T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TEKNİKLERİYLE ROBOTLARDA HEDEF TAKİBİ

ÖZGÜR ALTINIŞIK

YÜKSEK LİSANS TEZİ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI
MEKATRONİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
DOÇ. DR. AYDIN YEŞİLDİREK

İSTANBUL, 2018

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME TEKNİKLERİYLE ROBOTLARDA HEDEF TAKİBİ

Özgür ALTINIŞIK tarafından hazırlanan tez çalışması 28.12.2018 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Aydın YEŞİLDİREK

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Aydın YEŞİLDİREK

Yıldız Teknik Üniversitesi _____

Prof. Dr. Ahmet KOYUN

Yıldız Teknik Üniversitesi _____

Prof. Dr. Murat DOĞRUEL

Marmara Üniversitesi _____

ÖNSÖZ

Öncelikle bugünlere gelmemde büyük emekleri olan değerli **aileme** her şey için çok teşekkür ederim.

Çalışmamda bana fikir veren, geçmişte yapmış olduğu kendi çalışmasından faydalanmama izin veren gerek akademik hayatta gerek akademik hayat haricinde işlerim yolunda gitmediği zaman da desteğini hiç esirgemeyen değerli Tez danışman hocam **Doç.Dr. Aydın Yeşildirek** hocama çok teşekkür ederim.

Robotumun mekanik dizaynında çok büyük pay sahibi olan, her türlü mekanik problemde meseleyi ustalıkla çözen, yeri geldiğinde benimle beraber gecelere kadar atölyede çalışmaktan çekinmeyen sevgili ark-adaşım **Mirvahid Ahmedipournaeim'e** çok teşekkür ederim.

Aralık, 2018

Özgür ALTINIŞIK

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vi

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

ÖZET ix

ABSTRACT xi

BÖLÜM 1

GİRİŞ 1

1.1 Tezin Amacı 1
1.2 Literatür Özeti 2
1.3 Hipotez 5

BÖLÜM 2

ALTYAPI 6

2.1 Robotun Modellenmesi 6
2.1.1 Kinematik Hesaplar ,Erişim Alanı ve Yörünge Planlama 6
2.1.2 Robotun Dinamik Modeli 10
2.2 Kontrol Metotları 12
2.2.1 Robotik Mekanizmalarda Kontrol Yöntemleri 12

BÖLÜM 3

TASARIM 20

3.1 Mekanik Tasarım 20
3.1.1 Robotun 3D çizimleri ve Görünüşü Ana Mekanik Parçalar 20
3.1.2 Yan Parçalar 24
3.2 Elektronik Dizayn 25
3.2.1 Eyleyiciler ve Sürücü Elektroniği 25
3.3 Kontrolcü Dizayn 27
3.3.1 Kontrolcü Mantığı ve Matematiksel Denklem 27

Hebbian Öğrenme Kuralı ve ReLU Aktivasyon Fonksiyonu 29
3.3.2 Saklı Katman Sayısı Değiştirme Algoritması 31
3.3.3 Kontrolcünün Matematiksel İspatı 32

BÖLÜM 4

BENZETİM 35

4.1 Benzetim Sonuçları 35

BÖLÜM 5

SONUÇ VE ÖNERİLER 46

KAYNAKLAR 48

EK-A

TEZ KAPSAMINDA KULLANILAN MATEMATİKSEL DEĞERLER 50

A-1 Robot Sabitleri 50
A-2 Robot Dinamik Matrisleri ve ReLU 51

ÖZGEÇMİŞ 53

SİMGE LİSTESİ

K_i	İntegral Kazanç
σ	Kayma Manifoldu
τ_g	Gravitasyonel Tork Değeri
e_q	Hata
q_d	İstenen açı
x_d	İstenen Konum
J	Jakobiyen Matrisi
f_c	Konum Durum Uzayında kuvvetler
α	Sabit Sayı
η	Sabit Sayı
τ	Tork Değeri
K_p	Oransal Kazanç
q	Radyan cinsinden açı
K_d	Türevsel Kazanç
C	Koriolis ve Merkezcil Kuvvet Matrisi
G	Yer çekimi Matrisi
g	yer çekimi sabiti
H	Genel Atalet matrisi
R	Rotasyon Matrisi

KISALTMA LİSTESİ

ANN	Yapay Sinir Ağları
CAD	Bilgisayar Destekli Tasarım
PID	Oransal Türevsel İntegral kontrol metodu
PWM	Sinyal Genişlik Modülasyonu
ReLU	Doğrultulmuş Doğrusal Birim Fonksiyon
SMC	Kayan Kipli Kontrol
Tanh	Tanjant Hiperbolik Fonksiyon
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 YSA sistemi açık iken ve kapalı iken benzetim sonuçları ... 3
Şekil 2.1 3D Çalışma Alanı ... 7
Şekil 2.2 Üstten Görünüm Çalışma Alanı ... 8
Şekil 3.1 Döner Alt tabla ... 21
Şekil 3.2 Birinci uzuv ... 21
Şekil 3.3 İkinci uzuv kıskaç mekanizması ... 22
Şekil 3.4 Robotun Yandan Görünüşü ... 22
Şekil 3.5 Robotun Tepeden Görünüşü ... 23
Şekil 3.6 Dizayn Edilen Robotun Tam Resmi ... 23
Şekil 3.7 Makeblock Robot Gripper ... 24
Şekil 3.8 AXK 2542 ... 24
Şekil 3.9 Rulman Yatakları ... 25
Şekil 3.10 PL3219 99:1 DC Motor ... 26
Şekil 3.11 PL3219 Teknik Özellikleri ... 26
Şekil 3.12 5 ara katmanlı için YSA Modeli ... 29
Şekil 4.1 1. Eklem için Hata ... 36
Şekil 4.2 2.Eklem için Hata ... 37
Şekil 4.3 3.Eklem için Hata ... 37
Şekil 4.4 Hatanın Normu 2π Periyotlu Ortalamaları ... 38
Şekil 4.5 YSA Toplam Katman Sayısı ... 38
Şekil 4.6 3D Konum Uzayı Yörünge Hedefimiz ... 39
Şekil 4.7 2D Konum Uzayı Yörünge Hedefimiz ... 40
Şekil 4.8 Eklem Açıları Hedef Yörüngemiz ... 40
Şekil 4.9 1.Eklem için Hata ... 41
Şekil 4.10 2.Eklem için Hata ... 41
Şekil 4.11 3.Eklem için Hata ... 42
Şekil 4.12 Eklem Açıları için Hataların Ortalama Normu ... 42
Şekil 4.13 X Koordinat Hata ... 43
Şekil 4.14 Y Koordinat Hata ... 43
Şekil 4.15 Z Koordinat Hata ... 44
Şekil 4.16 Konum Uzayındaki Hatanın Ortalama Normu ... 44
Şekil 4.17 Katman Sayısı ... 45
Şekil A.1 ReLU Fonksiyon Grafiği ... 52

ÖZET

DERİN ÖĞRENME TEKNİKLERİYLE ROBOTLARDA HEDEF TAKİBİ

Özgür ALTINIŞIK

Mekatronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Aydın YEŞİLDİREK

1970'lerden bugüne robotik mekanizmalar gerek endüstriyel alanda gerek de başka alanlarda kullanılmaktadır. Robotik mekanizmalar hız, ucuzluk, tehlikeli şartlar altında çalışabilme, verimlilik, keskinlik, kuvvet gibi pek çok kazanımı beraberinde getirmektedir. Fakat bu kazanımlardan etkili bir biçimde yararlanabilmek için bu robotik mekanizmaların konum-hız-ivme ve tork-kuvvet kontrollerinin hassas bir biçimde yapılmış olması gerekmektedir. Bu sorunu çözebilmek için zaman içerisinde çeşitli kontrol metotları önerilmiştir.

Tez çalışmamda bu kontrolcülerden ve metotlardan genel olarak bahsedilmiştir ve kontrolcülerin avantaj-dezavantajları bizim kontrolcümüzün getirdikleri anlatılmıştır. Geçmişte önerilmiş Yapay Sinir Ağı destekli Danışmansız öğrenme metotlu kontrolcüyle birleşmiş PD kontrolcü günümüz imkanlarıyla daha iyi geliştirilerek ve **derin öğrenme metotları** da eklenerek yeni ve daha efektif bir kontrolcü fikri ortaya atılmış ve bu hipotezin matematiksel ispatı yapılmıştır.

Buna ek olarak Tasarlanan kontrolcüyü test etmek amacıyla bir R-R-R tipi robot modeli, tasarlanmıştır. Bu model üzerinden hem benzetim programlarıyla testler yapılmış ve kontrolcü modelimiz denenmiştir. Bunun yanında kontrolcümüz diğer kontrolcü tipleri olan PD, Hesaplanmış Tork Metodu ve Kayan Kipli Kontrol Metodu ile de kıyaslanmıştır ve sonuçlar verilmiştir.

Benzetimden alınan sonuçlar Derin Öğrenme'nin danışmansız öğrenme ile robotlarda kullanımının olumlu sonuçlar verdiğini göstermiştir.

Anahtar Kelimeler: Yapay sinir ağları, derin öğrenme, Hebbian öğrenme, gizli katman sayısı ayarlama algoritması, robot kontrol metotları

ABSTRACT

TRAJECTORY TRACKING OF ROBOTS WITH DEEP LEARNING METHODS

Özgür ALTINIŞIK

Department of Mechatronics Engineering

MSc. Thesis

Adviser: Assoc. Prof. Dr Aydın YEŞİLDİREK

Since 1970s, robotic mechanisms have been used both in industrial and other fields. Robotic mechanisms bring many advantages such as speed, cheapness, working under dangerous conditions, efficiency, sharpness and power. However, in order to benefit from these advantages effectively, the position-velocity-acceleration and torque-force controls of these robotic mechanisms must be precisely carried out. Various control methods have been proposed over time to solve this problem.

In my thesis study, the advantages and disadvantages of these controllers and controllers are mentioned briefly and the assets what brought by our controller explained detailed. The unsupervised learning NN controller combined with PD which have proposed in the past is enhanced by nowadays feasibility and **deep learning** methods and this new and more efficient controller idea has been proposed and this hypothesis have mathematically proven.

In addition, an R-R-R robot model has been designed from the beginning which includes mechanical and electronic components, to test the proposed controller. Our controller models have been tested on this model both in simulation. Also, our controller algorithm was compared with the other control methods such as PD, Computed Torque Method and Sliding Mode Control Method, and the results were given.

Simulation and the results have shown that the using deep learning in robots with non-supervising learning improve positive controller's performance.

Keywords: Neural networks, Hebbian learning, deep learning, hidden layer size tuning algorithm, robot control methods

BÖLÜM 1

GİRİŞ

1.1 Tezin Amacı

Robotik Sistemler, özellikle 1970'lerdeki Endüstri 3.0 devriminden sonra oldukça yaygınlaşmaya başlamıştır. Robotik mekanizmaların insanlara karşı olan avantajları bunda en önemli etmen olmuştur. Bu avantajlardan bazıları, robotların fiziksel yorulmasının daha düşük olması, uygulayabildiği kuvvetlerin daha yüksek olması, çalışma hızlarının ve sonuçlarda keskinliğinin daha yüksek olması, işçi sayısını azaltmak ve düşük maliyet, tehlikeli ortamlarda çalıştırılabilmeleridir. Bu ve benzeri avantajları sebebiyle robotlar özellikle endüstride yaygın olarak kullanılmaya başlanmıştır.

Başlangıçta üretilen robotlar teknolojik yetersizlik sebebiyle genel olarak hidrolik eyleyicili olup gerek elektrikli eyleyici teknolojisinin zayıf olması gerek daha kontrol metotlarının tam gelişmemesi yüzünden çok hassas işlerde kullanılamamıştır. Fakat zaman içerisinde bu hassasiyet metodunu çözmek için çeşitli farklı eyleyiciler ve kontrol metotları geliştirilmiş olup hassas hareket ve kuvvet kontrolü üzerine çeşitli metotlar ortaya atılmıştır.

Bizim de bu tezde amacımız robotik mekanizmalar için matematik model hesabı gerektirmeyen bir kontrolcü tasarlamaktı. Doğrusal metotlarla çalışan veya doğrusal olmayan dinamikleri önceden matematik hesaplamalarıyla sisteme gönderen kontrolcüler, gerek yüksek serbesiyet dereceli modellerde matematik hesapların zor olması gerek sürtünme gibi bazı dış etmenler sebebiyle hassasiyet kaybı yaşamaktadır.

Bunu sorunu aşmak ve doğrusal olmayan dinamikleri tam olarak yok edebilmek adına YSA metodunun daha başarılı olacağını düşündük. Daha önceden 1996 yılında

önerilmiş danışmansız içeren YSA kontrol metodunu daha da geliştirip derin öğrenmeyi kullanarak hata çıktısını azaltmak ve sistemin bozucu etkilere karşı direncini artırmayı amaçladık.

