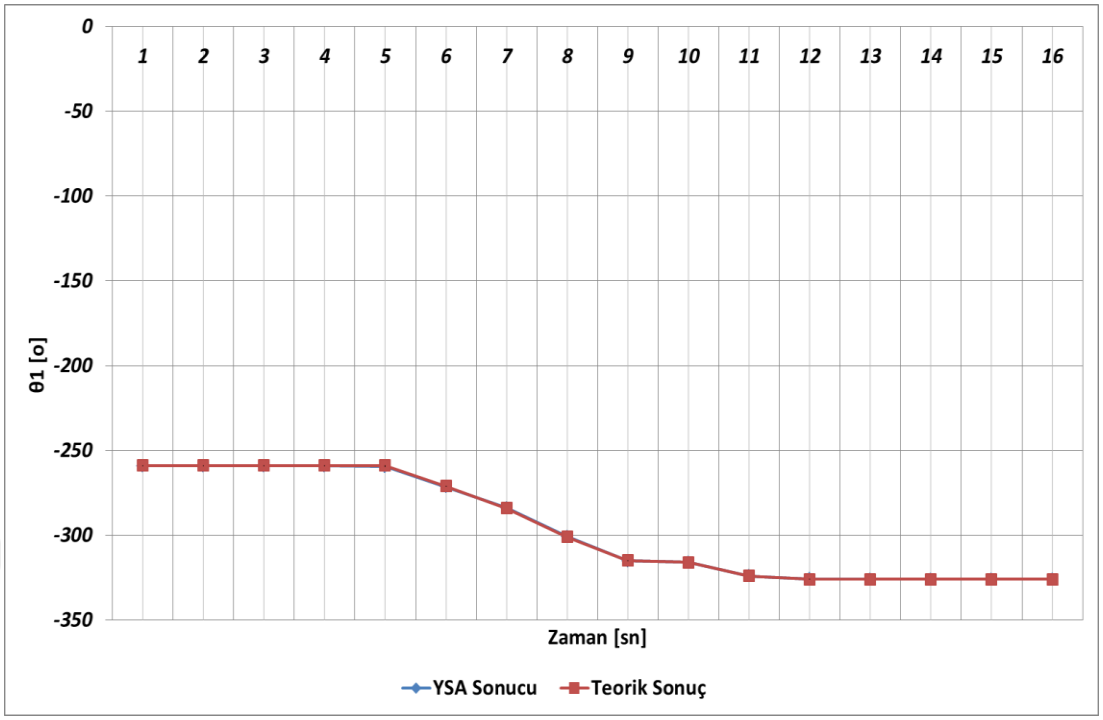

*Şekil 3.5.* 5.mafsal açısının **($\Theta_5$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot maipülatörün uç işlevcisinin en son dönme açısı özelliğinde olan, 6. Mafsal için yapılan yaklaşımda da pozisyonalma önemli yapıya sahip, dönme açısı olan 6. mafsalda, küçük tahminleme hataları yok denecek kadar azdır. Bunun nedeni ani açısal değişim olmamaması olarak vurgulanabilir. Benzer şekilde, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 6. açısal mafsalının, yine aynı algoritma olan, delta-bar-delta öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.6'da görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak, 16 saniyelik zaman aralığında detaylı olarak verilmiştir. Ani açısal değişimlerin etkisinin olmamasından dolayı, 1. ve 16. saniyelerde, teorik ve YSA yaklaşım arasında hataları görülmemektedir. Ani dönme etkisi çok sınırlı olsa bile, büyük açısal değişimler olmadığından, Kabul edeilebilir bir tahminmele gerçekleştirilmiştir. Netice olarak, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada çok tercih edilen bir durum olarak dikkate alınabilir.



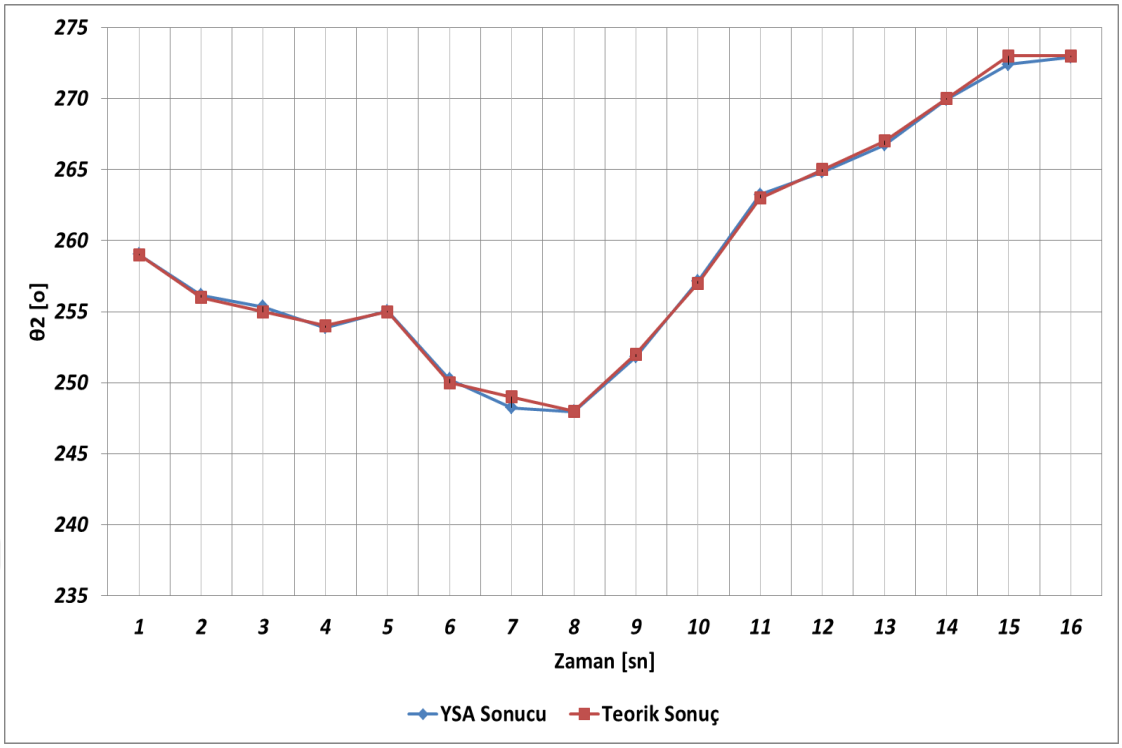

*Şekil 3.6.* 6.mafsal açısının **($\Theta_6$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Önceki grafik sonuçlarından da görüğleceği gibi, delta-bar-delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı simulasyon çalışması bazı açısal ($\Theta_1$ , $\Theta_4$ ,$\Theta_5$ , $\Theta_6$ ) değişimlerin tahminlmesinde iyi sonuçlar vermesine rağmen, özellikle 2. ve 3. ($\Theta_2$ ve $\Theta_3$ ) mafsal açılarında değişimde belirgin oranda hatalar oluşmuştur. Bu nedenle farklı bir öğrenme algoritması kullanılarak, YSA tahminleyici yapısında, her bir mafasal için simülasyon çalışması gerçekleştirilmiştir. Bu yaklaşımda kullanılan öğrenme algoritması on-line geri yayılım algoritması olup, 1.mafsal için tahminleme Şekil 3.11'de detaylandırılmıştır. Şekil 3.7'den de görüleceği gibi bu algoritma ile tahminleme, robot manipülatörün gövde dönme açısal değişimi olan $\Theta_1$ için iyi tahminleme gerçekleştirilmiştir. Burada, 5. ve 9. saniyelerde açısal değişim ani şekilde olmadığından yaklaşımda tahminelme sonucu hata çok az olduğundan, Kabul edilebilir düzeydedir.



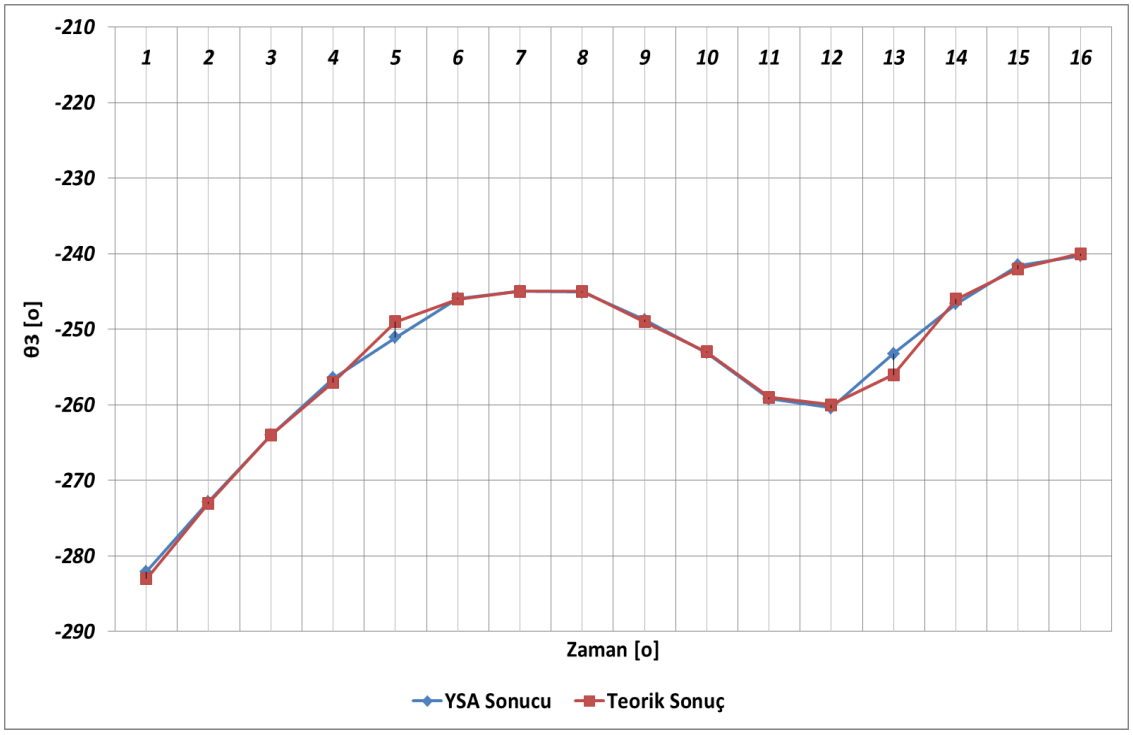

*Şekil 3.7.* 1.mafsal açısının **($\Theta_1$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Yine diğer yaklaşımlarda olduğu gibi, benzer şekilde On-Line Geri Yayımım öğrenme algoritması kullanarak, 2. mafsal açısının YSA esaslı tahminlemesi gerçekleştirilmiştir. Robot gövdesininin açısal değişiminden sonraki omuz açısı olarak uygulamlarda kullanıldığından, Şekil 3.8.'da görüldüğü gibi ani açısal değişimlerin etkisinden dolayı 2.ve 4. saniyelerde, 6.ve 8. saniyelerde benzer şekilde 11. ve 16. saniyelerde belirgin oranda hataların olduğu görülmektedir. Bu durum da göstermektedir ki, açısal olarak kimyasal püskürtülecek kumaş konumnda seri kimyasallamada ürün konum farklılığına sebebiyet verecektir. Özellikle 6. ve 8. Saniyeler aralığında açısal değişim hata oranı çok yüksektir. Bununu neticesinde, 2. Mafsal açısı için tahminleme yetersizliği söz konusudur.



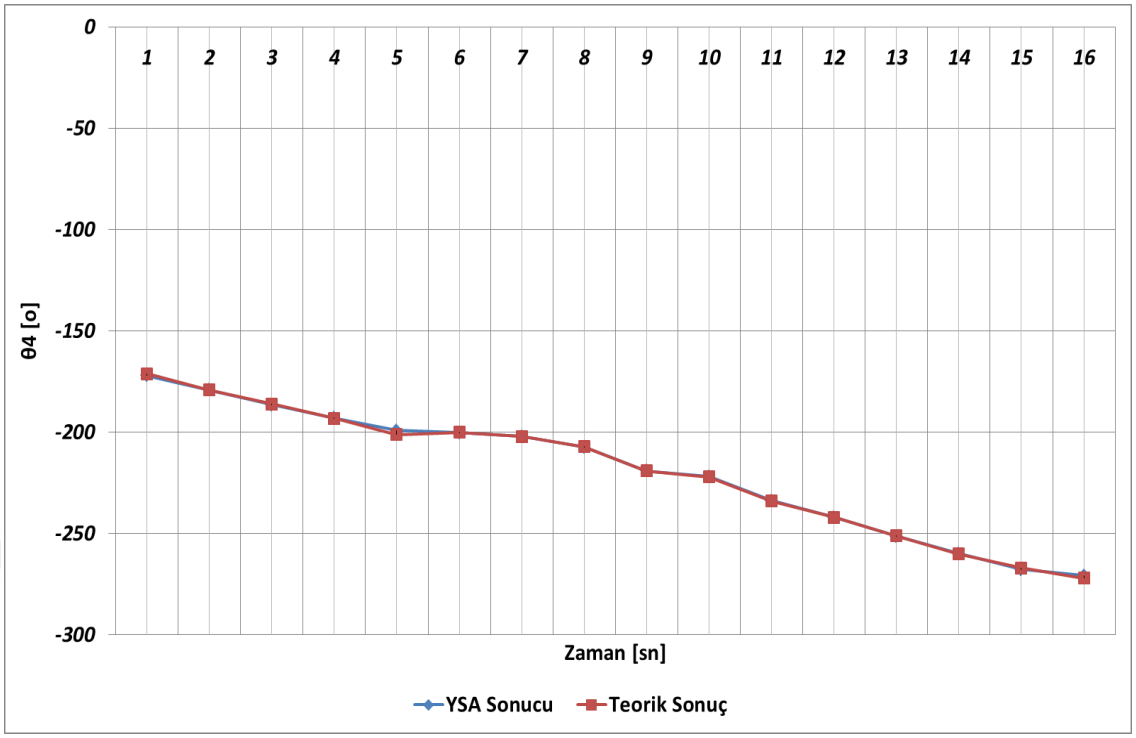

*Şekil 3.8.* 2.mafsal açısının **($\Theta_2$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Endüstriyel robot manipülatörün, insan yapısındaki dirsek olarak adlandırılan ve On-Line Geri Yayılım öğrenme algoritması kullanarak, 3. mafsal açısının YSA esaslı tahminlemesi gerçekleştirilmiştir. Robot gövdesininin açısal değişiminden sonraki omuz açısından sonraki dönem açısı yapısı olarak kullanıldığından, Şekil 3.9.'da görüldüğü gibi, belirlenen sonuçların yardımıyla, ani açısal değişimlerin etkisinden dolayı 1. Ve 2. saniyede, 4.ve 6. saniyelerde benzer şekilde 11. ve 16. saniyelerde belirgin oranda hatalar söz konusudur. Bu durum da göstermektedir ki, 2. Mafsalın açısal değişimdeki hata etkileri ile birlite, açısal olarak kimyasal püskürtülecek kumaş konumnda seri kimyasallamada ürün konum ve renk farklılığına sebebiyet verecektir. Bunun neticesinde, 3. Mafsal açısı için tahminlemenin çok etkili olmadığı ve yetersizliği olarak vurgulanabilir. Diğer taraftan, ani açısal değişimlerde oluşan hataların yanında, bu zaman aralıkları dışında çok iyi performans gösterdiği görülmektedir.