Daha önce önerilen kontrol metodunun dezavantajlarını (aktivasyon fonksiyon tipi, öğrenme metodu, ara katman sayısı, işlem çokluğu) geliştirerek sistemin günümüzde gelişmiş mikro denetleyicilerle gerçek dünyaya entegre edilebilir hale gelmesini de amaçladık.

Kontrol metodunu prototip gerçek bir RRR robot üzerinde farklı hedef yörüngelerde deneyerek sistemin gerçek dünyada uygulanabilir olduğunu ve gerek klasik PD ve Hesaplanmış Tork metoduyla kıyaslama yaparak avantaj ve dezavantajlarını hem simülasyonda hem de gerçek model üzerinde göstermek istedik. Aynı zamanda model üzerinde daha önce benzetimi yapılmamış konum uzayında hareket gibi veya sinüzoidal hareket gibi zorlayıcı yörünge takibinde derin öğrenmeyle birleştirilmiş algoritmamızı test ettik.

1.2 Literatür Özeti

YSA öğrenmenin kontrolcü tasarımı ve kontrol teorisinde kullanımına literatürde, 1992 yılında "A Neural Network Compensator for Uncertainties of Robotics Manipulators" isimli IEEE'de yayımlanan makalede rastlayabiliriz[1]. Bu makalede yazarlar öğrenilmesi gereken şey basitleştikçe lineer olmayan YSA öğrenmesinin daha başarılı performans sergileyeceği düşüncesi ile 2 katmanlı basit bir YSA modelini gözlemleyici olarak kullanmışlardır. Öğrenme yöntemi olarak ise danışmanlı öğrenmeyi kullanmışlardır.

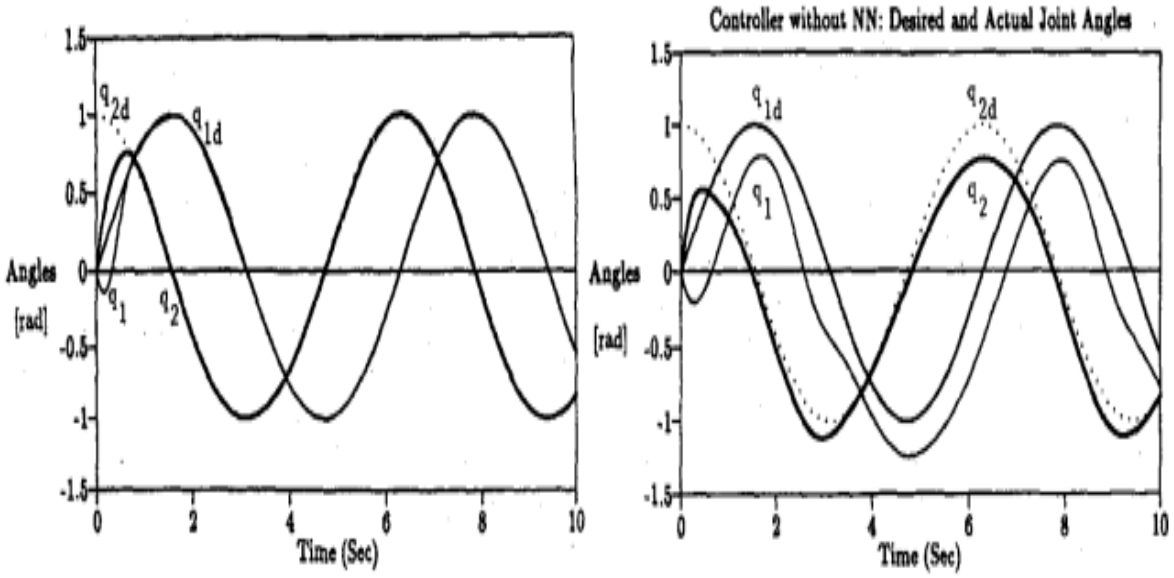
Sistemin kontrolcü tasarımında Hesaplanmış Tork Metodu kullanılmış olup, modelleme hataları için ise yapay sinir ağları kullanmayı önermişlerdir. Ağ metodu olarak ileri beslemeli tek gizli katmanlı sigmoid aktivasyon fonksiyonlu bir ağ ve öğrenme metodu olarak ise hatanın geriye yayılım metodunu kullanmışlardır.

Kontrolcü sistemi 2 serbesiyet dereceli R-R robot için simülasyon programı vasıtasıyla denenmiştir. Sistem ilk ötelemede iyi sonuç vermese de ileri ki seferlerde hata azalmıştır.

Bu sistem sadece simülasyonda denenmiş olup küçük bir ağ kullanılması ve önceden öğrenme süreci gerektirmesi dezavantajlarıdır.

YSA ile danışmansız öğrenme metodu ile robot kontrol etme fikri ise “Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance” isimli makalede 1996 yılında bahsedilmiştir. Bu makalede klasik bir PD manifoldu, YSA ‘dan alınan çıktı ve gürbüzleştirici sinyal robota tork girdisi olarak gönderilerek robotun gürbüz kontrol edilmesinin sağlanmasından bahsedilmiş ve kontrolcü metodu matematiksel olarak Lyapunov metodu ile ispatlanmıştır.[2]

YSA öğrenme metodu olarak sonuçların gradiyeni ve eski değerin çeşitli katsayılarla toplanmasıyla elde edilen sürekli diferansiyel denklem olarak tanımlanmıştır. YSA giriş verisi olarak sistemin arzulanan hareket verilerini ve tanımlanan hatanın değerlerini alırken çıkış olarak ise tork değeri çıkartır. Sistem simülasyon programı yardımıyla iki serbesiyet dereceli R-R robot üzerinde benzetim yapılmış ve sonuçlar aşağıdaki gibi olmuştur.



Şekil 1.1 YSA sistemi açık iken ve kapalı iken benzetim sonuçları

Danışmansız YSA sonuçları iyileştirmiştir ve gürbüzlüğü kanıtlanmıştır. Sadece 2 katman kullanılması , 2 serbesiyet dereceli bir robotta denenmiş olması ve benzetim aşamasından ileriye geçilmemiş olması makalenin eksikleridir.

1999 yılında yayınlanan “Neural Network Output Feedback Control of Robot Manipulators” isimli makalede danışmansız öğrenme yoluyla gözlemleyici tasarlayarak sistem kontrolcüsü dizayn edilmiş , 2 serbesiyet dereceli bir robotta denenmiş olması ve benzetim aşamasından ileriye geçilmemiştir.

1998 yılında Frank L. Lewis , Aydın Yeşildirek, Jagannathan Sarangapani tarafından yayımlanan “Neural Network Control of Robot Manipulators and Nonlinear Systems” adlı kitapta 1996 yılında yayınladıkları çalışmalara ek olarak farklı danışmansız öğrenme tekniklerinden daha detaylı bahsedilmiş olup matematiksel ispatlarına da yer verilmiştir fakat yine tüm çalışmalar 2 serbesiyet dereceli bir robotta denenmiş olması ve benzetim aşamasından ileriye geçmemiştir.[3]

2001 yılında yayımlanan “Neural Network-based Adaptive Controller Design of Robotic Manipulators with an Observer” isimli makalede iki yerine çok katmanlı YSA ve sürekli diferansiyel denklem üzerinden danışmansız bir öğrenme algoritması tasarlanmış olup gözlemleyici üzerinden başarılı sonuçlar elde edilmiştir.[4]

2008 yılında “Adaptive Control of Nonlinear PID-based Analog Neural Networks for a Nonholonomic Mobile Robot” ismiyle yayımlanmış makalede ve “Neural Network Controller for Mobile Robot Motion Control” isimli makalede danışmansız YSA mobil robotlar üzerinde kullanılmıştır[5], [6]. Jun Ye tarafında yazılan “Adaptive Control of Nonlinear PID-based Analog Neural Networks for a Nonholonomic Mobile Robot” isimli makalede PID katsayılarını belirlemek için danışmansız geriye yayılım metotlu öğrenme kullanılmışken “Neural Network Controller for Mobile Robot Motion Control” isimli makalede bir önceki basamaktan öğrenen farklı bir geriye yayılım algoritması vardır. Her iki YSA da sistemin hata oranın düşürmede başarılı olmuştur.

2007 yılında yazılmış “A Neural Network Controller for Continuum Robots” isimli makalede 1996 yılında Frank Lewis ve Aydın Yeşildirek tarafından önerilen kontrolcü danışmansız YSA kontrolcü tipi ileri beslemeli kontrolcü ile harmanlanarak denenmiştir. Aşırı gereğinden fazla serbesiyet dereceli robotların modelinin yüksek seviyede doğrusal olmayan öğeler içermesi gereği bu tarz metot izlemişler ve sistemlerini OCTARM isimli robotlarında gerçeklemişlerdir.[7]Kontrolcünün performansı sistem üzerinde gerçek dünya da olumludur.

1.3 Hipotez

Yeni önerdiğimiz kontrolcü sistemimiz sayesinde daha az hata ile robotun dinamiklerini kontrol edebildiğini hipotez olarak bu tezde ortaya sürüyoruz. Yeni geliştirilen metotlar ile bu kontrolcümüz geçmişte olan dezavantajlarının daha kabul edile bilinir seviyeye çekilebileceğini bu tezde ispatlayacağız.

BÖLÜM 2

ALTYAPI

2.1 Robotun Modellenmesi

2.1.1 Kinematik Hesaplar ,Erişim Alanı ve Yörünge Planlama

Sistemimizin hareket planlamasını yapabilmek ve dinamik modelini çıkarabilmek için öncelikle ileri kinematik modelinin çıkarılması gerekmektedir. Bir önceki bölümde verilen CAD çizimlerinden yola çıkarak rotasyon ve transformasyon matrisleri yardımıyla sistemimizin ileri kinematiği aşağıdaki denklemlerde verilecektir.

Eklemlerimizin hepsi döner mafsal olduğu için koordinat sistemlerimiz arasındaki rotasyon matrisleri 0 baz koordinat olmak üzere aşağıdaki gibi olacaktır.

$$R_0^1 = R_{y(q_1)} = \begin{bmatrix} \cos(q_1) & 0 & \sin(q_1) \\ 0 & 1 & 0 \\ -\sin(q_1) & 0 & \cos(q_1) \end{bmatrix} \tag{2.1}$$

$$R_1^2 = R_{z(q_2)} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 \\ \sin(q_2) & \cos(q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$R_2^3 = R_{z(q_3)} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 \\ \sin(q_3) & \cos(q_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$A_0^1 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) & 0 \\ 0 & 1 & 0 & l_1 \\ -\sin(q_1) & 0 & \cos(q_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

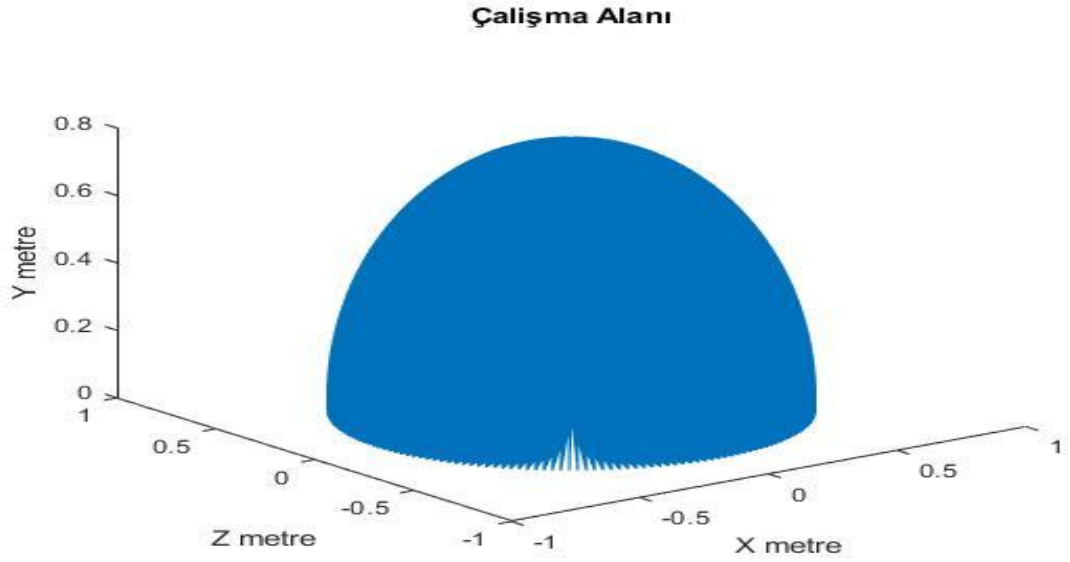
$$A_1^2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 \\ -\sin(q_2) & cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2\cos(q_2) \\ -\sin(q_2) & cos(q_2) & 0 & l_2\sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

$$A_2^3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & 0 \\ -\sin(q_3) & cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & l_3\cos(q_3) \\ -\sin(q_3) & cos(q_3) & 0 & l_3\sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

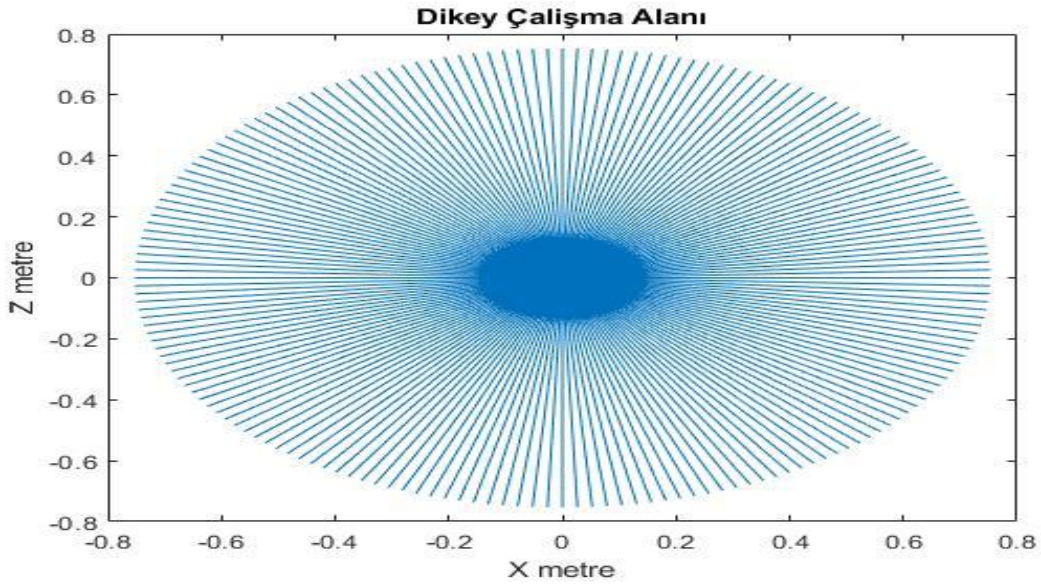
Koordinat sistemlerinin merkeze göre konumları ise aşağıdaki gibi olacaktır

$$P_0^1 = \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix}, P_0^2 = \begin{bmatrix} l_2\cos(q_1)\cos(q_2) \\ l_1 + l_2\sin(q_2) \\ -l_2\sin(q_1)\cos(q_2) \end{bmatrix}, P_0^3 = \begin{bmatrix} \cos(q_1)(l_2\cos(q_2) + l_3\cos(q_2+q_3)) \\ l_1 + l_2\sin(q_2) + l_3\sin(q_2+q_3) \\ -\sin(q_1)(l_2\cos(q_2) + l_3\cos(q_2+q_3)) \end{bmatrix} \tag{2.7}$$

Sistemimiz çalışma alanı ise fiziksel sınırlar da dikkate alınarak aşağıdaki görsellerde verilecektir.



Şekil 2.1 3D Çalışma Alanı



Şekil 2.2 Üstten Görünüm Çalışma Alanı

Sistemimizi konum durum uzayına göre de kullanacağımız için konum durum uzayındaki bir pozisyona ulaşmak için gerekli eklem açılarını bulmak amacıyla ters kinematik hesapları da gereklidir.

3 tane serbesiyet derecemiz olduğu için sistemimiz farklı çözümlere sahip olabilecektir.

$$P=\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow q_1 = a\tan2(-z,x) \tag{2.8}$$

$$a=\left(\frac{x}{\cos(q_1)}\right)^2+(y-l_1)^2, k=\frac{a-{l_2}^2-{l_3}^2}{2l_2l_3} \tag{2.9}$$

$$\left((\tan\left(\frac{q_3}{2}\right)\right)^2(k+1)+(k-1)=0 \Rightarrow \tan(q_3)=\mp\left(2\sqrt{\frac{1-k}{1+k}}\right), q_3=a\tan2\left(\mp2\sqrt{1-k},\sqrt{1+k}\right) \tag{2.10}$$

$$c=l_2+l_3\cos(q_3), p=l_3\sin(q_3), U=y-l_1 \tag{2.11}$$

$$\left(\tan\left(\frac{q_2}{2}\right)\right)^2(-c-U)+2\tan\left(\frac{q_2}{2}\right)p+c-U=0 \Rightarrow \tan(q_2)=\frac{2p\pm2\sqrt{n^2-U^2+p^2}}{c+U} \tag{2.12}$$

$$q_2=a\tan2(2p\pm2\sqrt{n^2-U^2+p^2},c+U) \tag{2.13}$$

Robotumuzun konum uzayında hedef yörünge takibi yapabilmesi için sadece ters kinematik yeterli olmayabilir. Gradiyent bazlı jakobiyen metodu daha stabil sonuçlar verecektir fakat jakobiyen matrisi determinantı sıfır olduğu yani tekillik durumunda bu metot da problem çıkarabilir. Bu sorunu da aşmak için sönümlü en küçük kareler metodunu kullandık.

Jakobiyen matrisi aşağıdaki denkleme göre hesaplanır.

$$J=\begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \\ \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \end{bmatrix} \tag{2.14}$$

Bizim jakobiyen matrislerimiz ise

$$J_v^1=\begin{bmatrix}0\\0\\0\end{bmatrix}, J_v^2=\begin{bmatrix} -l_{2c}\sin(q_1)\cos(q_2) & -l_{2c}\cos(q_1)\sin(q_2) & 0 \\ 0 & -l_{2c}\cos(q_2) & 0 \\ -l_{2c}\cos(q_1)\cos(q_2) & -l_{2c}\sin(q_1)\sin(q_2) & 0 \end{bmatrix} \tag{2.15}$$

$$J_v^3=\begin{bmatrix} -\sin(q_1)(l_2\cos(q_2)+l_{3c}\cos(q_2+q_3)) & -\cos(q_1)((l_2\sin(q_2)+l_{3c}\sin(q_2+q_3))) & -l_{3c}\cos(q_1)\sin(q_2+q_3) \\ 0 & (l_2\cos(q_2)+l_{3c}\cos(q_2+q_3)) & l_{3c}\cos(q_2+q_3) \\ -\cos(q_1)(l_2\cos(q_2)+l_{3c}\cos(q_2+q_3)) & \sin(q_1)((l_2\sin(q_2)+l_{3c}\sin(q_2+q_3))) & l_{3c}\sin(q_1)\sin(q_2+q_3) \end{bmatrix} \tag{2.16}$$

$$J_\omega^1=\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix}, J_\omega^2=\begin{bmatrix}0&\sin(q_1)&0\\1&0&0\\0&\cos(q_1)&0\end{bmatrix}, J_\omega^3=\begin{bmatrix}0&\sin(q_1)&\sin(q_1)\\1&0&0\\0&\cos(q_1)&\cos(q_1)\end{bmatrix} \tag{2.17}$$

En küçük sönümlü kareler metodu ise [8]

$$1>\varepsilon>0, \dot{q}=\left(\left(J_v^3\right)^T J_v^3+\varepsilon^2 I\right)^{-1}\left(J_v^3\right)^T \dot{x} \tag{2.18}$$

$$\ddot{q}=\left(\left(\left(J_v^3\right)^T J_v^3+\varepsilon^2 I\right)^{-1}\left(J_v^3\right)^T\right)\left(\ddot{x}-\frac{dJ_v^3}{dt}\dot{q}\right) \tag{2.19}$$

$$\left(\left(J_v^3\right)^T J_v^3\right)=\begin{bmatrix} R_{11} & 0 & 0 \\ 0 & R_{22} & R_{23} \\ 0 & R_{23} & R_{33} \end{bmatrix} \tag{2.20}$$

$$\left(\left(J_v^3\right)^T J_v^3+\varepsilon^2 I\right)=\begin{bmatrix} R_{11}+\varepsilon^2 & 0 & 0 \\ 0 & R_{22}+\varepsilon^2 & R_{23} \\ 0 & R_{23} & R_{33}+\varepsilon^2 \end{bmatrix} \tag{2.21}$$

Jakobiyen matrisimizin determinantı 3. Açımız 0 olduğunda sıfır olacaktır fakat sönümlü en küçük kareler metodunda jakobiyen matrisimiz tekil olsa bile determinantımız $\eta^4\left(\eta^2+\left(l_2+l_3\right)^2+l_3^2\right)$ olacaktır ve tekillik problemimiz ortadan kalkacaktır.

2.1.2 Robotun Dinamik Modeli

Robotumuzun dinamik modelini Lagrange denklemleri vasıtasıyla ve robotik metotlarıyla elde ettik. *L* Lagrangian değeri ve O sisteme sağlanan tork veya kuvvet olmak üzere sistemimizde sönümleme olmadığı varsayıldığı durum için genel Lagrange Denklemi aşağıda verilmiştir.

$$L=K_e-P_e \Rightarrow \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}}\right)-\left(\frac{\partial L}{\partial q}\right)=O \tag{2.22}$$

Lagrange denklemleri için gereken robotumuzdaki eklemlerin ağırlık merkezlerindeki doğrusal-açısal hız, dikey konum değerleri aşağıda verilecektir.

$$v_1=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, v_2=\begin{bmatrix} -l_{2c}(\sin(q_1)\cos(q_2)\dot{q}_1+\cos(q_1)\sin(q_2)\dot{q}_2) \\ l_{2c}\cos(q_2)\dot{q}_2 \\ l_{2c}(-cos(q_1)\cos(q_2)\dot{q}_1+\sin(q_1)\sin(q_2)\dot{q}_2) \end{bmatrix} \tag{2.23}$$

$$v_3=\begin{bmatrix} -(l_2+l_{3c})(\sin(q_1)\cos(q_2)\dot{q}_1+\cos(q_1)\sin(q_2)\dot{q}_2) \\ (l_2+l_{3c})\cos(q_2)\dot{q}_2 \\ -(l_2+l_{3c})(cos(q_1)\cos(q_2)\dot{q}_1-\sin(q_1)\sin(q_2)\dot{q}_2) \end{bmatrix} \tag{2.24}$$

Açısal hız hesaplamasında iz düşüm metodu ve çarpık simetrik matris metodunu kullandık.

$$\dot{R}R^T = S \Rightarrow S = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.25}$$

$$\omega_1 = \begin{bmatrix} \dot{R}_0^1 R_1^0(3,2) \\ \dot{R}_0^1 R_1^0(1,3) \\ \dot{R}_0^1 R_1^0(2,1) \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{q}_1 \\ 0 \end{bmatrix}, \omega_2 = \begin{bmatrix} \dot{R}_0^2 R_2^0(3,2) \\ \dot{R}_0^2 R_2^0(1,3) \\ \dot{R}_0^2 R_2^0(2,1) \end{bmatrix} = \begin{bmatrix} \sin(q_1)\dot{q}_2 \\ \dot{q}_1 \\ \cos(q_1)\dot{q}_2 \end{bmatrix} \tag{2.26}$$

$$\omega_3 = \begin{bmatrix} \dot{R}_0^3 R_3^0(3,2) \\ \dot{R}_0^3 R_3^0(1,3) \\ \dot{R}_0^3 R_3^0(2,1) \end{bmatrix} = \begin{bmatrix} \sin(q_1)(\dot{q}_2 + \dot{q}_3) \\ \dot{q}_1 \\ \cos(q_1)(\dot{q}_2 + \dot{q}_3) \end{bmatrix} \tag{2.27}$$

Lagrange metodunun robotik sistemlerde kolaylık sağlayan yaklaşımlarını kullanmak için doğrusal ve açısal hız için Jakobiyen matrislerine de ihtiyacımız vardır.

Bir robotun matematik denklemi daha önce de belirtiğimiz gibi

$$H(q)\ddot{q} + C(\dot{q},q)\dot{q} + G(q) + J^T F = \tau \tag{2.28}$$

Şeklinde ifade edilebilir ve robotik metotların bize sağladığı kolaylıklar ile H, C ve G matrislerini klasik Lagrange metodundan daha hızlı hesaplayabiliriz.

Robotik lagrange yaklaşımına göre

$$H(q) = m_1 (J_v^1)^T J_v^1 + m_2 (J_v^2)^T J_v^2 + m_3 (J_v^3)^T J_v^3 + (J_\omega^1)^T I_1 J_\omega^1 + (J_\omega^2)^T I_2 J_\omega^2 + (J_\omega^3)^T I_3 J_\omega^3 \tag{2.29}$$

Formülünü kullanırsak H atalet matrisi aşağıdaki şekilde bulunabilir;

$$H(q) = \begin{bmatrix} H_{11} & 0 & 0 \\ 0 & H_{22} & H_{23} \\ 0 & H_{23} & H_{33} \end{bmatrix} \tag{2.30}$$

C merkezcil-coriolis matrisini de Christoffel sembolleri yardımıyla kolaylıkla bulabiliriz.[9]

$$i,j \in Z \rightarrow i,j = [1,3], C_{ij}(\dot{q},q) = \frac{1}{2}\left(\sum_{k=1}^{n} \frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{kj}}{\partial q_i} \right) \dot{q}_k = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{22} \\ C_{31} & C_{32} & C_{32} \end{bmatrix} \tag{2.31}$$

G matrisi de aşağıdaki metot ile belirlene bilinir.

$$G = -\left(J_v^1 \begin{bmatrix} 0 \\ -m_1 g \\ 0 \end{bmatrix} + J_v^2 \begin{bmatrix} 0 \\ -m_2 g \\ 0 \end{bmatrix} + J_v^3 \begin{bmatrix} 0 \\ -m_3 g \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ G_{21} \\ G_{31} \end{bmatrix} \tag{2.32}$$

Uç eklentinin yüzeyle etkileşiminden meydana gelen kuvvetleri içinse

$$J = \begin{bmatrix} J_v^3 \\ J_w^3 \end{bmatrix}, F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{2.33}$$

Şeklinde matematiksel modelleme yapılır. Matrislerin ayrıntılı değerleri Ek-A bölümünde paylaşılacaktır.