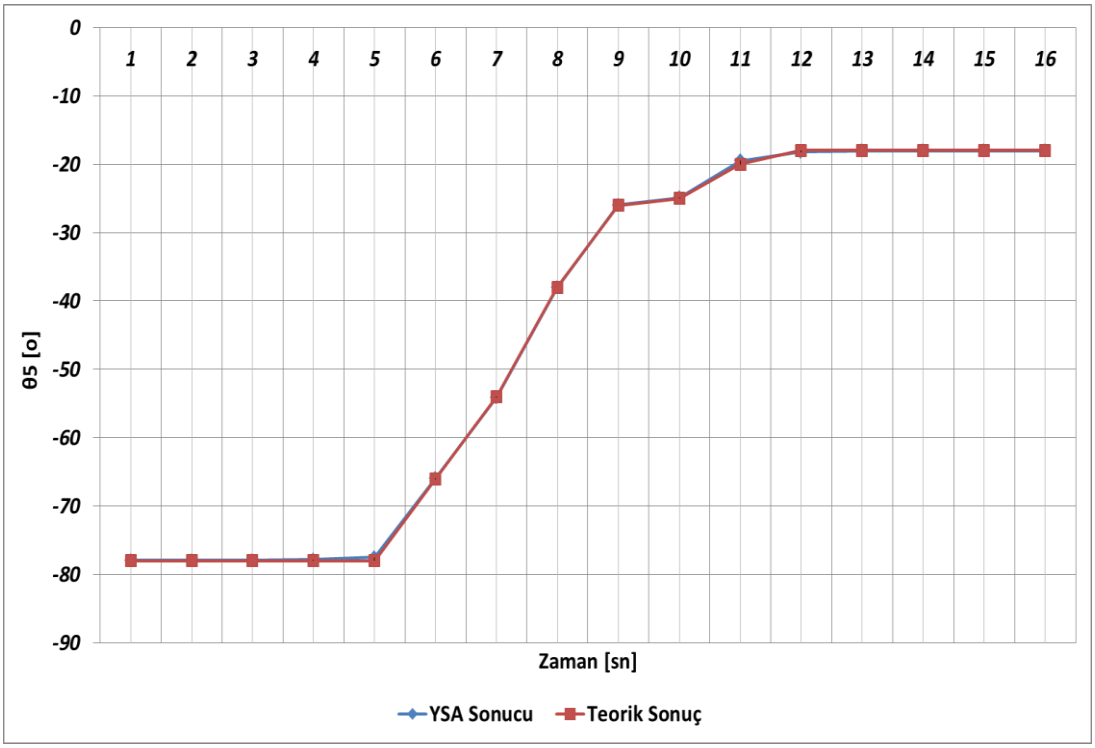

*Şekil 3.9.* 3.mafsal açısının **($\Theta_3$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Bu aşamadaki simülasyon çalışması yaklaşımında, robot manipülatörün 4. Açısal mafsalının, On-Line Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.10'de görüldüğü gibi, robot gripperin dönme açısının değişimi detayladırılmıştır. Ani açısal değişimlerin etkisinin olmamasına rağmen, 4. ve 6. saniyelerde, 14. ve 16. saniyelerde belli oranlarda, teorik ve YSA yaklaşım arasında hataların olduğu görülmektedir. Özellikle açısal değişimin 5. Saniyesinde maksimum açısal hatalar oluşmakta olduğu görülmektedir. Ani dönme etkisinden dolayı, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada tercih olarak kabul edilmemektedir.



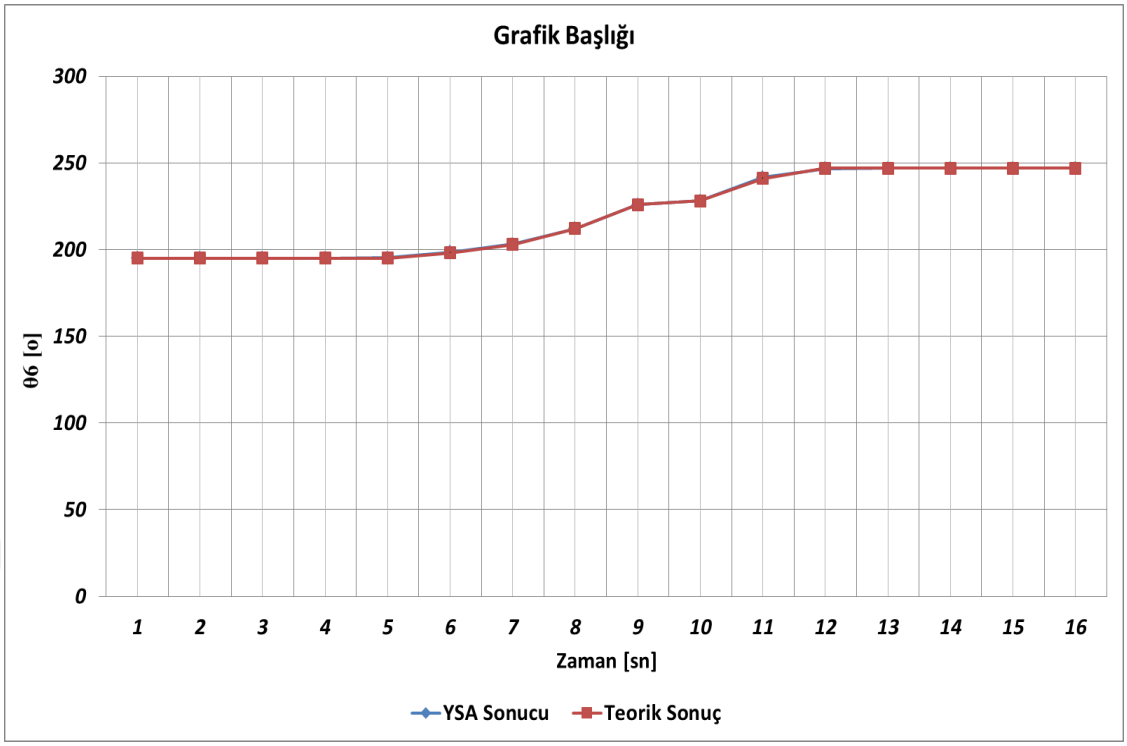

*Şekil 3.10.* 4.mafsal açısının **($\Theta_4$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Kimyasal püskürtme etkisinde kullanılan uç işlevcinin poziyonlam için dönme açısı çok önem arz etmeketdir. Bundan dolayı, robot manipülatörün pozisyonalma önemli yapıya sahip, dönme açısı olan 5. mafsalda, küçük tahminleme hataları bile kabul edilebilir düzeyde değildir. Buradaki yaklaşımında, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 5. açısal mafsalının, yine aynı algoritma olan, On-Line Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.11'de görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak verilmiştir. Ani açısal değişimlerin etkisinden dolayı, 3. ve 6. saniyelerde, benzer şekilde 10.ve 13. Saniyelerde ki açısal konum değişim aralıklarında, belli oranlarda, teorik ve YSA yaklaşım arasında hataların olduğu görülmektedir. Burada da, ani açısal dönme durumundan etkisinde, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada çok tercih edilen bir durum olarak dikkate alınmamaktadır.




*Şekil 3.11.* 5.mafsal açısının **($\Theta_5$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

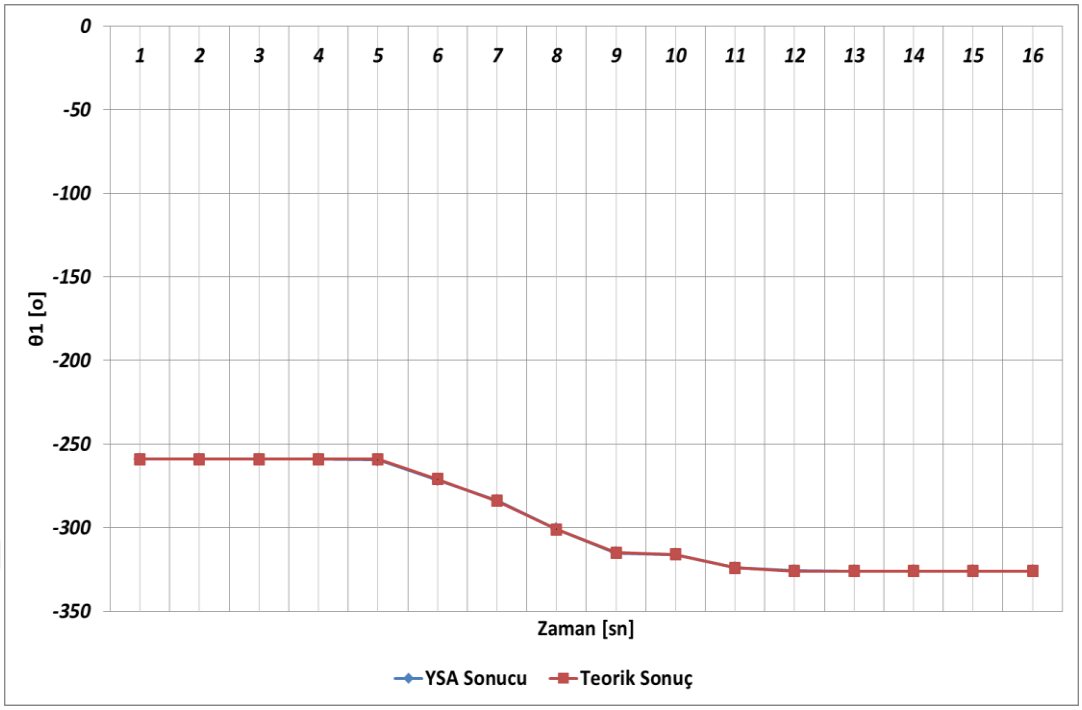
Robot kolun, uç işlevcisinin en son dönme açısı olan, 6. Mafsal için yapılan yaklaşımda da pozisyonlama önemli yapıya sahip, dönme açısı olan 6. mafsalda, küçük tahminleme hataları yok denecek kadar azdır. Bunun nedeni aniden konumlama olmadığından, açısal değişim olmamaması olarak vurgulanabilir. Benzer şekilde, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 6. açısal mafsalının, yine aynı algoritma olan, delta-bar-delta öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.12'da görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak, 16 saniyelik zaman aralığında detaylı olarak verilmiştir. Ani açısal değişimlerin etkisinin olmamasından dolayı, 1. ve 16. saniyelerde, teorik ve YSA yaklaşım arasında hataları görülmemektedir. Ani dönme etkisi çok sınırlı olsa bile, büyük açısal değişimler olmadığından, kabul edilebilir bir tahminmele gerçekleştirilmiştir. Netice olarak, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada çok tercih edilen bir durum olarak dikkate alınabilir.




*Şekil 3.12.* 6.mafsal açısının **($\Theta_6$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

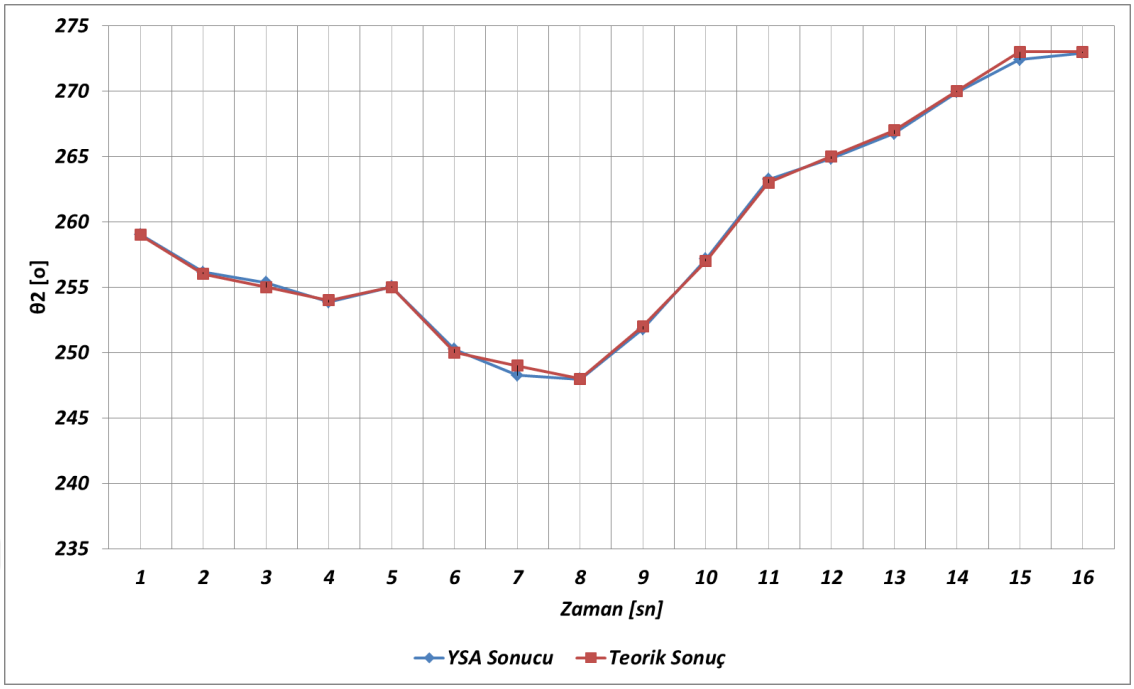
Önceki yaklaşımlarda her bir mafsal açısı için grafiksel sonuçlar yardımıyla, detaylandırıldığı gibi her bir mafsal açısal değişimi tahminlemesi için belli oranlarda hatalardan dolayı, bu yaklaşımların yetersizliği sonucuna varılmaktadır. Özellikle, Şekiller 3.9 ve 3.8 de görüleceği gibi, belirgin oranda yaklaşım hataları gözlenmiştir. Bu açısal (**$\Theta_2$** ve **$\Theta_3$)** hataların oluşmasına en büyük etken, aniden açısal konum değişme etkisi olarak verilebilir.

Burada farklı bir öğrenme algoritması kullanarak, optimum tahminleme üzerine sonuçlar elede edilmeye çalışılmıştır. YSA yapısında öğrenme etkisinde kullanılan algoritma Hızlı Yayılım algoritması olarak seçilmiştir. Endüstriyel robot manipülatörün gövdesinin dönme açısal değişimi olarak kullanılan 1.mafsal açısal değişimin tahminlemesini gösteren sonuç Şekil 3.13'da verilmiştir.



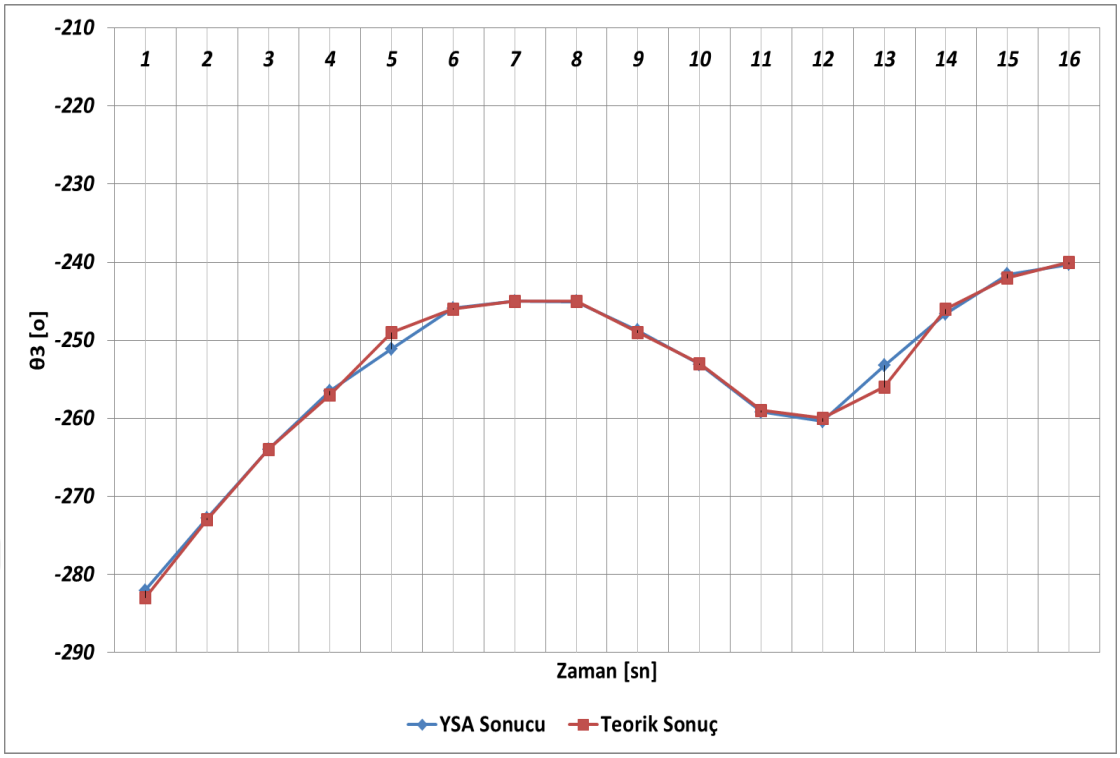

*Şekil 3.13*. 1.mafsal açısının **($\Theta_1$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Şekil 3.16'dan da görüleceği gibi, Hızlı Geri Yayılım öğrenme algoritması ile en iyi YSA yaklaşım sonuçları elde edilmiştir. 5. ve 6. Saniye alağında açısal değişimde çok küçük hatalar gözlemlemiştir. Robot manipülatörün omuz açısını ifade den açısının **($\Theta_2$)** Hızlı Geri Yayılım algoritması kullanarak YSA ile tahminleme yaklaşımı değişimi Şekil 3.17'de görüldüğü gibidir. Benzer şekilde, Şekil 3.14' den de görüleceği gibi, ani açısal değişimlerin etkisinden dolayı 2.ve 4. saniyelerde, 6.ve 8. saniyelerde benzer şekilde 11. ve 16. saniyelerde belirgin oranda hataların olduğu görülmektedir. Bu durum da göstermektedir ki, açısal olarak kimyasal püskürtülecek kumaş konumnda seri kimyasallamada ürün konum farklılığına sebebiyet verecektir. Özellikle 6. ve 8. Saniyeler aralığında açısal değişim hata oranı çok yüksektir. Bununu neticesinde, 2. Mafsal açısı için tahminleme yetersizliği kimyalllama hataları oluşturabilecektir.