2.2 Kontrol Metotları

2.2.1 Robotik Mekanizmalarda Kontrol Yöntemleri

Bu metotları basitçe **Eklem durum uzayı , Konum Durum Uzayı ve Eklemlerin birbirinden bağımsız kontrol** edildiği kontrolcüler olarak gruplandırabiliriz.

Özel durumlar haricinde, robotlarda hareket planlanırken, hedeflenen yörünge konum durum uzayında planlanır . **Eklem uzayındaki kontrolcülerde** eklem açıları hareket planına bağlı olarak kontrol edilir. Öncelikle verilen yörünge çeşitli metodlar (ters - kinematik veya jakobiyen) vasıtasıyla konum uzayından eklem durum uzayı çevirilir ve bunun üzerinden aşağıdaki genel şematik gibi bir kontrolcü tasarlanır.

Bu tip kontrolcüde amaç istenen konumu eklem açısı cinsinden ters kinematik metoduya yeniden yazıp istenilen yörüngenin takip edilebilmesini sağlamaktır.

Diğer tip genel metod ise **Konum Durum Uzayında Kontrolcü** tasarlamaktır. Bu tip kontrolcüler genel olarak daha az belirgin veya robot uç noktasının hedefinin online olarak değişebildiği veya daha zor (doğrusal,eğrisel) yörüngelerin olduğu durumlarda kullanılırlar. Robotun denklemi eklem uzayından çeşitli dönüşümler yapılarak yeniden

oluşturulur ve kontrolcü bu yeni denklem üzerinden uç noktasının pozisyonuna göre çıktı sinyali verir.

Tüm endüstriyel robotların genel denklemi aşağıdaki gibi yazılabilir.

$$\tau = H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g(q) \tag{2.34}$$

Jakobiyen matrisi yardımıyla konum ve eklem uzayı arasında hız(birinci türev) seviyesinde geçişlerde yapılabilir.

$$\dot{x} = J(q)\dot{q} \tag{2.35}$$

Robot modelimizi konum uzayı cinsinden 1.1 nolu denkleme benzetmek istediğimiz için bir başka f_c kuvvet-tork değerini aşağıdaki denklemdeki bir değer tanımlamamız gerekir.[10], [11]

$$f_c = M(q)\ddot{x} + V(q,\dot{q})\dot{x} + G(q) \tag{2.36}$$

Bu yeni paratmeleri de aşağıdaki denklemlerle elde edebiliriz.

$$M(q) = J^{-T}H(q)J^{-1} \tag{2.37}$$

$$V(\dot{q},q) = J^{-T}C(\dot{q},q)J^{-1} - M(\ddot{q})\dot{J}J^{-1} \tag{2.38}$$

$$G(q) = J^{-T}\tau_g(q) \tag{2.39}$$

Bu metodun en önemli avantajı algoritmanın hız düzeyinde ileri kinetmatik kullanmasından dolayı , ters kinematik hesabına gereksinimi olmamasıdır. Bu sayede daha hasas yörünge (çizgisel ,dairesel) daha düzgün takip edilebilir.

Diğer bir metot ise **tüm eklemlerin bağımsız kontrol** edilmesidir. Bu metot , genel robot tork denklemi kontrolcude hesaplanmadan sadece eyleyicinin içsel döngüyle istenilen pozisyona göre kontrol edilmesiyle gerçeklenir. Gereken açıların hesaplanıp kontrolcuye gonderilmesi de ayrı bir işlem olarak gerçekleşir ve eyleyiciyi kontrol eden sistem sadece eyleyicinin açısını kontrol etme ile ilgilenir. Genel olarak eyleyicilerde PID kullanılsa da bazen doğrusal olmayan metodlarda kullanılabilir. Bu metodun avantajı basit olması , düşük hesaplama ve iletişim gereksinimi olarak sayılabilir.

PID metodu genel olarak basit bir matematik hesap olup kontrol sistemlerinde, çok fazla kaynak gerektirmeyen yapısı sebebiyle çokça tercih edilmektedir. Klasik bir PID tablosu bağımsız eklem kontrol için denklem 1.5'deki gibi ifade edilebilir.

$$\tau = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t) d(t) \tag{2.40}$$

PID kontrolcüler doğrusal modeller için dizayn edilmişlerdir ve doğrusala yakın kabul edilebilecek modellerde iyi performans gösterirler fakat model doğrusal değilse sistemin kararlığı ve performansı her şartta garanti edilemeyebilir. Robot modeli doğrusal olmadığı için ve kaotik bir diferansiyel denklemi olduğu için bir koşulda işe yarayan parametreler diğer koşullarda veya geçiş sürecinde aynı sonucu vermeyebilir. Bu durumlar için kazanç oranlarını, bulanık mantık gibi, adaptif ayarlayan metotlar gerekir veya modelin doğru sallaştırılması gerekir. Bu sebeple eğer eklemler birbirinden bağımsız kontrol edilmeyecekse robotik mekanizma yaklaşık tork denkleminin çıkarılması ve çeşitli geri besleme kontrolü mantığına dayanan metotlarla yapılandırılması veya başka doğrusal olmayan metotlara geçiş gerekmektedir.

Bu bağlamda **pasiflik bazlı ve hesaplanmış tork** metotları; geri besleme doğru sallaştırması yaparak daha stabil bir kontrol sağlayabilir kullanılabilir.

Her iki kontrolcü de bir başka yardımcı kontrol girdisi oluşturularak bu girdi yardımıyla sistemin hata çıktısının sabit evrede olarak yok edilmesi amaçlanır. **Pasiflik bazlı kontrolde** oluşturulan girdisi ve sisteme verilen tork denklem 1.8'deki gibi olur. [10]

$$\dot{q}_r = \dot{q}_d + \alpha e_q, \alpha > 0 \tag{2.41}$$

$$\tau = H(q)\ddot{q}_r + C(\dot{q},q)\dot{q}_r + \tau_g(q) + K_v \dot{e}_q + K_p e_q \tag{2.42}$$

Bu kontrol girdisinin sonucu olarak ise denklem 1.9'daki genel sonuç edilir.

$$s_q = \dot{e}_q + \alpha e_q \tag{2.43}$$

$$H(q)\dot{s}_q + C(\dot{q},q)s_q + K_v \dot{e}_q + K_p e_q = 0 \tag{2.44}$$

Pasiflik bazlı kontrolde sistemin doğrusal olmayan özelliği tamamen değiştirilmediği için daha gürbüz bir kontrolcüdür. Sistem Lyapunov teoremine göre kararlıdır.

Hesaplanmış Tork metodunda ise yardımcı kontrol girdisi yardımıyla, geri besleme metoduyla doğru sallaştırılmış sistemin hata çıktısının sıfıra gitmesi amaçlanır. Sistemin tork denklemi 1.10'daki gibi olacaktır.

$$\ddot{q} = \nu, K_p > 0, K_v > 0 \tag{2.45}$$

$$\nu = \ddot{q}_d + K_v \dot{e}_q + K_p e_q \tag{2.46}$$

$$\tau = H(q)(\ddot{q}_d + K_v \dot{e}_q + K_p e_q) + C(\dot{q}, q)\dot{q} + \tau_g(q) \tag{2.47}$$

Bu kontrol girdisinin sonucu olarak ise hata denklemi aşağıdaki denklemi takip edecektir.[10]

$$\ddot{e}_q + K_v \dot{e}_q + K_p e_q = 0 \tag{2.48}$$

$$\Delta = \sqrt{K_v{}^2 - 4K_p}, c_1 = \frac{\dot{e}_q(0)}{\Delta} + \frac{e_q(0)(\frac{K_v}{\Delta} + 1)}{2}, c_2 = -\frac{\dot{e}_q(0)}{\Delta} - \frac{e_q(0)(\frac{K_v}{\Delta} - 1)}{2} \tag{2.49}$$

$$e_q(t) = c_1 e^{\frac{-t(K_v - \Delta)}{2}} + c_2 e^{\frac{-t(K_v + \Delta)}{2}} \tag{2.50}$$

Bu denklem sonucunda hata stabil aşamada sıfıra yakınsar ve $\sqrt{K_v^2 - 4K_p}$ eğer sıfırdan küçük olursa gelecek negatif kutuplar sebebiyle sistemde salınımlar olacaktır. Bu sebeple $K_v \approx 2\sqrt{K_p}$ ve şartını sağlayacak şekilde parametre seçimi yapmak sonuç açısından daha faydalı olacaktır.

Hesaplanmış Tork metodu gibi geri besleme doğru sallaştırması yapılan metotların ana sıkıntısı modelin sıfır dinamiklerinin çıkarabileceği problemlerdir. Buna ek olarak hem pasiflik bazlı kontrolde hem de hesaplanmış tork bazlı kontrol uygulamalarında robotun matematiksel modeli kullanılmakta olup hesaplamada yapılabilecek hatalar sistemde büyük hatalara sebebiyet verebilirler. Bunu durumu önlemek amaçlı **Adaptif Kontrol, H ∞** gibi metotlarda önerilmiştir. Bu metotlar hataları minimalize etse de çok fazla hesaplama gereksinimi içerebilirler. Aynı zamanda bu metotlarda modelin dinamik modelinin hesaplanmasını gerektirmektedir.

Sistemin matematik modelinden kaynaklı hesap hatalarından kaçınabilmek adına doğrusal olmayan kontrol metotları da kullanılmaktadır. Bu metotlardan bazıları **Kayan Kipli Kontrol (Değişken Yapılı Kontrol), Yapay Sinir Ağları** olarak sayılabilir.

Bu metotlardan Kayan Kipli Kontrol metodu matematiksel olarak oldukça basit birtakım hesaplamalara dayanmakta olup değişken yapılı bir kontrol metodudur, içerisinde sürekli ve kesikli kontrolcüleri barındırır. Erişim ve kayma olarak iki aşamadan oluşur. Nispeten diğer metotlara göre daha az kompleks bir denklemi vardır. Sistem Lyapunov teoremine göre stabil olacak şekilde bir kayma yüzeyi seçildikten sonra sistemi bu manifold σ üzerinde tutacak bir anahtarlama fonksiyonu (signum veya tanjant hiperbolik) içerir. Pek çok doğrusal olmayan kontrolcünün ana dayanak noktası olan Lyapunov Stabilite teoreminin bir bölümü aşağıda verilmiştir.

$$V > 0 \Rightarrow \dot{V} < 0 \Lambda V = 0 \Rightarrow \dot{V} = 0 \tag{2.51}$$

Bu teorimi kayan kipli kontrol için uygularsak kontrolcünün çalışabilmesi için aşağıdaki şartlar sağlanmalıdır.[12], [13]

$$V(\sigma(x)) = \|\sigma(x)\|_2^2 \Rightarrow \frac{dV}{dt} = \sigma^T \dot{\sigma} < 0 \tag{2.52}$$

Sistemin manifold içerisinde kalması ve belli zamanda yakınsaması içinse aşağıdaki şartlar gereklidir.

$$0 < \alpha \leq 1, \eta > 0 \tag{2.53}$$

$$\frac{dV}{dt} \leq -\eta(\sqrt{V})^{\alpha}, \frac{dV}{dt} = \sigma^T \dot{\sigma}, \|\sigma\|_2 = \sqrt{V} \tag{2.54}$$

$$\sigma\dot{\sigma} \leq -\eta\|\sigma\|_2 \tag{2.55}$$

Sistemin skaler olduğu durumlarda ise sistem stabilite durumu aşağıdaki gibi olur.

$$\text{sgn}(\sigma)\dot{\sigma} \leq -\eta \tag{2.56}$$

Daha önceden manifold seçimi yaparken her zaman pozitif ve türevinin her zaman negatif olduğunu belirttiğimiz için sistemimiz kararlı olacaktır. Üstteki şartları sağlayacak manifold seçimi yapılırsa, kayma manifoldunda sistemin davranışı sıfıra gidecek şekilde olacaktır.

Genel pratik kullanımda kayan kipli kontrol metodunda 2 çeşit kontrol girdisi tanımlanır. Birinci olarak sisteme, kayma yüzeyinde sıfıra götüren sürekli kontrol sinyali u_s ve diğeri sistemin lineer olmayan dinamiklerini kontrol etmek amaçlı kullanılan kesiklikli kontrol sinyali u_k 'nin toplamı verilir.

Oransal erişme oranlı PD tabanlı bir kayan kipli kontrol örneğinde[14]

$$\sigma = e + K_d\dot{e}, \dot{\sigma} = \dot{e} + K_d\ddot{e} \tag{2.57}$$

Bu manifold için seçilen diferansiyel kayma yüzeyi ve kontrol girdileri de aşağıdaki gibi olucaktır.

$$\dot{\sigma} = -\rho sign(\sigma) - K_p\sigma \tag{2.58}$$

$$u_s = \dot{\sigma} + K_p\sigma, u_k = \rho sign(\sigma) \tag{2.59}$$

$$u = u_s + u_k \tag{2.60}$$

Bu denklemin çıktı sonucu Hesaplanmış Tork Metodu ile aynı şekilde gibi 2. Dereceden diferansiyel bir denklem olacaktır. Hesaplanmış Tork Metodundan farklı olarak kayan kipli kontrol sistemin doğrusal olmayan dinamiklerini yenmek için sistemin denklemini önceden hesaplamak yerine kesikli bir kontrol girdisi ile bu dinamikler kontrol altında tutulur.

Bu metodun problemli yanı sürekli açma kapama yapması sebebiyle sistemde düzensizlikler oluşturup, bazen de sisteme zarar verebilmesidir. Bunu biraz azaltmak amacıyla kesikli kontrol fonksiyonu olarak signum yerine tanjant hiperbolik kullanılabilir fakat ON-OFF tipli çalışan eyleyicilerde bunu yapmak kolay olmayabilir. Aynı zamanda hızlı anahtarlama kapasiteli eyleyici sistemleri de gereksinimler arasındadır.

Diğer metodumuz olan **Yapay Sinir Ağları** metodu, köken olarak insan beynindeki öğrenmeyi bilgisayar ortamında matematiksel olarak modelleme ve uygulama mantığına dayanmaktadır. Bu şekilde insan beyninin öğrenme yapısı taklit edilerek makinelerde öğrenme yapabiliriz. Belirli bir girdi her bir yapay nöron için farklı katsayılarla çarpılarak girdi katmanındaki nöronlara transfer fonksiyonları aracılığıyla gönderilir ve nörona aktivasyon fonksiyonuna göre nöron aktif veya pasif olarak kalır.

Her ara katmanda aynı şekilde bilgiyi bir sonraki nörona aktarır. Çıkış nöronunda ise kullanım amacına uygun aktivasyon fonksiyonu ile sistem sınıflandırılır.

Yapısal olarak, YSA ileri ve geri beslemeli olarak 2 tipe ayrılır. İleri beslemeli YSA'da bilgi akış yönü girdi nörondan çıkış nöronu yönüne iken geri beslemeli versiyonda bir önceki nörona da akış vardır. Bu projede yapısal olarak daha basit olduğu için ileri beslemeli YSA kullandık. Bir ileri beslemeli YSA'nın de denklemi, girdi matrisi X giriş katmanı ağırlıkları matrisi V, çıktı katmanı ağırlıkları matrisi W, ϕ aktivasyon fonksiyonu, transfer fonksiyonu $\sum$ (çarpım değerlerinin toplamı), O çıktı sonuç olmak üzere

$$O = W^T \phi\left(V^T X\right) \tag{2.61}$$

Şeklinde matris çarpımı olarak yazılarak işlemler kolaylaştırılabilmektedir.

YSA, Öğrenme Metotları olarak da danışmanlı ve danışmansız olmak üzere ikiye ayrılır. Danışmanlı öğrenmede öncelikle ağ ağırlıkları önceden bilinen doğru çıktılara göre bir danışman tarafından çeşitli öğrenme fonksiyonu ile güncellenir ve öğrenme süreci bittikten sonra ana kontrolcüde kullanılır. Öğretme metodu olarak hatanın gradiyent, jakobiyen ve hessian değerleri çeşitli algoritmalar vasıtasıyla kullanılır. Daha çok iterasyon ile öğrenme sürecinde daha doğru sonuçlar elde edilebilmesi ve çalışan robot sistemin ekstra hesaplama yükü ilave etmemesi avantajları olsa da sadece öğretilen yörüngenin izlenebilmesi dezavantajıdır. Önceden belirlenmiş veri setlerinin mevcut olduğu veri madenciliği, sınıflandırma, görüntü tanıma gibi alanlarda daha çok kullanılırlar.

Danışmansız öğrenmede ise YSA herhangi bir müdahaleye gerek kalmadan sistem içerisinde kendi kendisine öğrenebilir. Öncede oluşturulmuş bir veri setine, danışmana ve öğrenme sürecine gerek duymaz. Robotik sistemlerde danışmansız YSA öğrenme metodu çalışmadan önce stabil olacak şekilde tanımlandığı zaman önceden planlanan yörüngenin değişmesi durumunda dahi sistem gürbüz olarak çalışmaya devam edecektir ve bu sistemi çeşitli lineer olmayan fonksiyonlar vasıtasıyla modellemeye çalışan danışmanlı öğrenmeye göre büyük bir avantajdır. En büyük dezavantajı ise bir örnek tek bir iterasyon yapabilmesi ve bu sebeple hatanın sıfıra çekilmesinde biraz daha zorlanabilmesidir.

Eğer bir yapay sinir ağı içerisinde ikiden daha fazla gizli katman barındırıyor ise bu yapay sinir ağı **Derin Öğrenme** olarak adlandırılır. Derin öğrenme kavramı oldukça yeni bir kavram olup 2006 yılında yayımlanan "A fast learning algorithm for deep belief nets" makalede üzerinden tartışılmaya başlanmıştır[15]. Daha çok şekil, görüntü tanıma, veri sınıflandırma gibi çok büyük veri setlerinin mevcut olduğu danışmanlı öğrenme metotlu işlemlerde kullanılırlar. Veri ve işlem miktarı çok fazla olduğu için eğitim sürecinde ciddi bir işlem gücüne ihtiyaç duyarken, çalışma süresi normal YSA daha kısadır. Gelişen bilgisayarlar, yeni ağırlık başlatma metotları, daha büyük veri setleri ile YSA'dan daha iyi sonuç vermektedirler. Çalışmamızda ara katman sayımızın yüksek olmasından dolayı çalışmamız da Derin Öğrenme kapsamında değerlendirilmektedir

Biz çalışmamız süresince avantajları çalışmamızın amacına daha çok uyduğu için Danışmansız YSA ve Derin Öğrenme metotlarını kullandık. Çalışmamızda; temel mantık açısından kayan kipli kontrol veya hesaplanmış tork metotları gibi sistemin belli lineer olmayan dinamiklerini online öğrenme ile hesaplayıp sisteme sağlayarak sistemin hata dinamiğini sıfıra götürmeyi amaçlamıştır.

BÖLÜM 3

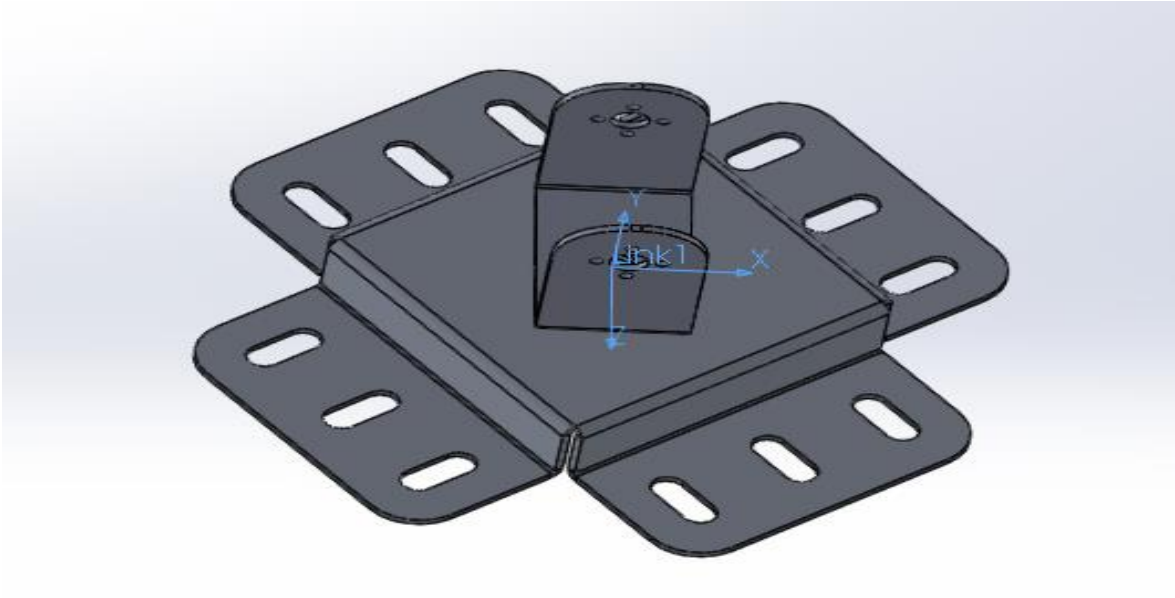
TASARIM

3.1 Mekanik Tasarım

3.1.1 Robotun 3D çizimleri ve Görünüşü Ana Mekanik Parçalar

Hipotezimiz olarak ileri sürdüğümüz kontrolcüyü deneyebilmek için bir robot dizayn edilmiştir. Bu robotumuz 3 serbesiyet dereceli olup R-R-R robot tipinde mini bir endüstriyel robot prototipindedir. Robotun malzeme seçimi olarak dayanıklı ve hafif olması açısından 2 milimetre kalınlığında alüminyum saç metal kullanılmıştır. 2 uzuvlu robota 1 tane de döner alt tabla eklenerek robotun X-Y-Z ekseninde çalışma sansına sahip olması sağlanmıştır.

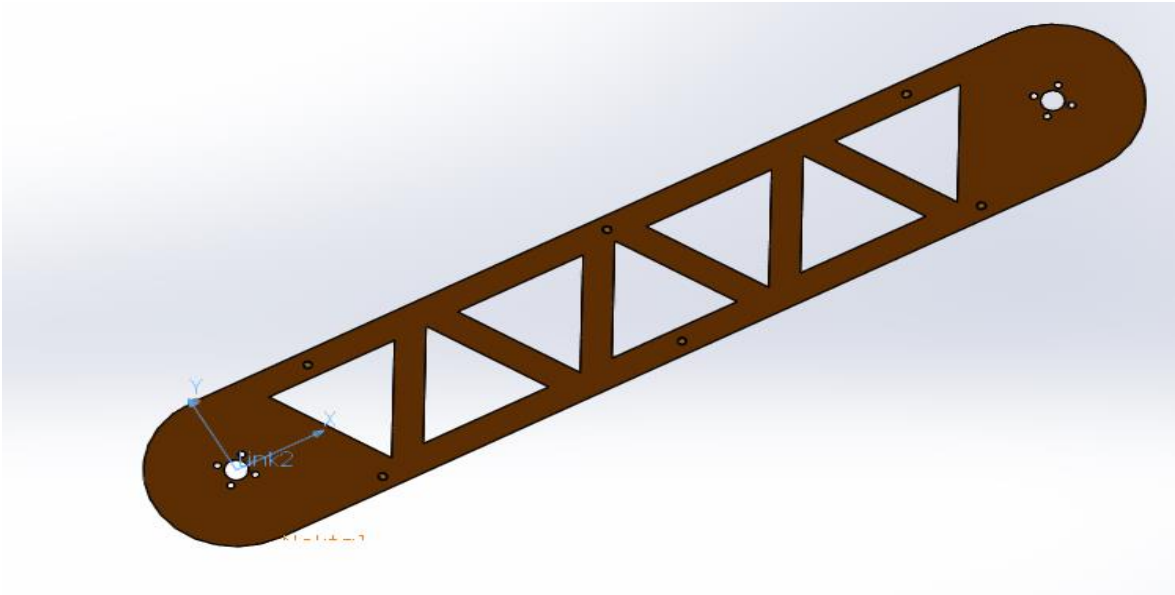
Robotumuzun parça parça çizimleri şekildeki gibidir. Z eksenindeki erişebilmek için dizayn edilen döner tabla ve sistemi dengeleyecek taban verilmiştir.