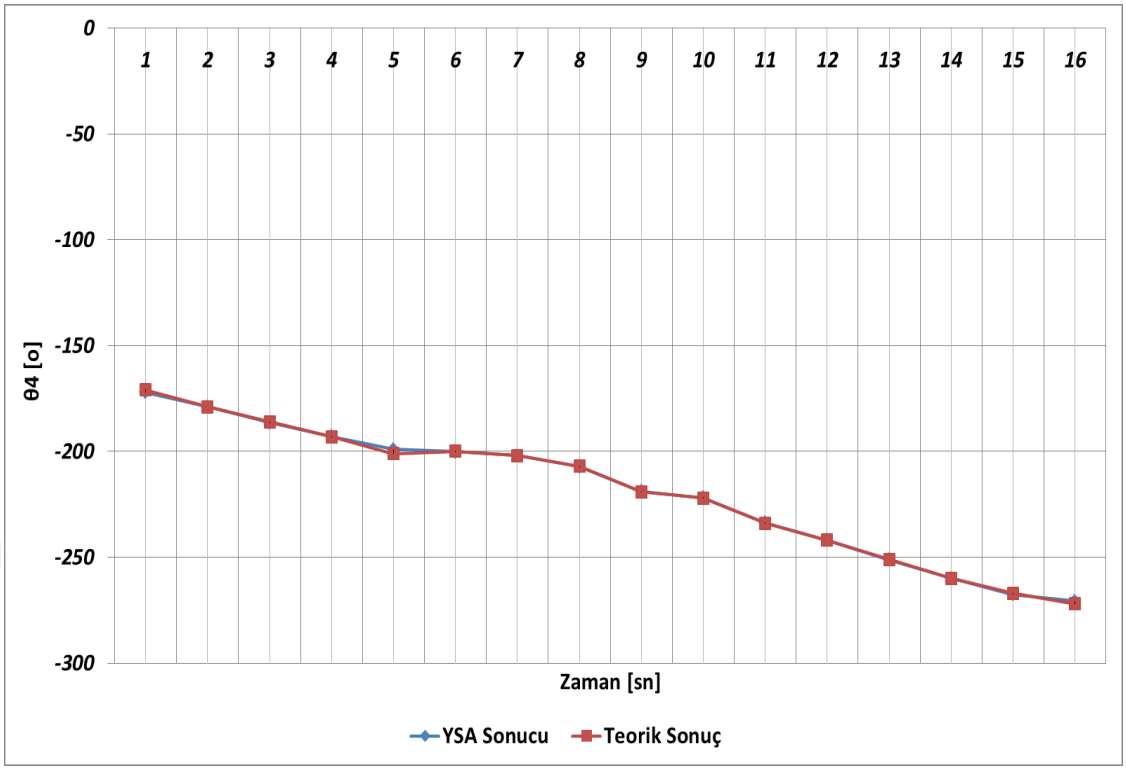

*Şekil 3.14.* 2. mafsal açısının **($\Theta_2$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot manipülatörün, insan yapısında ki dirsek olarak adlandırılan ve Hızlı Geri Yayılım öğrenme algoritması kullanarak, 3. mafsal açısının YSA esaslı tahminlemesi gerçekleştirilmiştir. Robot gövdesininin açısal değişiminden sonraki omuz açısından sonraki dönem açısı yapısı olarak kullanıldığından, Şekil 3.15.'da görüldüğü gibi, belirlenen sonuçların yardımıyla, ani açısal değişimlerin etkisinden dolayı 1. ve 2. saniyede, 4.ve 6. saniyelerde benzer şekilde 11. ve 14. saniyelerde belirgin oranda hatalar söz konusudur. Bu durum da göstermektedir ki, benzer şekilde 2. mafsalın açısal değişimdeki hata etkileri ile birlite, açısal olarak kimyasal püskürtülecek kumaş konumunda seri kimyasallamada ürün konum ve renk farklılığına sebebiyet verecektir. Bunun neticesinde, 3. Mafsal açısı için tahminlemenin çok etkili olmadığı ve yetersizliği olarak vurgulanabilir. Diğer taraftan, ani açısal değişimlerde oluşan hataların yanında, bu zaman aralıkları dışında çok iyi performans gösterdiği görülmektedir.



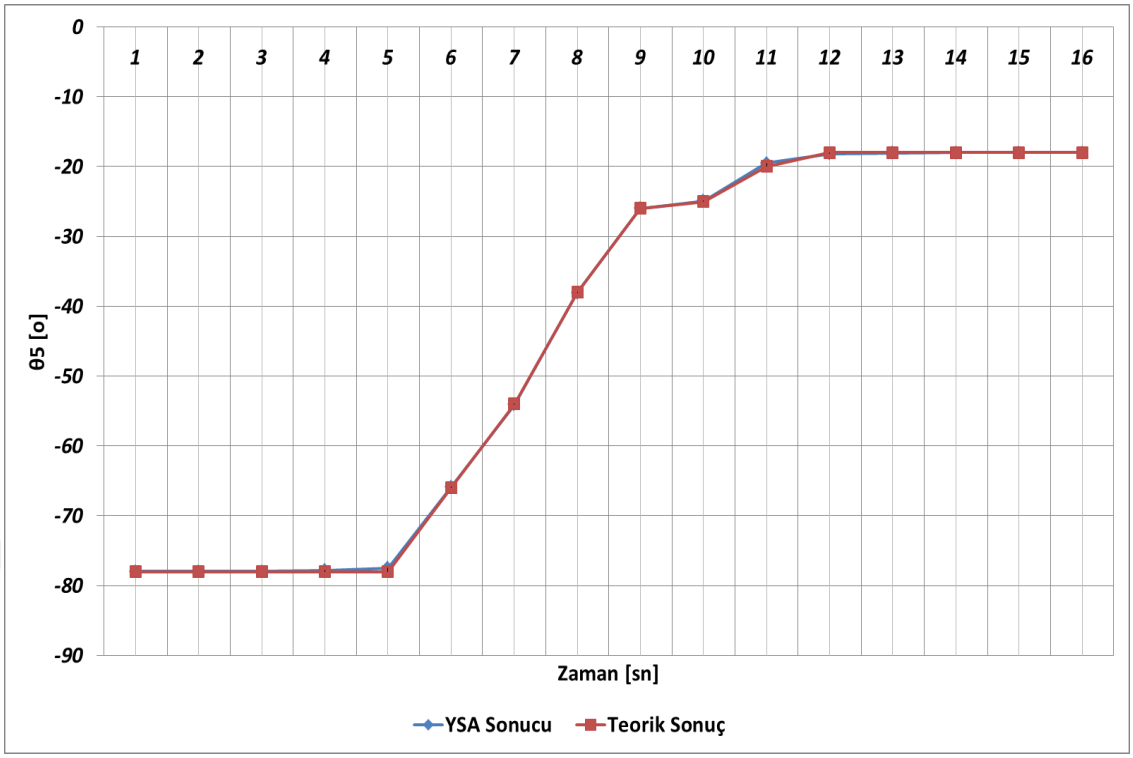

*Şekil 3.15.* 3. mafsal açısının **($\Theta_3$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

4. Açısal mafsalının, Hızlı Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.16'da detaylı grafiksel olarak verildiği gibi, robot uç işlevcisi dönme açısının değişimi detayladırılmıştır. Ani açısal değişimlerin etkisinin olmamasına rağmen, 4. ve 6. saniyelerde, 14. ve 16. saniyelerde çok az da olsa, teorik ve YSA yaklaşım arasında hataların olduğu görülmektedir. Özellikle açısal değişimin 5. Saniyesinde maksimum açısal hatalar oluşmakta olduğu görülmektedir. Ani dönme etkisinden dolayı, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarda tercih olarak kabul edilmemektedir.



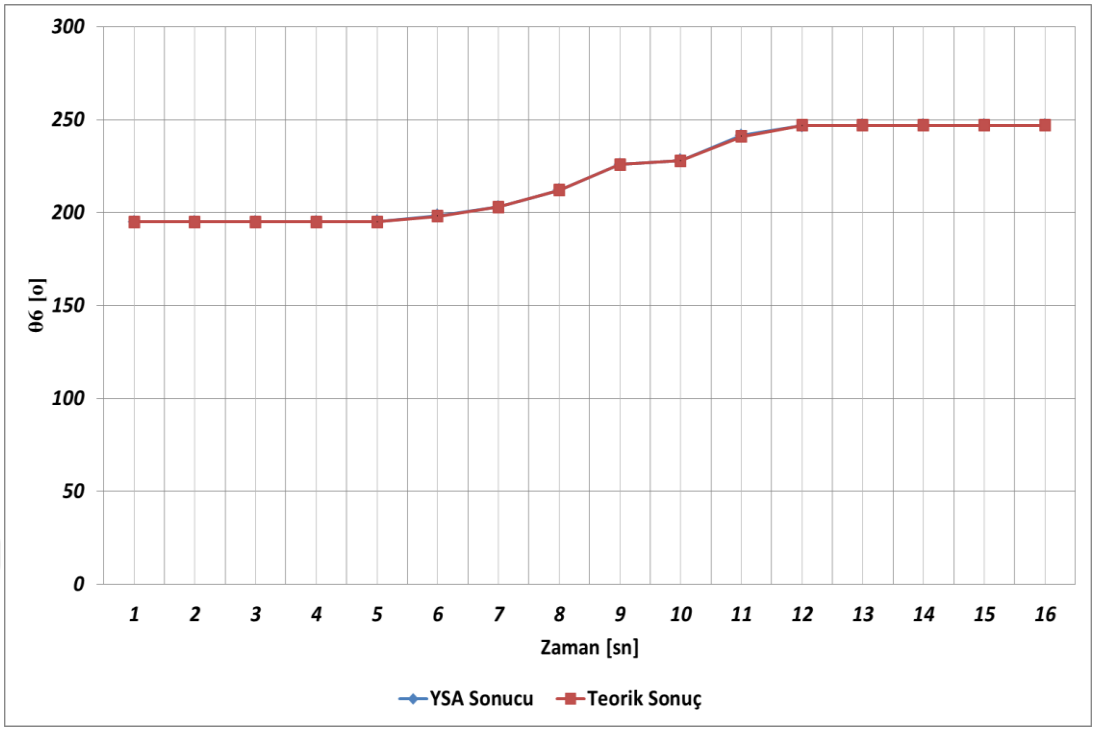

*Şekil 3.16.* 4. mafsal açısının **($\Theta_4$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot manipülatörün pozisyonalma önemli yapıya sahip, dönme açısı olan 5. mafsalda, küçük tahminleme hataları bile kabul edilebilir düzeyde değildir. Buradaki yaklaşımında, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 5. açısal mafsalının, yine aynı algoritma olan, Hızlı Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.17'de görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak verilmiştir. Ani açısal değişimlerin etkisinden dolayı, 3. ve 6. saniyelerde, benzer şekilde 10. ve 13. saniyelerde ki açısal konum değişim aralıklarında, belli oranlarda, teorik ve YSA yaklaşım arasında hataların olduğu görülmektedir. Burada da, diğer öğrenme algoritmalarında gözlemlendiği gibi, ani açısal dönme durumundan etkisinde, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarada çok tercih edilen bir durum olarak dikkate alınmamaktadır.




*Şekil 3.17.* 5. mafsal açısının **($\Theta_5$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

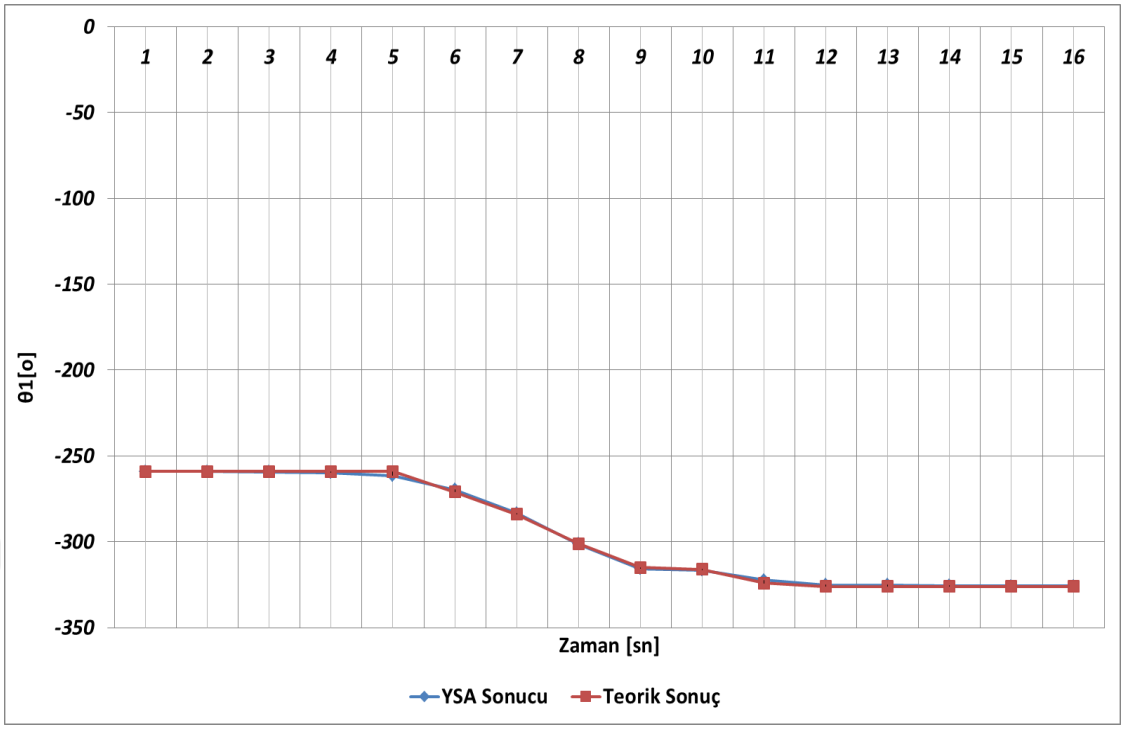
Robot kolun, uç işlevcisinin en son dönme açısı olan, 6. Mafsal için yapılan yaklaşımda da pozisyonlama önemli yapıya sahip, dönme açısı olan 6. mafsalda, küçük tahminleme hataları yok denecek kadar azdır. Benzer şekilde, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 6. açısal mafsalının, yine aynı algoritma olan, Hızlı Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.18'da görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak, 16 saniyelik zaman aralığında detaylı olarak verilmiştir. Ani açısal değişimlerin etkisinin olmamasından dolayı, 1. ve 16. saniyelerdeki haraket aralığında, teorik ve YSA yaklaşım arasında hataları görülmemektedir. Ani dönme etkisi çok sınırlı olsa bile, büyük açısal değişimler olmadığından, kabul edilebilir bir tahminmele gerçekleştirilmiştir.




*Şekil 3.18.* 6. mafsal açısının **($\Theta_6$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

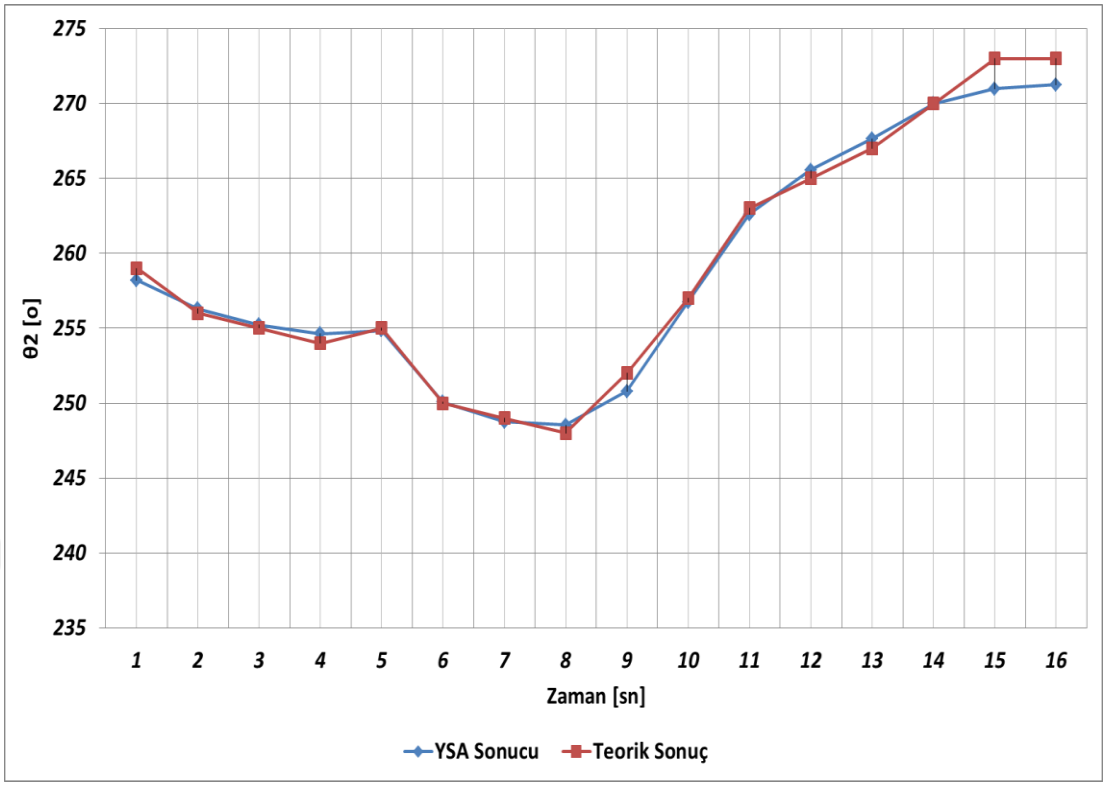
Önceki gerçekleştirlen, farklı öğrenme algoritmalral aynı tip YSA, tahminleme yaklaşımlarda her bir mafsal açısı için grafiksel sonuçlar yardımıyla, detaylandırıldığı gibi her bir mafsal açısal değişimi tahminlemesi için belli oranlarda hatalardan dolayı, bu yaklaşımların bazı mafsal açıları için, yetersizliği sonucuna varılmaktadır. Özellikle, Şekiller 3.14 ve 3.15 de görüleceği gibi, belirgin oranda yaklaşım hataları gözlenmiştir. Bu açısal (**$\Theta_2$** ve **$\Theta_3$)** hataların oluşmasına en büyük etken, aniden açısal konum değişme etkisi olarak verilebilir.

Burada farklı bir öğrenme algoritması kullanarak, optimum tahminleme üzerine sonuçlar elede edilmeye çalışılmıştır. YSA yapısında öğrenme etkisinde kullanılan algoritma Random (Gelişigüzel) Geri Yayılım algoritması olarak seçilmiştir. Endüstriyel robot manipülatörün gövdesinin dönme açısal değişimi olarak kullanılan 1.mafsal açısal değişimin tahminlemesini gösteren sonuç Şekil 3.19' da verilmiştir.



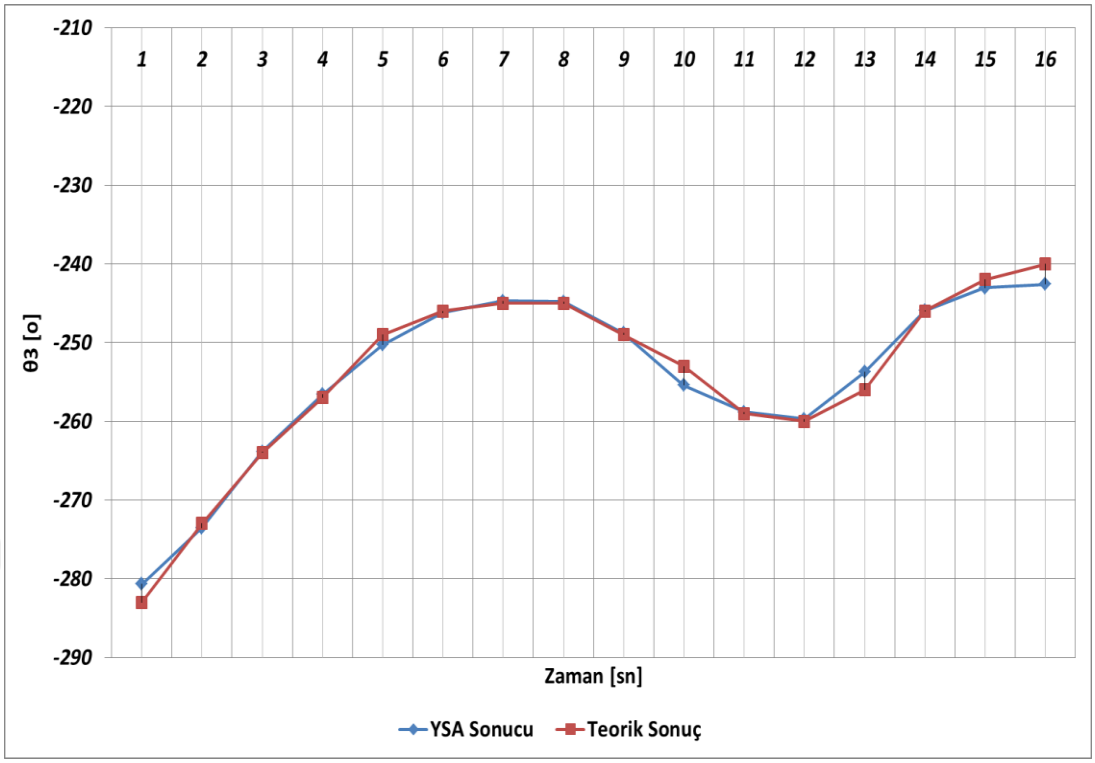

*Şekil 3.19.* 1. mafsal açısının **($\Theta_1$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Şekil 3.22'den da görüleceği gibi, Random (gelişigüzel) ı Geri Yayılım öğrenme algoritması ile en iyi YSA yaklaşım sonuçları elde edilmiştir. 4. ve 7. saniyeler alağında açısal değişimde çok belirgin hatalar gözlemlemiştir. Bunun yanında robot manipülatörün omuz açısını ifade den açısının **($\Theta_2$)** Hızlı Geri Yayılım algoritması kullanarak YSA ile tahminleme yaklaşımı değişimi Şekil 3.20'de görüldüğü gibidir. Benzer şekilde, Şekil 3.20' den de görüleceği gibi, ani açısal değişimlerin etkisinden dolayı 1.ve 5. saniyelerde, benzer şekilde 6.ve 16. saniyelerde belirgin oranda, açısal değişim hatalarının olduğu görülmektedir. Yine burada da vurgulanabilir, bu durum da göstermektedir ki, açısal olarak kimyasal püskürtülecek kumaş konumnda seri kimyasallamada ürün konum farklılığına sebebiyet verecektir. Özellikle 14. ve 16. saniyeler aralığında açısal değişim yaklaşım hatalarının oranı çok yüksektir. Bunun neticesinde, 2. Mafsal açısı için tahminleme yetersizliği kimyalllama hataları oluşturabilecek olacağı kaçınılmazdır.



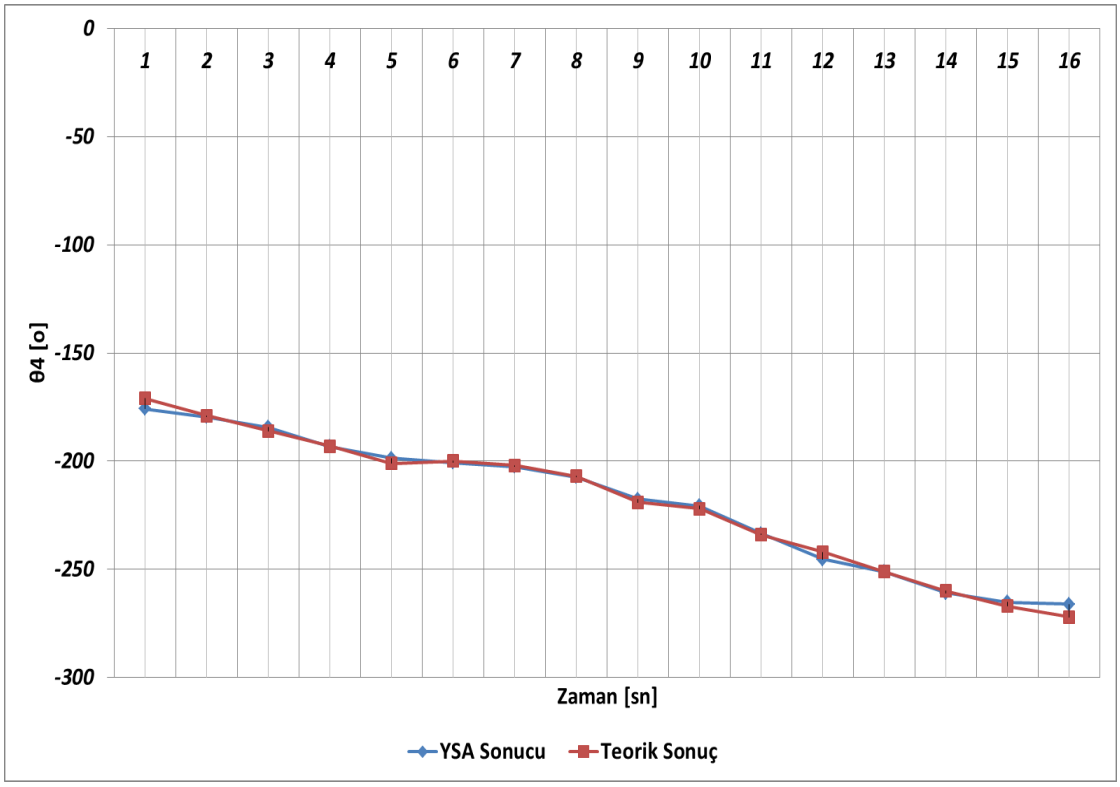

*Şekil 3.20.* 2. mafsal açısının **($\Theta_2$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot manipülatörün, insan yapısındaki dirsek olarak adlandırılan ve Random (Gelişigüzel) Geri Yayılım öğrenme algoritması kullanarak, 3. mafsal açısının YSA esaslı tahminlemesi gerçekleştirilmiştir. Robot gövdesininin açısal değişiminden sonraki omuz açısından sonraki dönem açısı yapısı olarak kullanıldığından, Şekil 3.21'de görüldüğü gibi, belirlenen sonuçların yardımıyla, ani açısal değişimlerin etkisinden dolayı 1. ve 2. saniyede, 4.ve 6. saniyelerde benzer şekilde 12. ve 16. saniyelerde belirgin oranda hatalar söz konusudur. Bu durum da göstermektedir ki, benzer şekilde 2. mafsalın açısal değişimdeki hata etkileri ile birlite, açısal olarak kimyasal püskürtülecek kumaş konumunda seri kimyasallamada ürün konum ve renk farklılığına sebebiyet verecektir. Özellikle 9. Saniyedeki açısal konum değişimde istenen referans açısal değişimi izleme durumu oluşmamaktadır. Bunun neticesinde, 3. Mafsal açısı için tahminlemenin çok etkili olmadığı ve yetersizliği olarak vurgulanabilir.



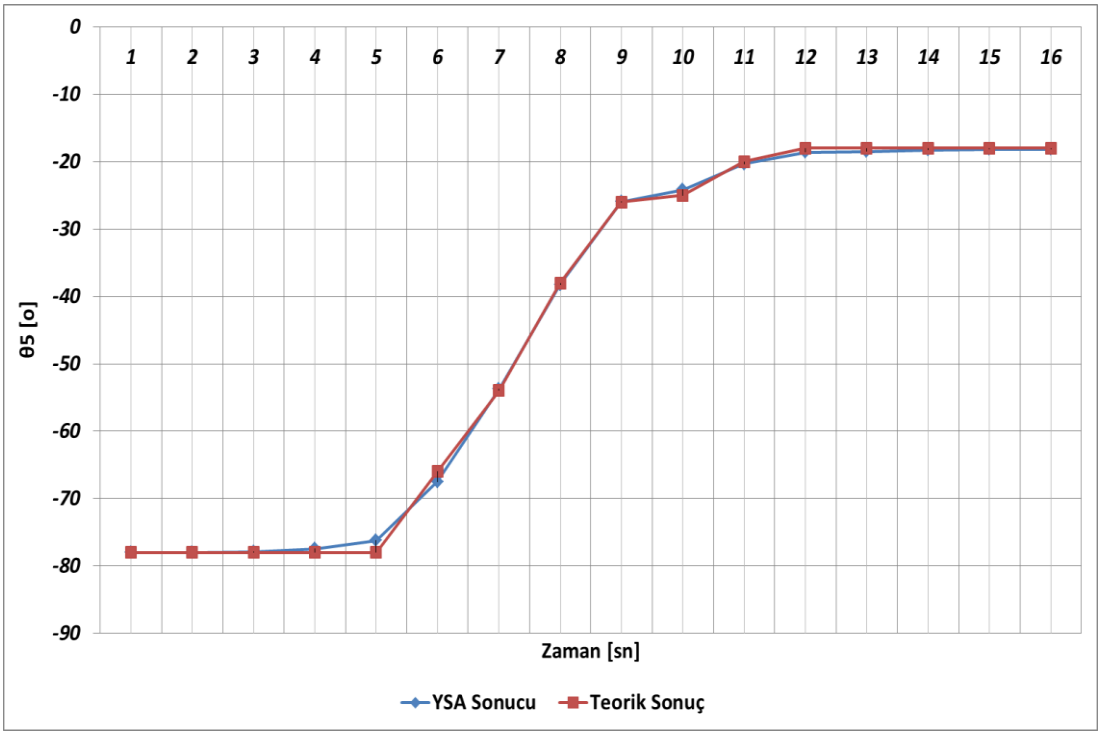

*Şekil 3.21.* 3. mafsal açısının **($\Theta_3$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

4. Açısal mafsalının, Random (Gelişigüzel) Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.22'de detaylı grafiksel olarak verildiği gibi, robot uç işlevcisi dönme açısının değişimi detayladırılmıştır. Ani açısal değişimlerin etkisinin olmamasına rağmen, 1. ve 5. saniyelerde, 9. ve 16. saniyelerde belirgin olarak hatalatrın olduğu görülmeketedir. Referenas ve YSA açısal tahminleme yaklaşımı arasında kabul edilmeyen hataların olduğu görülmektedir. Özellikle açısal değişimin her saniyesinde azalan ve artan açısal hatalar oluşmakta olduğu görülmektedir. Ani dönme etkisi olmamasına rağmenı, uzun zaman aralığında hatalar deneysel uygulamalı çalışmalarda tercih sebebi olarak kabul edilmesi mümkün değildir.