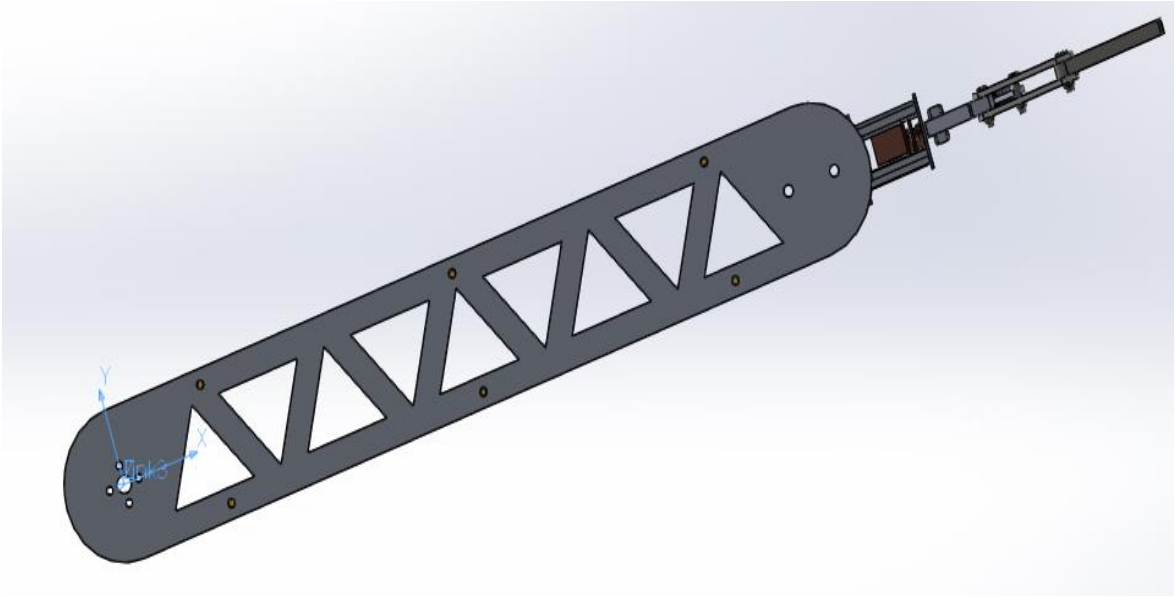
Şekil 3.1 Döner Alt tabla

Sistemin erişe bilirlik alanını artırmak ve daha optimal yörüngeleri kullanabilmek adına 2 tane uzuv tasarlandı. Üçüncü uzvun sonuna ise cisimleri tutabilmek adına elektrik motorlu tutucu kıskaç eklenmiştir.

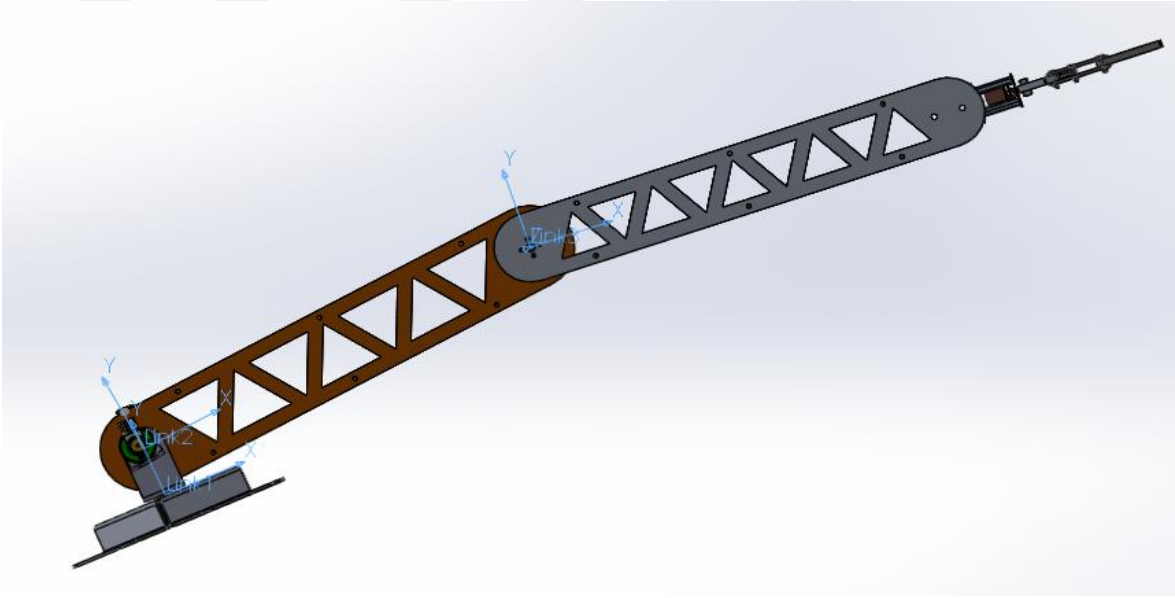
Uzuvların şekli net olarak verilmiştir. Ağırlık hafifletmek amacıyla uzuvların ortası simetrik olarak kesilmiştir.



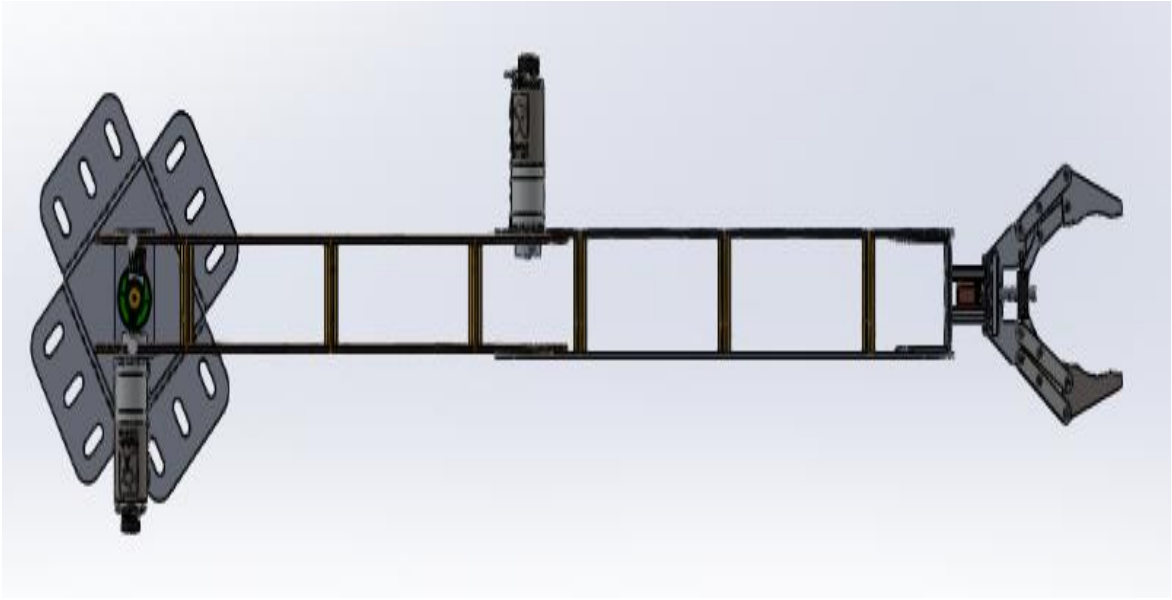
Şekil 3.2 Birinci uzuv



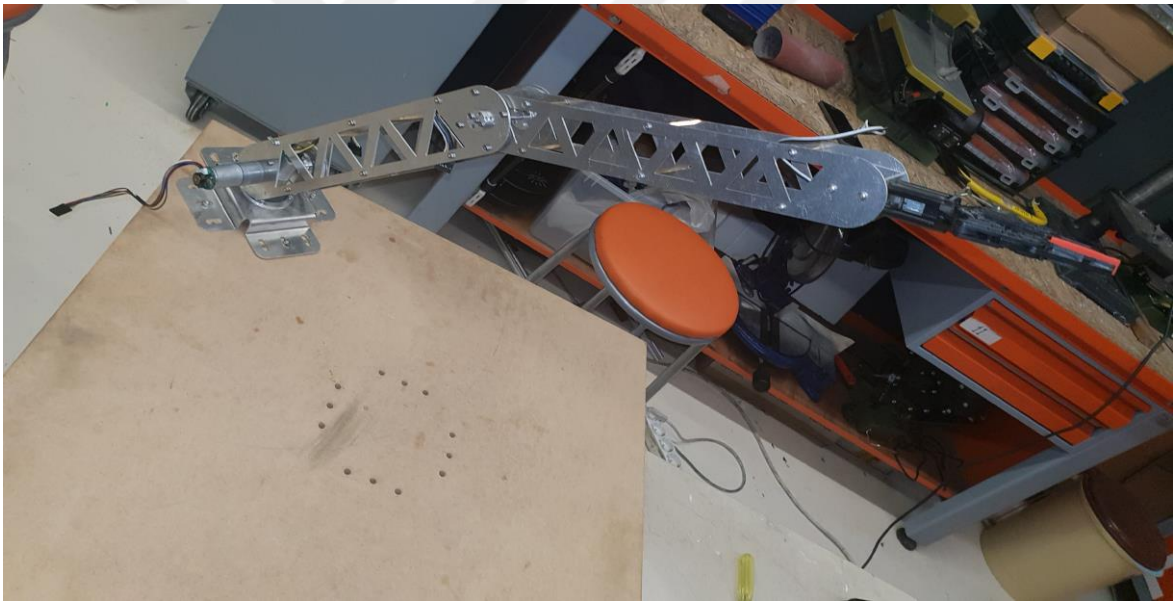
Şekil 3.3 İkinci uzuv kıskaç mekanizması



Şekil 3.4 Robotun Yandan Görünüşü



Şekil 3.5 Robotun Tepeden Görünüşü



Şekil 3.6 Dizayn Edilen Robotun Tam Resmi

Tutucu eklenti olarak ise Make block Robot Gripper seçilmiştir. Kendini kilitleyen kayma önleyici sistemi, çabuk açılma kapanma süresi (1sn) hafif kütlesi ve 1.5 kilograma kadar kaldırabilmesi, 12 volt ile çalışırken 0.3 amper akım çekmesi sistemimiz için iyi bir seçenek haline getirmiştir. Buna ek olarak N20 orta vida sistemi ile aşırı akım koruması da sağlamaktadır.



Şekil 3.7 Makeblock Robot Gripper

3.1.2 Yan Parçalar

Sistemimizde olası sürtünmeleri azaltmak amacıyla rulman kullanmamız gerekliydi. Bu sebeple AXK 2542 kodlu iç yarı çapı motorumuzun redüktör yarıçapına eşit olan rulmanı kullandık.



Şekil 3.8 AXK 2542

Rulmanın kaymasını önlemek amacıyla kalınlıkları 2-4 mm arası değişen rulman yataklarını SolidWorks programı ile dizayn ettirip ürettik.



Şekil 3.9 Rulman Yatakları

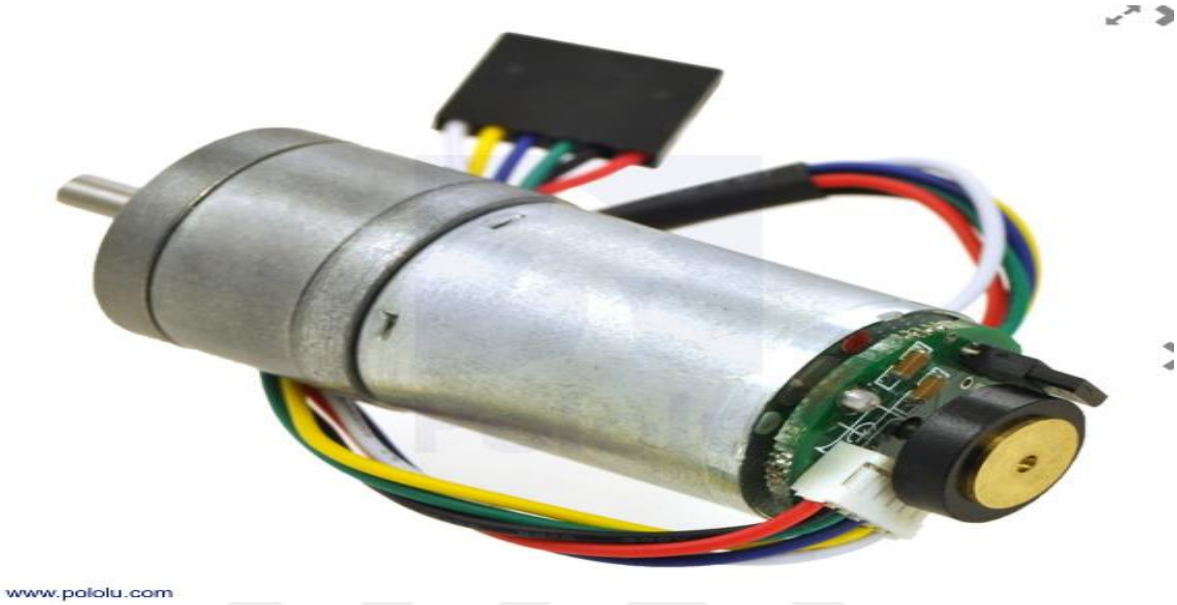
3.2 Elektronik Dizayn

3.2.1 Eyleyiciler ve Sürücü Elektroniği

Sistemimizde 3 tanesi hassas kontrol gerektiren 4 adet motora gereksinimimiz vardı. Robotumuzun boyutları küçük olduğu için eyleyici seçerken mecburen küçük boyutlu DC motorlar seçmek zorunda kaldık. Bu durumda en zorlayıcı nokta DC motorların düşük torkları olmasıydı. Bunu aşmak için yüksek redüktör oranlı motorlar mevcut olsa da bu da hız sınırlamalarını beraberinde getirmektedir ($\tau_1\omega_1 = \tau_2\omega_2$) . Kontrol elektroniğinin kolaylığı ve motor parametrelerinin doğrusal olması önemli bir avantajları olduğu için fırçalı DC motor tipini tercih ettik.