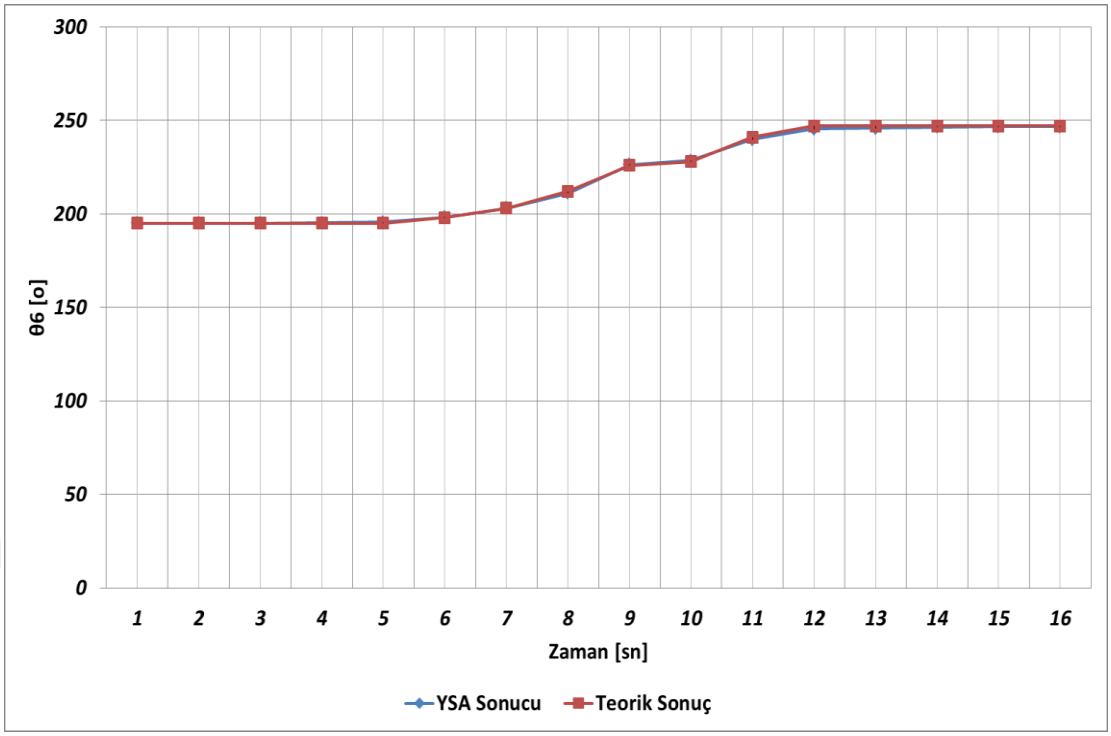

*Şekil 3.22.* 4. mafsal açısının **($\Theta_4$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot manipülatörün pozisyonalma önemli yapıya sahip, dönme açısı olan 5. mafsalda, küçük tahminleme hataları bile kabul edilebilir düzeyde değildir. Buradaki yaklaşımında, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 5. açısal mafsalının, yine aynı algoritma olan, Random (Gelişigüzel) Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.23'de görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak verilmiştir. Ani açısal değişimlerin etkisinden dolayı, 3. ve 7. saniyelerde, benzer şekilde 9.ve 16. saniyelerdeki zaman aralığında açısal konum değişim aralıklarında, Kabul edilmeyen oranlarda, referenas ve YSA yaklaşım arasında hataların olduğu görülmektedir. Burada da, diğer öğrenme algoritmalarında belirgin olarak gözlemlenmediği halde, ani açısal dönme durumundan etkisinde, uzun zaman aralığında çok belirgin hatalar deneysel uygulamalı çalışmalarada çok tercih edilen bir durum olarak dikkate alınmamaktadır.




*Şekil 3.23.* 5. mafsal açısının **($\Theta_5$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Robot kolun, uç işlevcisinin en son dönme açısı olan, 6. Mafsal için yapılan yaklaşımda da pozisyonlama önemli yapıya sahip, dönme açısı olan 6. mafsalda, küçük tahminleme hataları yok denecek kadar azdır. Benzer şekilde, robot manipülatörün uç işlevcisinin dönel mafsallı olan, 6. açısal mafsalının, yine aynı algoritma olan, Random (Gelişigüzel) Geri Yayılım öğrenme algoritması kullanılarak YSA esaslı yaklaşımında, Şekil 3.24'de görüldüğü gibi, robot uç işlevcinin dönme açısının değişimi grafiksel olarak, 16 saniyelik zaman aralığında detaylı olarak verilmiştir. Ani açısal değişimlerin etkisinin olmamasından dolayı, 11. ve 14. saniyelerdeki haraket aralığında, teorik ve YSA yaklaşım arasında hataları görülmektedir. Ani dönme etkisi çok sınırlı olsa bile, büyük açısal değişimler olmadığından, kabul edilebilir düzeyde bir tahminmele gerçekleştirilmemiştir.




*Şekil 3.24.* 6. mafsal açısının **($\Theta_6$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi

Önceki grafiksel sonuçlarda da detaylandırıldığı gibi, farklı dört adet öğrenme algoritması kullanılarak YSA yapısının ağırlıkları değiştirilerek referens (istenen) açısal değişimin tahminlemesi gerçekleştilmiştir. Bunun yanında, daha hızlı öğrenme etkisi sağlayacak YSA yapısında gizli katmanda 5 adet lineer olmayan hüzcre kullanılarak tahminleme işlemei gerçekleştirilmiştir. Tablo 3.1'de de görüleceği gibi bu YSA yapısı, gizli katmada 10 adet lineer olmayan hücre kullanımına göre çok iyi yaklaşım göstermemiştir. Benzer şekilde, farklı 4 adet öğrenme algoritması bu tip YSA içinde kullanılmıştır. YSA yapısının daha basit hale getirilmesine rağmen hata sonuçları Kabul edilebilir bir YSA tipi olmayacağını göstermektedir.

*Tablo 3.1.* Öğrenme Algoritmaları, YSA yapısı ve ortalama- maksimum hatalar

| Öğrenme Algoritması | Öğrenme Oranı (μ) | Momentum Terimi (α) | Çevrim Sayısı (N) | YSA Yapısı | Ortalama RMSE (Öğrenme) | Maksimum RMSE (Öğrenme) | Ortalama RMSE (Test) | Maksimum RMSE (Test) |
|---|---|---|---|---|---|---|---|---|
| GY | 0.1 | 0.0 | 1000000 | 3-10-6 | 0.485862 | 1.20600 | 0.507729 | 1.20600 |
| RGY | 0.1 | 0.0 | 1000000 | 3-10-6 | 1.210040 | 2.86877 | 1.267080 | 1.27537 |
| **HGY** | **0.1** | **0.0** | **1000000** | **3-10-6** | **0.373176** | **1.19047** | **0.374058** | **1.19047** |
| DBD | 0.1 | 0.0 | 1000000 | 3-10-6 | 0.427328 | 1.20820 | 0.418576 | 1.20820 |
| GY | 0.1 | 0.0 | 1000000 | 3-5-6 | 0.826176 | 1.73226 | 0.804328 | 1.73226 |
| RGY | 0.1 | 0.0 | 1000000 | 3-5-6 | 1.385830 | 2.72113 | 1.523790 | 2.72113 |
| **HGY** | 0.1 | 0.0 | 1000000 | 3-5-6 | 0.878670 | 1.98618 | 0.911634 | 1.98618 |
| DBD | 0.1 | 0.0 | 1000000 | 3-5-6 | 0.726305 | 1.60402 | 0.734876 | 1.34779 |

# BÖLÜM 4

# TARTIŞMA-SONUÇ ve ÖNERİLER

## 4.1.Tartışma -Sonuç

Gelişen teknoloji ile birlikte endüstriyel robotik sistemler, endüstriyel üretimde, işlemlerde çok yaygın olarak kullanılmaktadır. Endüstriyel uygulamlarda işlemlerin ve üretim tekniğinin çeşitliliğine göre farklı tipte ve yapıda robotik sistemler üretilmekte ve bu işlem çeşitliliklerinde yaygın olarak tercih edilmektedir. Robot manipülatörlerin universal çalışma yapılarından dolayı, tercih sebebi nedenleri olarak verilebilir.

Bununla birlikte, son yıllarda kumaş, özellikle jeans pantalonların sade renk olarak kullanılması pek tercih edilmemektedir. Bundan dolayı, kimyasallarla renk tonları değiştirme yönünde araştırmalar ve çalışmalar yapılmıştır. Bu renk tonu değiştirme işlemi yıllardır insan esaslı, operatörlerle gerçekleştirilmekteydi. Bu operator eseslı kimyasal ya da kumlama işlemi, sonraları kullanıcılarda, operatörlerde çok büyük ornda sağlık problemlerine, can kayıplarına, sakat kalma durumlarına sebebiyet vermektedir. Bu tür uygulamlar, daha çok ekonomik olarak gelişmemiş ülkelerde hala operatör esaslı olarak yapılmaktadır.

Son yıllarda bu problemleri ortadan kaldırmak amacıyla, robotic sistemlerin kullanılması tercih sebebi olmuştur. Robotik sistemlerin bir çok, hız, hassasiyet, esneklik, universal yapı, gibi üstünlüklerinden dolayı, jeans kimyasalları üzerine bir nozzle vasıtasıyla püskürtme gerçekleştilmektedir.

Bu işlemler seri ve otonom bir yapıda yapıldığından, kullanılan robot manipülatörün pozisyonlaması ve senkronize çalışması çok önem arz etmektedir. Belirlenen uç

işlevcinin (end-effector) pozisyonu yardımıyla, her mafsalda bulunan servo yapıdaki motorların açısal değişimi etkisi, robotic sistemlerin ters kinematiği olarak bilinmektedir.

Yapay Sinir ağları (YSA), son yıllarda bir çok alanda, system kontrolü, modellenmesi, analizi ve tahminlemesi konularında, çok yaygın ve verimli olarak kullanılmaktadır. YSA'nın yapısının adaptif ve öğrenme yetkinliğinden dolayı, gerçek zamanlı bir çok sistemde de aktif bir şekilde kullanılmaktadır.

Bu tez çalışmasında, endüstriye bir robot yapısında olan 6 serbestlik dereceli robot manipülatörün, ters kinematik analizi, YSA kullanılarak analiz edilmiştir. Burada, analiz yapılan robotic system, jeans üzerinde kimyasalları püskürterek renk değişimi sağlamada kullanılan işlem için tercih edilmiştir. Seri işlem tam otomasyonlu system yapısı ile gerçekleştildiğinden, robot manipülatörün uç işlevcisinin pozişsyon ve yörüngesi çok hassas ve her işlemde aynı konumda olması gereklidir. Bunun yanında, en optimum yapıda YSA modelini belirlemek için, ileri beslemeli, YSA ağırlık parametrelerini değiştirmede, öğrenme işleminde dört farklı yapıda öğrenme algoritmaları kullanılmıştır. Bu algoritmalar, bölüm 1'de detaylı olarak açıklandığı gibi aşağıdaki şekildeki gibi isimlendirilmiştir;

- Geri Yayılım (GY) Öğrenme Algoritması
- Random Geri Yayılım (RGY) Öğrenme Algoritması
- Hızlı Geri Yayılım (HGY) Öğrenme Algoritması
- Delta –Bar-Delta (DBD) Öğrenme Algoritması

Bu çalışmanın temel amaçlarından en önemlisi, en optimum YSA tahminleyici, modeli belirleyerek gerçek zamanlı robot manipulator yardımıyla, jeans pantalonlar üzerine nozzle yardımıyla, robot uç işlevci tarafından, püskürtülen kimyasalların, jeans üzerinde tam pozisyonlama ve yörünge planlama gerçekleştirmektir.

Yapılan simulasyon çalışmaları sonuçlarından da görüleceği gibi, robot manipülatörün ters kinematic analizi, ayrıca Bölüm 2, Şekil 2.5'de verilen blok diyagramından da

görüleceği gibi, yapay sinir ağı yapısı 3 giriş hücresi (robot uç işlevcinin poziyonu, X,Y,Z) 10 gizli katman hücresi ve 6 çıkış katmanı hücresinden, robot manipülator dönme açısal değişimleri, ($\Theta_1$ , $\Theta_2$ ,$\Theta_3$ , $\Theta_4$ , $\Theta_5$ , $\Theta_6$) oluşmakta olup, ters kinematik modeli üzerine çözümler gerçekleştirilmiştir.

İlk olarak YSA modeli, Delta-Bar-Delta algoritması kullanarak giriş katmanı ile gizli katman arası, gizli katman ile çıkış katmanı arasındaki, ileri beslemeli sinyal bağlantılarını gösteren ağırlıklar değiştirilerek öğrenme gerçekleştirilmiştir. Sonuçlardan da görüleceği gibi, Şekil 3.1'de verilen, 1.mafsal açısının **($\Theta_1$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişiminde çok küçük hatalı, Kabul edilebilir yaklaşım sonuçları görülmektedir. Bunun yanında, Şekil 3.2. grafiksel olarak detaylandırılan, 2. mafsal açısının **($\Theta_2$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişiminde belirgin oranda bazı zaman aralıklarında kabul edilmeyen açısal hatalar oluşmuştur. Benzer şekilde, Şekil 3.3 'de verildiği gibi, 3.mafsal açısının **($\Theta_3$)** Delta-Bar-Delta algoritması kullanarak YSA ve teorik yaklaşımı değişimine de açısal yaklaşım hataları mevcuttur. Şekil 3.4'de görüldüğü gibi, 4.mafsal açısının **($\Theta_4$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişiminde yaklaşım, tahminleme hataları kabul edeilebilir düzeydedir. Ronot manipülatörün uç-işelevcisi dönme açısı değişimini gösteren, Şekil 3.5'da. 5.mafsal açısının **($\Theta_5$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımı değişimi tahmineleme hataları kabul edilebilir düzeyde edeğildir. Robot manipülatörün, en son uç işlevci dönme açısı, yaklaşım sonuçları, Şekil 3.6'da, 6.mafsal açısının **($\Theta_6$)** Delta-Bar-Delta öğrenme algoritması kullanarak YSA ve teorik yaklaşımında hata olarak küçük değerlerde olduğu belirgindir.

İkinci olarak, On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi çalışması yapılmıştır. Bu yaklaşımın grefiksel sonucu, Şekil 3.7'de 1.mafsal açısının ($\Theta_1$) tahminlemesini gösteren grafiksel değişim verilmiştir. Burada, ani açısal değişim söz konusu olmadığından, çok iyi düzeyde bir yaklaşım gerçekleşmiştir. Şekil 3.8'de verilen 2.mafsal açısının ($\Theta_2$) On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişiminde is ani açısal değişimlerden dolayı, çok büyük hatasal açısal değişim farklılıkları görülmektedir. Bu açısal tahminleme yaklaşımı kanul edilebilir bir yaklaşım özelliğinde değildir. Benzer şekilde, Şekil 3.9'da 3.mafsal açısının **($\Theta_3$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımında

da kanul edilebilir hatalar mevcut değildir. Şekil 3.10. 4.mafsal açısının **($\Theta_4$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik sonucu da kabul edileilir düzeyde açısal değişim izlemiştir.

Şekil 3.11. 5.mafsal açısının **($\Theta_5$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik tahminleme grafiksel sonuçları kabul edilebilir, açısal hatalar grafiksel bir değişim izlemiştir. Şekil 3.12. 6.mafsal açısının **($\Theta_6$)** On-Line Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşım tahminelmesi ani değişim olamayan açısal dutrumdan dolayı da kabul edilebilir düzeydedir.

Üçüncü olarak gereçekşetirilen simülasyon çalışmasında, Hızlı Geri Yayılım algoritması kullanarak, YSA modelinin ağırlıkları değişimi gerçekleştirilerek öğrenme işlemi yapılmıştır. Şekil 3.13'de 1.mafsal açısının **($\Theta_1$)** YSA ve teorik yaklaşımı sonuçları grafiksel olarak verilmiştir. 1. ve 16. saniye zaman aralığında, kabul edilebilir bir açısal değişim tahminlemsi olduğu görülmektedir. Şekil 3.14'de ise 2. mafsal açısının **($\Theta_2$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişiminde ise belirgin oranda açısal sapma farklılıkları görülmektedir. Şekil 3.15'de grafiksel olarak, verilen, 3. mafsal açısının **($\Theta_3$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi de 2. Mafsal açısal değişimde olduğu gibi, belirgin hatalar oluşmuştur. Şekil 3.16'da verilem sonuçta ise 4. mafsal açısının **($\Theta_4$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi ise kabul edilebilir düzeydedir. Uç işlevcini açısal değişimi, Şekil 3.17'de 5. mafsal açısının **($\Theta_5$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı sonucu da, açısal olarak kabul edilebilir düzeyde olduğu belirgindir. Şekil 3.18. 6. mafsal açısının **($\Theta_6$)** Hızlı Geri Yayılım algoritması kullanarak YSA ve teorik, simulasyon yaklaşımı sonucu, değişimi de çok iyi sonuç vermiştir.

Dördüncü olarak, Random (Gelişigüzel) Geri Yayılım öğrenme algoritması kullanarak YSA ve teorik tahminleme yapılmıştır. Şekil 3.19. 1. mafsal açısının **($\Theta_1$),** bu algoritma kullanılarak simulasyon çalışması sonucu verilmiştir. Grafiksel yaklaşımdan da görüleceği gibi, bazı zaman aralıklarında hatalar oluşmuştur. Diğer taraftan, Şekil 3.20. 2. mafsal açısının **($\Theta_2$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi, grafiksel sonucu verilmiştir. Sonuçtan da görüleceği gibi, tehminmeme de belirgin hatalar olduğu görülmüştür. Grafiksel sonuç olarak, Şekil

3.21'de verilen, 3. mafsal açısının **($\Theta_3$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik tahminlemesi sonucu grafiksel olarak verilmiştir.

Şekil 3.22. robot manipulator uç işlevcisinin, 4. mafsal açısının **($\Theta_4$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik değişim grafiğinden de görüleceği gibi, ani açısal değişim olmamasına rağmen, belirgin oranda hatalar söz konusudur. Özellikle, başlangıç ve biriş zamanlarda ani sapmaların olduğu görülmektedir. Şekil 3.23'de ise 5. mafsal açısının **($\Theta_5$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik grafiksel değişiminde görüldüğü gibi, özellikle ani açısal değişimlerde belirgin oranda yaklaşım hataları görülmektedir. Robotik sistemin uç işlevcisinin her iki yaklaşım için, açısal değişimi, Şekil 3.24. 6. mafsal açısının **($\Theta_6$)** Random (Gelişigüzel) Geri Yayılım algoritması kullanarak YSA ve teorik yaklaşımı değişimi belli zaman aralıklarında hatalar oluşmuştur.

Diğer taraftan, önceki kısımda detaylı olarak tartışıldığı gibi, özellikle hızlı öğrenme etkisinde dolayı, gizli katmanda toplam beş adet nöron kullanılması durumunda, öğrenme etkisi çok hızlı olmasına rağmen, kullanılan toplam on adet gizli katman hücresi kullanılması sonucunda çok iyi performans sergilememiştir. Tablo 3.1'de detaylandırıldığı gibi, gizli katmada kullanılan, az hücre sayısı ve farklı öğrenme algoritmaları da dahil çok hata oranları RMSE olarak iyi düzeyde olmadığı görülmektedir.

## 4.2. Öneriler

Öneceki bölümlerde detayladırıldığı gibi, robot manipülatörün yörünge ve poizyon performansını en iyi şekilde tahminleme için, farklı yapıda öğrenme algoritmaları kullanılarak, iki tip YSA yapısında model kullanılmıştır. Burada, robotic sistemlerinters kinematic analizi için öğrenme hızı ne kadar önemli ise, çok iyi performans ile modellemed çok önem arz etmektedir. Bunun yanında, Tablo 3.1'de deyalı hata sonuçlarından da görüleceği gibi, Hızlı Geri Yayılım algoritması ile YSA modellemesi en az hata ile en iyi model olduğu sonuçunu vermiştir. Diğer taraftan, Random (Gelişi Güzel) Öğrenme Algoritması ile YSA modellenmesi her iki tip için de en büyük hatalı performansı göstermiştir. Buradan anlaşıldığı gibi, bu algoritmalı YSA modelleri gerçek zamanlı uygulamalarda tercih sebebi olmamaktadır.

Bu tez çalışmasında, endüstriyel altı serbestlik dereceli, daha çok jeans pantalonların renk değişimi işleminde nozzle vasıtasıyla uç işlevci tarafından püskürtme işelminde kullanılan, bir robot manipülatörün ters kinematik analiz farklı öğrenme algoritmaları kullanarak, yapılan simulasyon çözüm yaklaşımı, özellikle bu tür uygulamalarda kulllanılan robotik sistemlerde kinematik ve dinamik analizinde bir öncülük edebileceği belirgindir.

Önceki böümde grafiksel olarak verilen sonuçlardan da anlaşılacağı gibi, her öğrenme algoritması, bu tür sistemlerin modellenmesine çok iyi sonuç verememektedir. Bundan dolayı, en optimum öğrenme algoritmasının araştırılması kaçınılmazdır. Grafiksel sonuçlar dışında, toplam hata performansını gösteren Tablo 4.1'de, Hızlı Geri Yayılım Algoritmasının kullanılarak, robot manipülatörlerin yörünge analizinde, deneysel olarak uygulanabileceği sonucuna varılmaktadır.

Bir çok sistemin doğasında, biyolojik sistemleri de içeren, olduğu gibi, dinamik yapıya sahiptirler. Robotik sistemlerde dinamik özelliklerinden dolayı, çalışma ortamında bir çok bozucu etkenlere sahiptirler, bunlar, zemin mekaniği, hava sirkilasyonu, hava nem oranı, mekanik aksamlardaki sürtünme kuvvetleri, şeklinde sıralanbilir. Bu bozucu etkenlerden dolayı, robotic sistemlerde posizyonlama ve yöründe izleme işlemi çok hassas olması gerektiğinde, yapay sinir ağları yapısı ile modelleme ve kontrol yapılması tercih nedenlerinden biri olmalıdır. Yapay sinir ağlarının dinamik sistemlerde uygulanması çok kompleks yapıdaki çözümler üzerine olduğundan, adaptif yapısından dolayı, belirgin oranda çok iyi performanslarla yaklaşım ve tahminleme gerçekleştirilebilir.

Netice itibarıyla, çok gizli katmanlı ve gizli katmanında çok nöron olan yapay sinir ağı yapısı ile daha iyi modelleme ve tahminleme gerçekleştirilebilmektedir. Bununla birlikte, uygun öğrenme algoritması çok iyi şekilde belirlendiğinde, hızlı öğrenme ve modelleme olabilmektedir.

# KAYNAKLAR


Bigras, P., Lambert, M., and Perron, C., 2007. New Formulation for an Industrial Robot Force Controller: Real-Time Implementation on a KUKA Robot. *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2007)*, October 7–10, 2007, Montreal, QC, Canada .

Bigras, P., Lambert, M., and Perron, C., 2008. New Optimal Formulation for an Industrial Robot Force Controller, **International Journal of Robotics and Automation**, 23: 199-208.

Bigras, P., Saad, M., and O'Shea, J., 1997. Adaptive Trajectory Tracking in the Workspace of a Class of Flexible Robots. *SYROCO'97*, pp. 669-676, 1997, Nantes, France.

Bingün, Z., Küçük, S., 2009. Robot Kinematiği, Birsen Yayınevi, İstanbul.

Bonev, I.A., and Gosselin, C.M., 2000. A Geometric Algorithm for the Computation of the Constant-Orientation Workspace of 6-RUS Parallel Manipulators. *ASME Design Engineering Technical Conferences (DETC 2000)*, September 10–13, 2000, Baltimore, MD, USA.

Bonev, I.A., and Gosselin, C.M., 2001. Singularity Loci of Planar Parallel Manipulators with Revolute Joints. *2nd Workshop on Computational Kinematics*, pp. 291–299, May 20–22, 2001, Seoul, South Korea.

Bonev, I.A., and Gosselin, C.M., 2002. Geometric Algorithms for the Computation of the Constant-Orientation Workspace and Singularity Surfaces of a Special 6-RUS Parallel Manipulator. *ASME Design Engineering Technical Conferences (DETC 2002)*, September 29 – October 2, 2002, Montreal, QC, Canada,

Bonev, I.A., and Ryu, J., 1999. Workspace Analysis of 6-PRRS Parallel Manipulators Based on the Vertex Space Concept. *ASME Design Engineering Technical Conferences (DETC 1999)*, September 12–15, 1999, Las Vegas, NV, USA.

Bonev, I.A., and Ryu, J., 2001. A Geometrical Method for Computing the Constant-Orientation Workspace of 6-PRRS Parallel Manipulators. **Mechanism and Machine Theory**, 36: 1–13.

Bonev, I.A., Briot, S., Wenger, P., et Chablat, D., 2008. Changing Assembly modes without Passing Parallel Singularities in Non-Cuspidal 3-RPR Planar Parallel Robots. *Second International Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, September 21–22, 2008, Montpellier, France.

Bonev, I.A., Ryu, J., Kim, S.-G., and Lee, S.-K., 2001. A Closed-Form Solution to the Direct Kinematics of Nearly General Parallel Manipulators with Optimally Located Three Linear Extra Sensors. **IEEE Transactions on Robotics and Automation,** 17: 148–156.

Bonev, I.A., Ryu, J., Lee, S.-K., and Kim, N.-J., 1999. A Simple New Closed-Form Solution of the Direct Kinematics of Parallel Manipulators Using Three Extra Sensors. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 1999)*, pp. 526–530, September 19–22, 1999, Atlanta, GA, USA.

Bonev, I.A., Zlatanov, D., and Gosselin, C.M., 2003. Singularity Analysis of 3-DOF Planar Parallel Mechanisms via Screw Theory. **ASME Journal of Mechanical Design,** 125: 573–581.

Briot, S., and Bonev, I.A., 2008. A New Fully Decoupled 3-DOF Translational Parallel Robot for Pick-and-Place Applications. **Journal of Mechanisms and Robotics**, 1: 1-9.

Briot, S., and Bonev, I.A., 2008. Accuracy Analysis of 3-DOF Planar Parallel Robots. **Mechanism and Machine Theory**, 43: 445-458.

Briot, S., Arakelian, V., Bonev, I.A., Chablat, D., and Wenger, P., 2008. Self Motions of General 3-RPR Planar Parallel Robots, **The International Journal of Robotics Research**, 27: 855–866.

Chablat, D., Bonev, I.A., and Wenger, P., 2006. Self Motions of a Special 3-RPR Planar Parallel Robot. *10th International Symposium on Advances in Robot Kinematics (ARK 2006)*, pp. 221–228, June 25–29, 2006, Ljubljana, Slovenia.

Craig, J.J. 1989. Introduction to Robotics: Mechanics and Control., USA.

Duran, M.A., Ankaralı, A., 2010. Üç serbestlik dereceli Puma tipi bir manipülatörün PID kontrolü. **Selçuk-Teknik Dergisi**, 9: 79-98.

Figielski, A., Bonev, I.A., and Bigras, P., 2007. Towards Development of a 2-DOF Planar Parallel Robot with Optimal Workspace Use. *IEEE*

*International Conference on Systems, Man, and Cybernetics (SMC 2007)*, October 7–10, 2007, Montreal, QC, Canada .

Hamelin, P., Bigras, P., Beaudry, J., Lemieux, S., and Blain, M., 2008. Hardware-in-the-loop Simulation of an Impedance Controlled Robot Using a Direct-Drive Test Bench. *IEEE International Symposium on Industrial Electronics*, July, 2008, Cambridge UK.

Hamelin, P., Bigras, P., Lemieux, S., and Blain, M., 2008. Hardware-in-the-loop Simulation of an Impedance Controlled Robot Interacting with a Real Environment. *10th IASTED International Conference on Control and Applications*, May, 2008, Quebec, Canada.

Haykin, S. 2009. N Neural Networks and Learning Machines Third Edition. Pearson Prentice Hall, New York, 938 pp.

https://www.universal-robots.com/tr/urunler/ur5-robot/

Janvier, M.-A., Durant, L.G., Roy Cardinal, M.H., Renaud, I., Chayer, B., Bigras, P., de Guise, J., Soulez, G., and Cloutier, G., 2007. Performance Evaluation of a Medical Robotic 3D-Ultrasound Imaging System. **Medical Image Analysis Journal,** 12: 275-290.

Janvier, M.-A., Mercure, É., Godbout, B., Chayer, B., Allard, L., Bigras, P., Deguise, J., Soulez, G., and Cloutier, G., 2005. Evaluation of a robot's optimal performance for medical 3D-US imaging. *Medical Image Computing and Computer Assisted Intervention*, 2005, Palm Springs, CA, USA .

Khayati, K., Bigras, P., and Dessaint, L-A, 2006. A Multi-stage Position/Force Control for Constrained Robotic Systems with Friction: Joint-space Decomposition, Linearization and Multi-objective Observer/Controller Synthesis using LMI Formalism. **IEEE Transactions on Industrial Electronics**, 53:1698-1712.

Khayati, K., Bigras, P., and Dessaint, L-A., 2004. A Robust Feedback Linearization Force Control of a Pneumatic Actuator. *IEEE International Conference on Systems, Man and Cybernetics*, pp. 6113-6119, October, 2004, Hague, The Netherlands

Kuşcu, H., 2015. Temel Robotik Ders Notları, Trakya Üniversitesi.

Lessard, S., Bigras, P., Bonev, I.A., 2007. A New Medical Parallel Robot and its Static Balancing Optimization. **Journal of Medical Devices**, 1: 272-278.

Lessard, S., Bigras, P., Bonev, I.A., Briot, S., and Arakelyan, V., 2007. Optimum Static Balancing of the Parallel Robot for Medical 3D-Ultrasound Imaging. *The 12th World Congress in Mechanism and Machine Science (IFToMM 2007)*, June 17–21, 2007, Besançon, France

Lessard, S., Bonev, I.A., Bigras, P., Durand, L.-G., Soulez, G., Cloutier, G., and De Guise, J.A., 2006. Parallel Robot for Medical 3D-Ultrasound Imaging. *IEEE International Symposium on Industrial Electronics (ISIE 2006)*, July 9–13, 2006, Montreal, QC, Canada,.

Liu, X.-J., and Bonev, I.A., 2008. Orientation Capability, Error Analysis and Dimension Optimization of Two Articulated Tool Heads with Parallel Kinematics. **Journal of Manufacturing Science and Engineering**, 130: 1-25.

Lupien, S., Lambert, G., Bigras, P., Lessard, P., Demers, J.-C., and Sankar, T. S., 2001. A Versatile Design Platform for a Walking Robot in Harsh Environments. *ASME Design Engineering Technical Conference*, pp. 841-849, September, 2001, Pittsburgh, PA, USA.

Poiré, V., Lemieux, S., Bigras, P., and Blain, M., 2006. Migration Procedure of Control Laws from a Graphical Simulation Environment to a Robotic Framework. *IASTED International Conference on Control and Aplications*, May, 2006, Montreal, QC, Canada.

Saad, M., Dessaint, L-A., Bigras, P., and Al Haddad, K., 1994. Adaptive Versus Neural Adaptive Control: Application to Robotics. **International Journal of Adaptive Control and Signal Processing**, 8: 223–236.

Yu, A., Bonev, I.A., and Zsombor-Murray, P., 2008. Geometric Approach to the Accuracy Analysis of a Class of 3-DOF Planar Parallel Robots. **Mechanism and Machine Theory,** 43:363-375**.**

Zlatanov, D., Bonev, I.A., and Gosselin, C.M., 2002. Constraint Singularities as C-Space Singularities. *8th International Symposium on Advances in Robot Kinematics (ARK 2002)*, 24–28 June, 2002, Caldes de Malavella, Spain.