Aynı zamanda mekanizmamız seri bir mekanizma olduğu için 1. Motor, 2.Motor'un ağırlığını da taşımak zorundadır. Bu sebeple motorumuzun ağırlık başına ürettiği tork önemli bir parametre olacaktır. Bundan hareketle sistemimize sığabilecek motorlar arasında en yüksek (20.3698 Nm/kg) birim kilogram başına torka ve en yüksek birim akım başına torka (0.378297Nm/A) sahip 3 adet Pololu 3219 99:1 redüktörlü DC Motor'u seçtik.

Araştırma sırasında robotun hassas kontrolü gerekeceği için enkoderli yüksek hassasiyetli 4 faz 48 pals versiyon olan tipi tercih ettik. [16] $(360 / \left(\frac{118533536}{25000}\right) = 0.07592^{\circ}$ derece hassasiyet).



Şekil 3.10 PL3219 99:1 DC Motor

Dimensions

Size:	25D x 66L mm
Weight:	104 g
Shaft diameter:	4 mm

General specifications

Gear ratio:	98.78:1
No-load speed @ 12V:	100 rpm
No-load current @ 12V:	300 mA
Stall current @ 12V:	5600 mA
Stall torque @ 12V:	300 oz-in[1]
No-load speed @ 6V:	50 rpm[2]
Stall current @ 6V:	2800 mA[2]
Stall torque @ 6V:	150 oz-in[2]
Lead length:	8 in[3]
Motor type:	5.6A stall @ 12V (HP 12V)
Encoders?:	Y

Şekil 3.11 PL3219 Teknik Özellikleri

Motorumuzun genel denklem ise aşağıda verilmiştir.

$$\tau = K_t i - \tau_l = J\ddot{\theta} + b\dot{\theta} \tag{3.1}$$

$$V = Ri + L\frac{di}{dt} + K_{\dot{\theta}}\dot{\theta} \tag{3.2}$$

Motor sürme işini voltaj kontrolü yaparak gerçekleştireceğimiz için ve ideal koşullarda bozucu etkiler, armatür endüktansı ihmal edilebilecek kadar küçük olduğu için motora vermemiz gereken voltajın formülü aşağıdaki gibi olacaktır.

$$V = \left(K_w + \frac{Rb}{K_t} \right) \dot{\theta} + \left(\frac{JR}{K_t} \right) \ddot{\theta} \tag{3.3}$$

Motor Sürücüsü olarak ise 4 adet H köprü devresi kullandık. Bunun sebebi ise daha kompleks motor sürücülerinin içerisindeki kontrolcü tipi dışında kontrolcülere izin vermemesi ve bizimde kontrolcüler üzerine çalışıyor olmamızdı.

3.3 Kontrolcü Dizayn

3.3.1 Kontrolcü Mantığı ve Matematiksel Denklem

Kontrolcümüzün mantığı hipotezde belirttiğimiz üzere sistemin bilinmeyen kuvvetlerini stabil şekilde tutup, sistemin lineer olmayan modelini lineer şekilde kontrol edebilmeye dayanmaktadır. Hesaplanmış Tork Metodunda; sistemin matematiksel denkleminin önden hesaplanıp yeni kontrol girdisi verilmesiyle, Kayan Kipli Kontrol ’de sistemin doğrusal olmayan faktörlerin açma kapama fonksiyonu yardımıyla kontrol edilmesiyle, Bulanık Mantık Metodunda kontrolcünün parametrelerinin bulanık mantık yardımıyla yenilerek yapılan sistemin istenen doğrusal diferansiyel denklemi izlemesi metodu bizim kontrolcümüzde doğrusal olmayan dinamiklerin öğrenilmesiyle yapılacaktır.

Kontrolcümüz için temel mantık aşağıdaki denklemlerde verilmiştir.[3]

$$\tau = H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g(q) + \tau_d \tag{3.4}$$

$$e = q_d - q, r = \dot{e} + \lambda e, \lambda > 0 \tag{3.5}$$

$$H(q)\dot{r} + C(q,\dot{q})r - f(x) + \tau = 0 \tag{3.6}$$

$$f(x) = H(q)(\ddot{q}_d + \lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \lambda e) + \tau_g + \tau_d \tag{3.7}$$

Bizim önerdiğimiz kontrolcü dizaynına göre $f(x)$ bilinmeyen değişkeni yapay sinir ağı tarafından öğrenilerek kontrolcüye eklenecek ve kontrolcü girdimiz ile ilgili hesaplamalar aşağıdaki denklemlerdeki gibi olacaktır.

$$Z=\begin{bmatrix} W & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & V \end{bmatrix}, \|Z\|_F \le Z_b \tag{3.8}$$

$$C_z > 0, \vartheta(t) = -C_z r\left(\left\|\hat{Z}\right\|_F + Z_B\right) \tag{3.9}$$

$$\tau_{NN} = \hat{f}(x), C_v > 0 \Rightarrow \tau = C_v r + \tau_{NN} - \vartheta \tag{3.10}$$

Sistemimizdeki f_x değerinin hesaplanabilmesi için bir nöron ağırlık değerlerini online olarak güncelleyen danışmansız bir algoritma tasarlandı. Algoritmamızda Çok katmanlı ileri beslemeli derin yapay sinir ağı kullandık. Toplam katman sayımız 2 ile 7 arasında değişken olarak tanımlandı. Danışmansız öğrenme metodu kullandığımız için ve her örnekleme için bir basamak öğrenme işlemi yapma şansımız seçtiğimiz aktivasyon fonksiyonun matematiksel yapı olarak basit olmasını istedik bu sebeple ReLU fonksiyonunu seçtik. Gradient bazlı öğrenme sürecinde ReLU fonksiyonun türevinden kaynaklanabilecek muhtemelen sorunları aşmak için Hebbian öğrenme metodu kullandık.

Yapay sinir ağımız için girdi değerlerimiz ve n sayıdaki katman sayısı için çıkışlarımız ve ağımızın görünüşü aşağıdaki gibi olacaktır.

$$x(t)=\begin{bmatrix} e \\ \dot{e} \\ q_d \\ \dot{q}_d \\ \ddot{q}_d \end{bmatrix}, \phi(a) = \mathrm{Re}LU(a) \tag{3.11}$$

$$n=2 \rightarrow \hat{f}(x) = \hat{W}^T \phi\left(\hat{V}^T x\right) \tag{3.12}$$

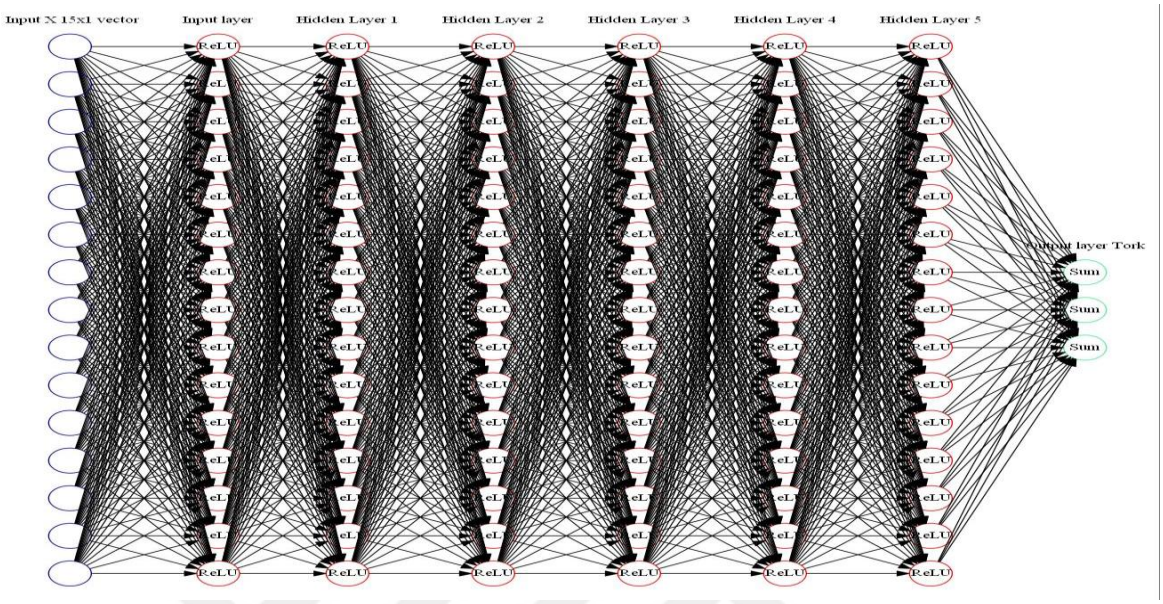
$$n=3 \rightarrow \hat{f}(x) = \hat{W}^T \phi\left(\hat{R}^T \phi\left(\hat{V}^T x\right)\right) \tag{3.13}$$

$$n=4 \rightarrow \hat{f}(x) = \hat{W}^T \phi\left(\hat{N}^T \phi\left(\hat{R}^T \phi\left(\hat{V}^T x\right)\right)\right) \tag{3.14}$$

$$n=5 \rightarrow \hat{f}(x) = \hat{W}^T \phi\left(\hat{P}^T \phi\left(\hat{N}^T \phi\left(\hat{R}^T \phi\left(\hat{V}^T x\right)\right)\right)\right) \tag{3.15}$$

$$n=6 \rightarrow \hat{f}(x)=\hat{W}^T\phi\left(\hat{M}^T\phi\left(\hat{P}^T\phi\left(\hat{N}^T\phi\left(\hat{R}^T\phi\left(\hat{V}^T x\right)\right)\right)\right)\right) \tag{3.16}$$

$$n=7 \rightarrow \hat{f}(x)=\hat{W}^T\phi\left(\hat{C}^T\left(\hat{M}^T\phi\left(\hat{P}^T\phi\left(\hat{N}^T\phi\left(\hat{R}^T\phi\left(\hat{V}^T x\right)\right)\right)\right)\right)\right) \tag{3.17}$$



Şekil 3.12 5 ara katmanlı için YSA Modeli

Hebbian Öğrenme Kuralı ve ReLU Aktivasyon Fonksiyonu

Hebb teorisi Nöropsikoloji alanın da öncü Donald Hebb tarafından nöronların davranışını açıklamak için ortaya atılmıştır. Hebb teorisi; birlikte ateşlenen nöronların birlikte hareket edeceğini, girdi ve çıktının bağlantılı olduğunu öne süren nöropsikolojik bir teoridir. Bu teori yapay zekâ kavramı ve YSA'nın gelişmesiyle, YSA için öğrenme teorisi olarak denenmiştir.

Matematiksel Hebb Kuralında bir nöronun girdi ve çıktı sonuçlarının çarpımı nöronun değerine eklenerek öğrenme yapılmış olur. Y_o Nöron sonuç çıktısı ve Y_i girdisi olmak üzere Nöron ağırlık değişimi [3]

$$\Delta W=\eta Y_i Y_o \tag{3.18}$$

Şeklinde olacaktır. Kesikli cebir yerine sürekli cebir kullanılan bir öğrenme içinse

$$\dot{W}=\eta Y_i Y_o \tag{3.19}$$

Metodu kullanılabilir.

Bu metodun en büyük avantajı Gradient, Türev, Jakobiyen veya Hessian hesabı gerektirmemesi ve basit bir çarpma işleminden oluştuğu için işlemciyi fazla yormamasıdır. Aynı zamanda ReLU aktivasyon fonksiyonun en büyük dezavantajı olan, negatife gelinince nöronun ölmesi, sorunu da bu sayede aşıla bilenecek ve ReLU kullanımının önü açılacaktır.

ReLU'nun diğer bir dezavantajı olarak tüm katmanlarda pozitif gelmesi durumunda doğrusallaşma sıkıntısını da testlerimizde denedik. 7 katmanlı, 2 katmanlı ve değişken katmanlı YSA kontrolcü sistemleri dizayn ettik.

Bu sebeplerden dolayı algoritmamız da Hebbian kuralını temel aldık. Bunun yanında hatanın sıfıra gideceği diğer metotların aksine tam olarak kesin değildir. Bu sebeple bizim sistemimiz için bu metodu kullanırken sistemimizde bu metodun kararlı olduğunu ispatladık.