# EKLER

## EK 1. Geri Yayılım (GY) Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++ Kodu

```
 * Structure:        3/10/6

#include <math.h>

#include <stdlib.h>

#define INPUTS    3

#define HIDDEN1   10

#define OUTPUTS   6

#define HIDDEN1_INDEX  (INPUTS)

#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)

#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)

static double NEURON[NUMNEURONS];

/* Weight arrays */

static const double W_In_H1[HIDDEN1][INPUTS+1] = {

 -0.672949, -0.674642, -1.22214, -1.02272,

 1.88585, -0.441412, 1.28496, -0.411775,

 1.30373, 0.545383, -3.33235, -0.115032,

 -2.18836, 0.491685, -1.79041, 1.30606,

 -1.75734, -0.36396, -0.228791, -0.514939,

 2.04491, -0.193454, 2.00638, 1.63257,

 0.378226, 1.1819, 0.352562, 0.214003,

 2.77662, -0.729806, 1.86581, -1.80812,
```

```
-1.57758, -0.381939, -0.956764, -1.37512,

-0.0111766, -0.098062, -0.748642, -0.0913002

};

static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {

-0.380939, 1.89814, -0.297288, -2.40301, -1.23765, 1.57455, 0.261426, 2.27464, -0.732586, -0.244238, -0.0392707,

1.45304, 0.532878, 0.589188, -1.36328, 1.05254, -2.19995, -0.935175, 2.29009, 1.85215, 0.243157, -0.13248,

0.128554, -0.844341, 3.33395, 0.896567, -0.525615, -0.0319206, -0.258965, -1.42655, -0.57994, 0.544749, -0.432615,

-0.651476, 0.482043, -1.72697, -0.407795, -0.37778, 1.65626, 0.627956, 0.344556, -0.601274, -0.184823, 0.189519,

0.580346, -1.94329, 0.153609, 2.4237, 1.28397, -1.24931, -0.230842, -2.40977, 0.780393, 0.103477, 0.0285127,

0.364278, -2.06523, 0.256051, 2.18238, 0.977424, -1.95932, -0.666382, -1.91441, 1.03983, -0.0248682, -0.214852

};

/* Scale factors */

static const double R_MIN[INPUTS+OUTPUTS] = {

-101, -613, 553, -326, 247, -283, -272, -78, 195

};

static const double R_MAX[INPUTS+OUTPUTS] = {

398, -583, 1158, -259, 273, -240, -171, -18, 247

};
```

```
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
static double linear( double x ) { return x; }
/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
 *   outputs  pointer to (double) array which receives the results
 * Return:
 *   NonZero (1) if successful, 0 if not.
 */
int eval_net(const double * inputs, double * outputs)
{
 double F;
 int i,j;
```

```
/* Check for invalid arguments */
if (inputs == NULL || outputs == NULL)
    return 0;
/* Zero neuron sums */
for (i=0;i<NUMNEURONS;i++)
    NEURON[i] = 0.0;
```

**EK 2. Random (Gelişi Güzel) Geri Yayılım (RGY) Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++ Kodu**

```
* Structure:        3/10/6
#include <math.h>
#include <stdlib.h>
#define INPUTS    3
#define HIDDEN1   10
#define OUTPUTS   6
#define HIDDEN1_INDEX  (INPUTS)
#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)
#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)
static double NEURON[NUMNEURONS];
/* Weight arrays */
static const double W_In_H1[HIDDEN1][INPUTS+1] = {
 -1.86267, 0.499485, -1.34933, -0.611446,
 -1.82114, -0.234928, -0.501528, -0.965629,
```

2.02095, -0.0347862, 0.450193, 0.648579,

-0.727786, 0.0192121, 1.83205, 0.205981,

0.646383, -0.274761, -1.79683, -0.181484,

0.16334, 0.921794, 2.10621, 1.46144,

-1.9012, 0.838895, -2.01431, 1.08287,

-1.64362, -0.305721, -1.48357, -1.54454,

0.37363, 0.869103, -1.84393, -0.256248,

3.2759, -1.00136, 2.20488, -1.87089

};

static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {

-2.05537, -1.26174, 1.86308, 0.174164, -0.40703, 0.387733, -1.34729, -1.19471, -0.0820784, 2.60111, 0.375371,

0.621633, 1.2706, -0.596049, -0.301455, 0.969416, -2.09028, -0.991537, 2.06591, -0.253953, 2.60514, -0.0296526,

-0.178509, -0.736212, 0.51207, -1.71688, 1.68093, -0.429046, 1.6637, -0.23933, 1.80883, -1.60794, -0.0893305,

-0.820156, -0.33852, 0.582735, 1.0523, -1.04721, 1.29131, -0.53499, -0.888475, -0.646662, 0.367467, 0.116246,

1.95851, 1.33847, -2.04932, -0.127132, 0.525787, -0.397596, 1.45577, 1.07042, 0.0444554, -2.72693, -0.155254,

1.96256, 1.13096, -2.08665, -0.357962, 0.26441, -0.903684, 1.37432, 1.22746, -0.180879, -2.15006, -0.700679

};

```
/* Scale factors */
static const double R_MIN[INPUTS+OUTPUTS] = {
 -101, -613, 553, -326, 247, -283, -272, -78, 195
 };
static const double R_MAX[INPUTS+OUTPUTS] = {
 398, -583, 1158, -259, 273, -240, -171, -18, 247
 };
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
static double linear( double x ) { return x; }
/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
 *   outputs  pointer to (double) array which receives the results
```

```
* Return:
*   NonZero (1) if successful, 0 if not.
int eval_net(const double * inputs, double * outputs)
{
double F;
int i,j;
/* Check for invalid arguments */
if (inputs == NULL || outputs == NULL)
   return 0;
/* Zero neuron sums */
for (i=0;i<NUMNEURONS;i++)
   NEURON[i] = 0.0;

/* Apply inputs */
for (i=0;i<INPUTS;i++)
{
   F = (S_MAX[i]-S_MIN[i])/(R_MAX[i]-R_MIN[i]); /* scale factor */
   NEURON[i] = F * inputs[i] + S_MIN[i] - F * R_MIN[i];
}
/* Compute hidden layer #1 */
for (i=0;i<HIDDEN1;i++)
{
```

```
    for (j=0;j<INPUTS;j++)
    {
       NEURON[HIDDEN1_INDEX + i] += NEURON[j]*W_In_H1[i][j];
    }
    NEURON[HIDDEN1_INDEX + i] += W_In_H1[i][INPUTS]; /* BIAS */
    NEURON[HIDDEN1_INDEX + i] = sigmoid( NEURON[HIDDEN1_INDEX + i] );
 }
 /* Compute network outputs */
 for (i=0;i<OUTPUTS;i++)
 {
    for (j=0;j<HIDDEN1;j++)
    {
       NEURON[OUTPUT_INDEX + i] += NEURON[HIDDEN1_INDEX + j]*W_H1_Out[i][j];
    }
    NEURON[OUTPUT_INDEX + i] += W_H1_Out[i][HIDDEN1]; /* BIAS */
    NEURON[OUTPUT_INDEX + i] = sigmoid( NEURON[OUTPUT_INDEX + i] );
 }
 /* Copy outputs to output array */
 for (i=0;i<OUTPUTS;i++)
 {
    F = (R_MAX[i+INPUTS]-R_MIN[i+INPUTS])/(S_MAX[i+INPUTS]-S_MIN[i+INPUTS]); /* unscale factor */
```

```
    outputs[i] = F * NEURON[OUTPUT_INDEX + i] + R_MIN[i+INPUTS] - F * S_MIN[i+INPUTS];
 }
 return 1;
 }
```

**EK 3. Hızlı Geri Yayılım (HGY) Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++ Kodu**

```
 * Structure:        3/10/6
 #include <math.h>
#include <stdlib.h>
#define INPUTS    3
#define HIDDEN1   10
#define OUTPUTS   6
#define HIDDEN1_INDEX  (INPUTS)
#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)
#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)
static double NEURON[NUMNEURONS];
/* Weight arrays */
static const double W_In_H1[HIDDEN1][INPUTS+1] = {
 0.774149, 0.391923, -3.39983, -0.441169,
 3.65308, 0.00399228, -5.1931, 0.190298,
 -1.83391, 0.0701449, -24.6767, 0.77055,
```

```
-5.43204, 0.610162, -0.443802, -2.08348,

6.50421, 0.586302, 1.40052, 6.10743,

0.555714, -2.38615, -9.53359, -0.653112,

-3.20868, -1.19295, 5.67105, -0.365443,

3.38588, 0.999492, 1.86655, 0.335125,

4.2816, -2.43851, 1.31325, -1.61728,

-1.21129, -0.636719, 9.00841, 4.95727

};

static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {

-0.212194, -1.04472, -3.42049, -1.71503, 0.966496, -0.636668, 0.244034, 1.04499, 4.8499, 0.878409, -0.0766877,

1.06488, 0.562009, 0.0183961, 2.35255, -1.54519, -0.124392, 1.19997, -0.375168, 3.77699, -2.75656, 0.333269,

-0.31484, 1.37577, 2.23075, -0.30517, 1.62534, 1.23051, -2.56892, -0.685329, -1.6826, -0.73537, 0.0915082,

-0.60118, -1.00672, -1.20383, -1.37111, 0.320295, -0.176139, 0.73152, 1.14802, 0.104312, 0.815841, 0.32834,

0.230496, 0.404323, 2.50751, 1.71814, -0.88079, 0.799843, -0.06175, -0.801187, -5.32624, -0.643505, 0.642326,

0.978962, 0.736277, 2.27135, 1.58497, -1.89216, 0.336719, 0.541211, -0.802015, -5.17994, -1.36123, 0.490559

};

/* Scale factors */

static const double R_MIN[INPUTS+OUTPUTS] = {
```

```
 -101, -613, 553, -326, 247, -283, -272, -78, 195
 };
static const double R_MAX[INPUTS+OUTPUTS] = {
 398, -583, 1158, -259, 273, -240, -171, -18, 247
 };
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
static double linear( double x ) { return x; }
/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
 *   outputs  pointer to (double) array which receives the results
 * Return:
 *   NonZero (1) if successful, 0 if not.
```

```
*/ int eval_net(const double * inputs, double * outputs)
{
 double F;
 int i,j;
 /* Check for invalid arguments */
 if (inputs == NULL || outputs == NULL)
    return 0;
```

**EK 4. Delta-Bar-Delta (DBD) Öğrenme Algoritması ile 3-10-6 tipi YSA Modeli C++ Kodu**

```
* Structure:        3/10/6
 * Date:             Thu Apr 26 12:19:51 2018
#include <math.h>
#include <stdlib.h>
#define INPUTS    3
#define HIDDEN1   10
#define OUTPUTS   6
#define HIDDEN1_INDEX  (INPUTS)
#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)
#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)
static double NEURON[NUMNEURONS];
/* Weight arrays */
static const double W_In_H1[HIDDEN1][INPUTS+1] = {
```

-1.32954, 0.726328, -6.66798, 0.215447,

-2.17003, 0.417187, -1.05755, -2.10598,

-6.68001, 1.02396, -1.3107, 4.66308,

2.55047, -2.28772, -2.54617, -0.72955,

-4.3007, -3.20624, 8.72024, 0.00853733,

1.06546, 0.847043, -0.200514, 0.358764,

4.69141, -0.2352, 1.06677, 3.7571,

-0.183777, 0.62334, 5.29216, -0.32462,

-1.81246, -0.383659, 0.266011, -0.782861,

1.58618, 0.345375, 0.138706, 0.76858

};

static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {

-2.48229, -0.720046, -2.71019, 0.841727, 1.06587, 0.714111, 2.46253, 1.4397, -3.30041, 1.02763, 0.165009,

-0.0573852, 5.91178, -3.11107, 1.6984, 0.446162, -0.85211, -0.120497, 0.555844, 0.439492, -0.491438, 0.25833,

0.160143, -0.721489, 2.72326, 3.35002, -3.22301, -0.420049, -0.0757006, -0.116112, 0.0902455, -0.413088, -0.0986093,

-0.376985, -1.94016, -0.369548, -1.59985, 1.54836, 0.345268, 0.496758, 0.405025, -1.35078, 0.736258, 0.228735,

1.85146, 0.582976, 3.01418, -0.725981, -1.75559, -0.900729, -2.63989, -1.36498, 4.79071, -0.715434, -0.118867,

2.07588, 2.47049, 2.14054, -0.347312, -1.13204, -0.855885, -2.37778, -1.68199, 2.04097, -1.21283, -0.250657

```
 };
/* Scale factors */
static const double R_MIN[INPUTS+OUTPUTS] = {
 -101, -613, 553, -326, 247, -283, -272, -78, 195
 };
static const double R_MAX[INPUTS+OUTPUTS] = {
 398, -583, 1158, -259, 273, -240, -171, -18, 247
 };
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
static double linear( double x ) { return x; }
/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
```

```
*   outputs  pointer to (double) array which receives the results
* Return:
*   NonZero (1) if successful, 0 if not.
*/
int eval_net(const double * inputs, double * outputs)
{
 double F;
 int i,j;
 /* Check for invalid arguments */
 if (inputs == NULL || outputs == NULL)
    return 0;

 /* Zero neuron sums */
 for (i=0;i<NUMNEURONS;i++)
    NEURON[i] = 0.0;
 /* Apply inputs */
 for (i=0;i<INPUTS;i++)
 {
    F = (S_MAX[i]-S_MIN[i])/(R_MAX[i]-R_MIN[i]); /* scale factor */
    NEURON[i] = F * inputs[i] + S_MIN[i] - F * R_MIN[i];
 }
 /* Compute hidden layer #1 */
```

```
for (i=0;i<HIDDEN1;i++)
{
   for (j=0;j<INPUTS;j++)
   {
      NEURON[HIDDEN1_INDEX + i] += NEURON[j]*W_In_H1[i][j];
   }
   NEURON[HIDDEN1_INDEX + i] += W_In_H1[i][INPUTS]; /* BIAS */
   NEURON[HIDDEN1_INDEX + i] = sigmoid( NEURON[HIDDEN1_INDEX + i] );
}
/* Compute network outputs */
for (i=0;i<OUTPUTS;i++)
{
   for (j=0;j<HIDDEN1;j++)
   {
      NEURON[OUTPUT_INDEX + i] += NEURON[HIDDEN1_INDEX + j]*W_H1_Out[i][j];
   }
   NEURON[OUTPUT_INDEX + i] += W_H1_Out[i][HIDDEN1]; /* BIAS */
   NEURON[OUTPUT_INDEX + i] = sigmoid( NEURON[OUTPUT_INDEX + i] );
}
/* Copy outputs to output array */
for (i=0;i<OUTPUTS;i++)
{
```

```
    F = (R_MAX[i+INPUTS]-R_MIN[i+INPUTS])/(S_MAX[i+INPUTS]-
S_MIN[i+INPUTS]); /* unscale factor */

    outputs[i] = F * NEURON[OUTPUT_INDEX + i] + R_MIN[i+INPUTS] - F *
S_MIN[i+INPUTS];

 }

 return 1;

 }
```

**EK 5. Geri Yayılım (GY) Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu**

```
* Structure:      3/5/6

 * Date:          Thu Apr 26 12:08:44 2018

#include <math.h>

#include <stdlib.h>

#define INPUTS    3

#define HIDDEN1   5

#define OUTPUTS   6

#define HIDDEN1_INDEX  (INPUTS)

#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)

#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)

static double NEURON[NUMNEURONS];

/* Weight arrays */

static const double W_In_H1[HIDDEN1][INPUTS+1] = {
```

```
 1.84882, 0.636135, 1.64022, 0.817593,

 -2.02711, -0.55207, -2.26386, -2.26781,

 -0.00101909, 1.44127, -3.57144, 0.0415121,

 -3.51794, 1.01531, -1.9218, 1.81747,

 2.35932, -0.276837, 1.20783, 0.0569519

 };

static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {

 2.30338, -2.3716, -1.82691, -2.27385, 2.85749, 0.00277103,

 -0.998154, 4.08201, -0.0351561, -2.34327, -0.159513, 0.816091,

 -0.829982, -0.993476, 3.34504, 1.32615, -0.180329, -1.08107,

 1.50331, -1.74388, -1.62955, -0.0273387, 0.663595, 0.392221,

 -2.25213, 2.15565, 1.82082, 2.53535, -2.99084, -0.0136871,

 -2.73568, 2.13497, 1.54714, 1.7957, -2.53053, -0.351847

 };

/* Scale factors */

static const double R_MIN[INPUTS+OUTPUTS] = {

 -101, -613, 553, -326, 247, -283, -272, -78, 195

 };

static const double R_MAX[INPUTS+OUTPUTS] = {

 398, -583, 1158, -259, 273, -240, -171, -18, 247

 };

static const double S_MIN[INPUTS+OUTPUTS] = {
```

```
-1, -1, -1, 0, 0, 0, 0, 0, 0
};
static const double S_MAX[INPUTS+OUTPUTS] = {
1, 1, 1, 1, 1, 1, 1, 1, 1
};
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
static double linear( double x ) { return x; }
/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
 *   outputs  pointer to (double) array which receives the results
 * Return:
 *   NonZero (1) if successful, 0 if not.
 */
int eval_net(const double * inputs, double * outputs)
{
 double F;
 int i,j;
 /* Check for invalid arguments */
```

```
if (inputs == NULL || outputs == NULL)
   return 0;
/* Zero neuron sums */
for (i=0;i<NUMNEURONS;i++)
   NEURON[i] = 0.0;
/* Apply inputs */
for (i=0;i<INPUTS;i++)
{
   F = (S_MAX[i]-S_MIN[i])/(R_MAX[i]-R_MIN[i]); /* scale factor */
   NEURON[i] = F * inputs[i] + S_MIN[i] - F * R_MIN[i];
}
/* Compute hidden layer #1 */
for (i=0;i<HIDDEN1;i++)
{
   for (j=0;j<INPUTS;j++)
   {
      NEURON[HIDDEN1_INDEX + i] += NEURON[j]*W_In_H1[i][j];
   }
   NEURON[HIDDEN1_INDEX + i] += W_In_H1[i][INPUTS]; /* BIAS */
   NEURON[HIDDEN1_INDEX + i] = sigmoid( NEURON[HIDDEN1_INDEX + i] );
}
/* Compute network outputs */
```

```
for (i=0;i<OUTPUTS;i++)
{
   for (j=0;j<HIDDEN1;j++)
   {
      NEURON[OUTPUT_INDEX + i] += NEURON[HIDDEN1_INDEX + j]*W_H1_Out[i][j];
   }
   NEURON[OUTPUT_INDEX + i] += W_H1_Out[i][HIDDEN1]; /* BIAS */
   NEURON[OUTPUT_INDEX + i] = sigmoid( NEURON[OUTPUT_INDEX + i] );
}
/* Copy outputs to output array */
for (i=0;i<OUTPUTS;i++)
{
   F = (R_MAX[i+INPUTS]-R_MIN[i+INPUTS])/(S_MAX[i+INPUTS]-S_MIN[i+INPUTS]); /* unscale factor */
   outputs[i] = F * NEURON[OUTPUT_INDEX + i] + R_MIN[i+INPUTS] - F * S_MIN[i+INPUTS];
}
return 1;
}
```

**EK 6. RGY Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu**

```
* Structure:        3/5/6

#include <math.h>

#include <stdlib.h>

#define INPUTS    3

#define HIDDEN1   5

#define OUTPUTS   6

#define HIDDEN1_INDEX  (INPUTS)

#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)

#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)

static double NEURON[NUMNEURONS];

/* Weight arrays */

static const double W_In_H1[HIDDEN1][INPUTS+1] = {

 1.84882, 0.636135, 1.64022, 0.817593,

 -2.02711, -0.55207, -2.26386, -2.26781,

 -0.00101909, 1.44127, -3.57144, 0.0415121,

 -3.51794, 1.01531, -1.9218, 1.81747,

 2.35932, -0.276837, 1.20783, 0.0569519

 };

static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {

 2.30338, -2.3716, -1.82691, -2.27385, 2.85749, 0.00277103,

 -0.998154, 4.08201, -0.0351561, -2.34327, -0.159513, 0.816091,
```

```
 -0.829982, -0.993476, 3.34504, 1.32615, -0.180329, -1.08107,
 1.50331, -1.74388, -1.62955, -0.0273387, 0.663595, 0.392221,
 -2.25213, 2.15565, 1.82082, 2.53535, -2.99084, -0.0136871,
 -2.73568, 2.13497, 1.54714, 1.7957, -2.53053, -0.351847
 };
/* Scale factors */
static const double R_MIN[INPUTS+OUTPUTS] = {
 -101, -613, 553, -326, 247, -283, -272, -78, 195
 };
static const double R_MAX[INPUTS+OUTPUTS] = {
 398, -583, 1158, -259, 273, -240, -171, -18, 247
 };
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
```

```
static double linear( double x ) { return x; }

/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
 *   outputs  pointer to (double) array which receives the results
 * Return:
 *   NonZero (1) if successful, 0 if not.
int eval_net(const double * inputs, double * outputs)
{
 double F;
 int i,j;
 /* Check for invalid arguments */
 if (inputs == NULL || outputs == NULL)
    return 0;
 /* Zero neuron sums */
 for (i=0;i<NUMNEURONS;i++)
    NEURON[i] = 0.0;
 /* Apply inputs */
 for (i=0;i<INPUTS;i++)
 {
    F = (S_MAX[i]-S_MIN[i])/(R_MAX[i]-R_MIN[i]); /* scale factor */
    NEURON[i] = F * inputs[i] + S_MIN[i] - F * R_MIN[i];
```

```
}
/* Compute hidden layer #1 */
for (i=0;i<HIDDEN1;i++)
{
   for (j=0;j<INPUTS;j++)
   {
      NEURON[HIDDEN1_INDEX + i] += NEURON[j]*W_In_H1[i][j];
   }
   NEURON[HIDDEN1_INDEX + i] += W_In_H1[i][INPUTS]; /* BIAS */
   NEURON[HIDDEN1_INDEX + i] = sigmoid( NEURON[HIDDEN1_INDEX + i] );
}
/* Compute network outputs */
for (i=0;i<OUTPUTS;i++)
{
   for (j=0;j<HIDDEN1;j++)
   {
      NEURON[OUTPUT_INDEX + i] += NEURON[HIDDEN1_INDEX + j]*W_H1_Out[i][j];
   }
   NEURON[OUTPUT_INDEX + i] += W_H1_Out[i][HIDDEN1]; /* BIAS */
   NEURON[OUTPUT_INDEX + i] = sigmoid( NEURON[OUTPUT_INDEX + i] );
}
/* Copy outputs to output array */
```

```
 for (i=0;i<OUTPUTS;i++)
 {
   F = (R_MAX[i+INPUTS]-R_MIN[i+INPUTS])/(S_MAX[i+INPUTS]-S_MIN[i+INPUTS]); /* unscale factor */
   outputs[i] = F * NEURON[OUTPUT_INDEX + i] + R_MIN[i+INPUTS] - F * S_MIN[i+INPUTS];
 }
 return 1;
 }
```

**EK 7. HGY Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu**

```
 * Structure:      3/5/6
 * Date:           February 02 12:17:00 2021
#include <math.h>
#include <stdlib.h>
#define INPUTS    3
#define HIDDEN1   5
#define OUTPUTS   6
#define HIDDEN1_INDEX  (INPUTS)
#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)
#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)
static double NEURON[NUMNEURONS];
/* Weight arrays */
```

```
static const double W_In_H1[HIDDEN1][INPUTS+1] = {
 1.84882, 0.636135, 1.64022, 0.817593,
 -2.02711, -0.55207, -2.26386, -2.26781,
 -0.00101909, 1.44127, -3.57144, 0.0415121,
 -3.51794, 1.01531, -1.9218, 1.81747,
 2.35932, -0.276837, 1.20783, 0.0569519
 };
static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {
 2.30338, -2.3716, -1.82691, -2.27385, 2.85749, 0.00277103,
 -0.998154, 4.08201, -0.0351561, -2.34327, -0.159513, 0.816091,
 -0.829982, -0.993476, 3.34504, 1.32615, -0.180329, -1.08107,
 1.50331, -1.74388, -1.62955, -0.0273387, 0.663595, 0.392221,
 -2.25213, 2.15565, 1.82082, 2.53535, -2.99084, -0.0136871,
 -2.73568, 2.13497, 1.54714, 1.7957, -2.53053, -0.351847
 };
/* Scale factors */
static const double R_MIN[INPUTS+OUTPUTS] = {
 -101, -613, 553, -326, 247, -283, -272, -78, 195
 };
static const double R_MAX[INPUTS+OUTPUTS] = {
 398, -583, 1158, -259, 273, -240, -171, -18, 247
 };
```

```
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
static double gaussian( double x ) { return exp( -x*x/2 ); }
static double linear( double x ) { return x; }
/* Compute the network response
 * Arguments:
 *   inputs   pointer to (double) array containing the input values
 *   outputs  pointer to (double) array which receives the results
 * Return:
 *   NonZero (1) if successful, 0 if not.
int eval_net(const double * inputs, double * outputs)
{
 double F;
 int i,j;
 /* Check for invalid arguments */
```

```
if (inputs == NULL || outputs == NULL)
   return 0;
/* Zero neuron sums */
for (i=0;i<NUMNEURONS;i++)
   NEURON[i] = 0.0;
/* Apply inputs */
for (i=0;i<INPUTS;i++)
{
   F = (S_MAX[i]-S_MIN[i])/(R_MAX[i]-R_MIN[i]); /* scale factor */
   NEURON[i] = F * inputs[i] + S_MIN[i] - F * R_MIN[i];
}
/* Compute hidden layer #1 */
for (i=0;i<HIDDEN1;i++)
{
   for (j=0;j<INPUTS;j++)
   {
      NEURON[HIDDEN1_INDEX + i] += NEURON[j]*W_In_H1[i][j];
   }
   NEURON[HIDDEN1_INDEX + i] += W_In_H1[i][INPUTS]; /* BIAS */
   NEURON[HIDDEN1_INDEX + i] = sigmoid( NEURON[HIDDEN1_INDEX + i] );
}
/* Compute network outputs */
```

```
for (i=0;i<OUTPUTS;i++)
{
   for (j=0;j<HIDDEN1;j++)
   {
      NEURON[OUTPUT_INDEX + i] += NEURON[HIDDEN1_INDEX + j]*W_H1_Out[i][j];
   }
   NEURON[OUTPUT_INDEX + i] += W_H1_Out[i][HIDDEN1]; /* BIAS */
   NEURON[OUTPUT_INDEX + i] = sigmoid( NEURON[OUTPUT_INDEX + i] );
}
/* Copy outputs to output array */
for (i=0;i<OUTPUTS;i++)
{
   F = (R_MAX[i+INPUTS]-R_MIN[i+INPUTS])/(S_MAX[i+INPUTS]-S_MIN[i+INPUTS]); /* unscale factor */
   outputs[i] = F * NEURON[OUTPUT_INDEX + i] + R_MIN[i+INPUTS] - F * S_MIN[i+INPUTS];
}
return 1;
}
```

**EK 8. DBD Öğrenme Algoritması ile 3-5-6 tipi YSA Modeli C++ Kodu**

```
* Structure:        3/5/6
 * Date:            February 02 12:17:35 2021
#include <math.h>
#include <stdlib.h>
#define INPUTS    3
#define HIDDEN1   5
#define OUTPUTS   6
#define HIDDEN1_INDEX  (INPUTS)
#define OUTPUT_INDEX   (HIDDEN1_INDEX+HIDDEN1)
#define NUMNEURONS  (INPUTS+HIDDEN1+OUTPUTS)
static double NEURON[NUMNEURONS];
/* Weight arrays */
static const double W_In_H1[HIDDEN1][INPUTS+1] = {
 1.84882, 0.636135, 1.64022, 0.817593,
 -2.02711, -0.55207, -2.26386, -2.26781,
 -0.00101909, 1.44127, -3.57144, 0.0415121,
 -3.51794, 1.01531, -1.9218, 1.81747,
 2.35932, -0.276837, 1.20783, 0.0569519
 };
static const double W_H1_Out[OUTPUTS][HIDDEN1+1] = {
 2.30338, -2.3716, -1.82691, -2.27385, 2.85749, 0.00277103,
```

```
 -0.998154, 4.08201, -0.0351561, -2.34327, -0.159513, 0.816091,
 -0.829982, -0.993476, 3.34504, 1.32615, -0.180329, -1.08107,
 1.50331, -1.74388, -1.62955, -0.0273387, 0.663595, 0.392221,
 -2.25213, 2.15565, 1.82082, 2.53535, -2.99084, -0.0136871,
 -2.73568, 2.13497, 1.54714, 1.7957, -2.53053, -0.351847
 };
/* Scale factors */
static const double R_MIN[INPUTS+OUTPUTS] = {
 -101, -613, 553, -326, 247, -283, -272, -78, 195
 };
static const double R_MAX[INPUTS+OUTPUTS] = {
 398, -583, 1158, -259, 273, -240, -171, -18, 247
 };
static const double S_MIN[INPUTS+OUTPUTS] = {
 -1, -1, -1, 0, 0, 0, 0, 0, 0
 };
static const double S_MAX[INPUTS+OUTPUTS] = {
 1, 1, 1, 1, 1, 1, 1, 1, 1
 };
/* Nonlinearity functions */
static double sigmoid( double x ) { return(1.0/(1.0 + exp(-x))); }
static double ftanh( double x ) { return tanh( x ); }
```

```
static double gaussian( double x ) { return exp( -x*x/2 ); }

static double linear( double x ) { return x; }

/* Compute the network response

 * Arguments:

 *   inputs   pointer to (double) array containing the input values

 *   outputs  pointer to (double) array which receives the results

 * Return:

 *   NonZero (1) if successful, 0 if not.

 */

int eval_net(const double * inputs, double * outputs)

{

 double F;

 int i,j;

 /* Check for invalid arguments */

 if (inputs == NULL || outputs == NULL)

    return 0;

 /* Zero neuron sums */

 for (i=0;i<NUMNEURONS;i++)

    NEURON[i] = 0.0;

 /* Apply inputs */

 for (i=0;i<INPUTS;i++)

 {
```

```
    F = (S_MAX[i]-S_MIN[i])/(R_MAX[i]-R_MIN[i]); /* scale factor */
    NEURON[i] = F * inputs[i] + S_MIN[i] - F * R_MIN[i];
  }
  /* Compute hidden layer #1 */
  for (i=0;i<HIDDEN1;i++)
  {
    for (j=0;j<INPUTS;j++)
    {
       NEURON[HIDDEN1_INDEX + i] += NEURON[j]*W_In_H1[i][j];
    }
    NEURON[HIDDEN1_INDEX + i] += W_In_H1[i][INPUTS]; /* BIAS */
    NEURON[HIDDEN1_INDEX + i] = sigmoid( NEURON[HIDDEN1_INDEX + i] );
  }
  /* Compute network outputs */
  for (i=0;i<OUTPUTS;i++)
  {
    for (j=0;j<HIDDEN1;j++)
    {
       NEURON[OUTPUT_INDEX + i] += NEURON[HIDDEN1_INDEX + j]*W_H1_Out[i][j];
    }
    NEURON[OUTPUT_INDEX + i] += W_H1_Out[i][HIDDEN1]; /* BIAS */
    NEURON[OUTPUT_INDEX + i] = sigmoid( NEURON[OUTPUT_INDEX + i] );
```

```
}

/* Copy outputs to output array */

for (i=0;i<OUTPUTS;i++)

{

   F = (R_MAX[i+INPUTS]-R_MIN[i+INPUTS])/(S_MAX[i+INPUTS]-
S_MIN[i+INPUTS]); /* unscale factor */

   outputs[i] = F * NEURON[OUTPUT_INDEX + i] + R_MIN[i+INPUTS] - F *
S_MIN[i+INPUTS];

}

return 1;

}
```

# ÖZGEÇMİŞ

## KİŞİSEL BİLGİLER

**Adı, Soyadı:** Kürşat YILDIRIM

**Uyruğu:** Türkiye (TC)

**Doğum Tarihi ve Yeri:** [redacted]

**Medeni Durumu:** [redacted]

## EĞİTİM

| Derece | Kurum | Mezuniyet Tarihi |
|---|---|---|
| Lisans | EÜ Mekatronik Mühendisliği | 2017 |
| Lise | Ahmet Eren Anadolu Lisesi, Kayseri | 2012 |

## İŞDENEYİMİ

Teknik Satış Müdürü - Mazaka A.Ş. OSB Kayseri 2020 -halen

## YABANCI DİL

İngilizce