Sistemimizin gürbüz olabilmesi için. η Pozitif tanımlı matris veya katsayı, $\kappa > 0$ ve oldukça küçük bir sayı olmak üzere öğrenme algoritmamız aşağıda verilmiştir.[3]

$$\alpha_1 = \phi(\hat{V}^T x), \alpha_2 = \phi(\hat{R}^T \alpha_1), \alpha_3 = \phi(\hat{N}^T \alpha_2), \alpha_4 = \phi(\hat{P}^T \alpha_3) \tag{3.20}$$

$$\alpha_5 = \phi(\hat{M}^T \alpha_4), \alpha_6 = \phi(C^T \alpha_5), \alpha_7 = \hat{W}^T \alpha_6 \tag{3.21}$$

$$\begin{cases} n = 2 \Rightarrow \dot{W} = \eta(\alpha_1 r^T - \kappa \|r\| \hat{W}) \\ \dot{\hat{V}} = \eta \|r\| (x \alpha_1^T - \kappa \hat{V}) \end{cases} \tag{3.22}$$

$$\begin{cases} n = 3 \Rightarrow \dot{\hat{W}} = \eta(\alpha_2 r^T - \kappa \|r\| \hat{W}) \\ \dot{\hat{R}} = \eta \|r\| (\alpha_1 \alpha_2^T - \kappa \hat{R}) \\ \dot{\hat{V}} = \eta \|r\| (x \alpha_1^T - \kappa \hat{V}) \end{cases} \tag{3.23}$$

$$\begin{cases} n = 4 \Rightarrow \dot{\hat{W}} = \eta(\alpha_3 r^T - \kappa \|r\| \hat{W}) \\ \dot{\hat{N}} = \eta \|r\| (\alpha_2 \alpha_3^T - \kappa \hat{N}) \\ \dot{\hat{R}} = \eta \|r\| (\alpha_1 \alpha_2^T - \kappa \hat{R}) \\ \dot{\hat{V}} = \eta \|r\| (x \alpha_1^T - \kappa \hat{V}) \end{cases} \tag{3.24}$$

$$\begin{cases} n=5 \Rightarrow \dot{\hat{W}} = \eta(\alpha_4 r^T - \kappa \|r\| \hat{W}) \\ \dot{\hat{P}} = \eta \|r\| (\alpha_3 \alpha_4^T - \kappa \hat{P}) \\ \dot{\hat{N}} = \eta \|r\| (\alpha_2 \alpha_3^T - \kappa \hat{N}) \\ \dot{\hat{R}} = \eta \|r\| (\alpha_1 \alpha_2^T - \kappa \hat{R}) \\ \dot{\hat{V}} = \eta \|r\| (x \alpha_1^T - \kappa \hat{V}) \end{cases} \quad (3.25)$$

$$\begin{cases} n=6 \Rightarrow \dot{\hat{W}} = \eta(\alpha_5 r^T - \kappa \|r\| \hat{W}) \\ \dot{\hat{M}} = \eta \|r\| (\alpha_4 \alpha_5^T - \kappa \hat{M}) \\ \dot{\hat{P}} = \eta \|r\| (\alpha_3 \alpha_4^T - \kappa \hat{P}) \\ \dot{\hat{N}} = \eta \|r\| (\alpha_2 \alpha_3^T - \kappa \hat{N}) \\ \dot{\hat{R}} = \eta \|r\| (\alpha_1 \alpha_2^T - \kappa \hat{R}) \\ \dot{\hat{V}} = \eta \|r\| (x \alpha_1^T - \kappa \hat{V}) \end{cases} \quad (3.26)$$

$$\begin{cases} n=7 \Rightarrow \dot{\hat{W}} = \eta(\alpha_6 r^T - \kappa \|r\| \hat{W}) \\ \dot{\hat{C}} = \eta \|r\| (\alpha_5 \alpha_6^T - \kappa \hat{C}) \\ \dot{\hat{M}} = \eta \|r\| (\alpha_4 \alpha_5^T - \kappa \hat{M}) \\ \dot{\hat{P}} = \eta \|r\| (\alpha_3 \alpha_4^T - \kappa \hat{P}) \\ \dot{\hat{N}} = \eta \|r\| (\alpha_2 \alpha_3^T - \kappa \hat{N}) \\ \dot{\hat{R}} = \eta \|r\| (\alpha_1 \alpha_2^T - \kappa \hat{R}) \\ \dot{\hat{V}} = \eta \|r\| (x \alpha_1^T - \kappa \hat{V}) \end{cases} \quad (3.27)$$

3.3.2 Saklı Katman Sayısı Değiştirme Algoritması

YSA sistemimizde gizli katman sayısı artıkça hata oranımız azalmaktadır fakat yaklaşık 5 gizli katmandan sora hatamızdaki azalma oranı oldukça düşük olup zamanla bu oran doyuma ulaşmaktadır. YSA algoritmamız n gizli sayısına karşılık $O(n)$ karmaşıklık derecesine sahip olup her bir katman karşılığında n^2 tane işlem sistemimize eklenmektedir. Kaynakların gereksiz kullanımının önlenmesi amacıyla bir gizli katman sayısı ayarlama metodu geliştirdik.

Metodumuzda yapılacak işin periyodik olması durumunda periyodik olarak, periyodik olmaması durumunda ise belli bir örnekleme zamanına bağlı olarak katman sayısının birere ikişer artırılıp azaltması olarak ayarladık. Azaltma kriteri olarak ise hatanın norm 'unun ortalamasını kullandık. Ortalama almak için kullandığımız integral metodu ise $T_{ö}$ örnekleme süresi olmak üzere

Mantıksal kodumuz aşağıdaki şekilde olmuştur:

$$\begin{cases} If((T \equiv (\mathrm{mod} T_{ö})) == 0 \,\&\&\, \|r\| \geq E_{\max}) \Rightarrow n = n+2 \\ if((T \equiv (\mathrm{mod} T_{ö})) == 0 \,\&\&\, E_{high} \leq \|r\| < E_{\max}) \Rightarrow n = n+1 \\ if((T \equiv (\mathrm{mod} T_{ö})) == 0 \,\&\&\, E_{low} \leq \|r\| < E_{high}) \Rightarrow n = n \\ if((T \equiv (\mathrm{mod} T_{ö})) == 0 \,\&\&\, E_{\min} \leq \|r\| < E_{low}) \Rightarrow n = n-1 \\ if((T \equiv (\mathrm{mod} T_{ö})) == 0 \,\&\&\, E_{\min} < \|r\|) \Rightarrow n = n-2 \end{cases} \quad (3.28)$$

Başlangıçta sistemin hızlı öğrenebilmesi açısından 7 katmanla başladık ve zamanla öğrenme tamamlandıkça katman sayısını düşürdük. Gerek gizli katman sayısı ayarlama gerek Hebbian öğrenmesiyle türevden kurtulma gerekse ReLU sayesinde işlemleri basite indirgeyip bazı nöronlardan 0 değeri çıkarma sayesinde işlemcimizin yükü düşürülerek örnekleme sayısı artırılmış ve daha verimli çalışması sağlanmıştır.

3.3.3 Kontrolcünün Matematiksel İspatı

Yedi katmanlı YSA için hata denklemimiz aşağıda verilmiştir

$$\tilde{W} = W - \hat{W} \quad (3.29)$$

$$H\dot{r} = -(K_v + C)r + \tilde{W}^T \alpha_6 + W^T \alpha_6 + (\varepsilon + \tau_d) + \vartheta \quad (3.30)$$

İlk etapta; ideal ağırlık oranlarında, f_x fonksiyonun öğrenmesindeki hatanın ε sabit sayısını geçemeyeceği ϑ 'nin 0 olduğu ve gerçek ağırlık değeri ile tahmin edilen ağırlık değerinin aynı olduğu varsayımlarına göre hareket ederek en ideal şartlar için ispat yaparsak;

$$f_x = \alpha_7 + \varepsilon, \|\varepsilon\| \leq \varepsilon_n, \|\tau_d\| \leq d_b \quad (3.31)$$

Sistemimizin hata dinamiği denklem 2.37, 2.40, 2.43 e göre aşağıdaki gibi olacaktır.[3]

$$H\dot{r} = -\left(K_v + C\right) + (\varepsilon + \tau_d) \quad (3.32)$$

H ve C matrisi matematiksel özellikleri nedeniyle maksimum değer olarak sınırlıdır.

$$I_n\lambda_{Min} \le H \le I_n\lambda_{Max}, \|C\| \le k_0\|\dot{q}\| \tag{3.33}$$

Bu durumda H ve C matrisinin maksimum değerleri üzerinden çözüm yaparsak

$$a_1 = \left(K_v + k_0\|\dot{q}\|_{\max} - \lambda I_n\lambda_{\max}\right), a_2 = a_1\left(K_v + k_0\|\dot{q}\|_{\max}\right) \tag{3.34}$$

$$a_3 = d_b + \varepsilon_n + \lambda\left(e(0)\left(K_v + k_0\|\dot{q}\|_{\max}\right) + I_n\lambda_{\max}\dot{e}(0)\right) \tag{3.35}$$

$$a_4 = -I_n\lambda_{\max}\left(d_b + \varepsilon_n + \lambda\left(e(0)\left(K_v + k_0\|\dot{q}\|_{\max}\right)\right) + k_0\|\dot{q}\|_{\max}\dot{e}(0)\right) \tag{3.36}$$

$$e(t) = \left(e^{-\lambda t}a_3 - \left(d_b + \varepsilon_n\right)\right)\left(\lambda a_1\right)^{-1} + \left(\left(a_4 e^{-t\left(K_v + k_0\|\dot{q}\|_{\max}\right)\left(I_n\lambda_{\max}\right)^{-1}}\right) + I_n\lambda_{\max}\left(d_b + \varepsilon_n\right)\right)a_2^{-1} \tag{3.37}$$

Hata denklemine göre eğer K_v yeterince büyük olmazsa sistemimiz sabit hatayı azaltmayabilir ve hatta artırabilir. Bu durumdan kaçınmak amacıyla, K_v yeterince büyük seçildiği zaman herhangi bir pozitif kutup gelmeyeceği için sistemimiz stabil olacaktır.

$f(x)$'in YSA tarafından öğrenilmesi ve bu sırada meydana gelen hata denklemlerinin de stabil olup olmadığını Lyapunov Teorimine göre kontrol edeceğiz.[3]

Lyapunov fonksiyonu adayımız 7 katmanlı YSA içeren bir sistem için

$$L(r,\tilde{W}\ldots,\tilde{V}) = \frac{\left(r^T Hr + tr\left(\tilde{W}^T\eta^{-1}\tilde{W} + \tilde{C}^T\eta^{-1}\tilde{C} + \tilde{M}^T\eta^{-1}\tilde{M} + \tilde{P}^T\eta^{-1}\tilde{P} + \tilde{N}^T\eta^{-1}\tilde{N} + \tilde{L}^T\eta^{-1}\tilde{L} + \tilde{V}^T\eta^{-1}\tilde{V}\right)\right)}{2} \tag{3.38}$$

$$\dot{L} = r^T\left(K_v r + W^T\tilde{\alpha}_6 + (\varepsilon + \tau_d)\right) + \|r\|tr\left(\kappa\tilde{W}^T(W - \tilde{W}) + .. + \kappa\tilde{V}^T(V - \tilde{V}) + \tilde{V}^T x\alpha_1^T + .. + \tilde{C}^T\alpha_5\alpha_6^T\right) \tag{3.39}$$

$$\dot{L} = r^T\left(K_v r + W^T\tilde{\alpha}_6 + (\varepsilon + \tau_d)\right) + \|r\|tr\left(\kappa\tilde{Z}^T(Z - \tilde{Z}) + \tilde{V}^T x\alpha_1^T + .. + \tilde{C}^T\alpha_5\alpha_6^T\right) \tag{3.40}$$`

İz ve frobenius norm işlemlerinin dağılma ve eşitsizlik özelliklerini kullanırsak

$$tr\left\{C^T D\right\} = < C, D > \le \|C\|_F\|D\|_F \tag{3.41}$$

$$\dot{L} \le -\|r\|\left(K_{v_{\min}}\|r\| - \kappa\|\tilde{Z}\|_F(Z_B - \tilde{Z}) - \|\tilde{V}\|_F\left\|x\hat{\alpha}_1^T\right\|_F \ldots - \|\tilde{W}\|_F\|\tilde{\alpha}_6\|_F + K_z(\|\tilde{Z}\|_F + \|r\|Z_b) - \left(\sqrt{L}Z_B + d_b + \varepsilon_b\right)\right) \tag{3.42}$$

Sistemimiz için X ve r matrislerinin maksimum ulaşabilecekleri limit değerleri bir katsayıyı geçemez.

$$\left\| \begin{matrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{matrix} \right\| \le q_b \tag{3.43}$$

$$S_x = \left\{ x \,|\, \|x\| \le c_1 + c_2 \|r\| \Lambda q_b + c_0 \|r_0\| + c_2 \|r\| \Lambda \|x\| < b_x \right\} \tag{3.44}$$

$$S_r = \left\{ r \,|\, \|r\| < \frac{b_x - q_b}{c_0 + c_2} \right\} \tag{3.45}$$

$K_z > c_2$ Şartını sağladığımız durumda

$$O_3 = \frac{\left(\kappa Z_B + c_1\right)}{2\kappa}, D = \kappa O_3^2 + \sqrt{L} Z_B + d_b + \varepsilon_n \tag{3.46}$$

$$\dot{L} \le 0 \Leftrightarrow K_{v_{\min}} \|r\| + \kappa(\left\|\tilde{Z}\right\|_F - O_3) - U \ge 0 \tag{3.47}$$

$$K_{v_{\min}} > \frac{U}{\|r\|} \tag{3.48}$$

Denklem 2.82'deki şart sağlandığı sürece Lyapunov fonksiyonumuz her durumda pozitif olmasına karşın türevi her şartta negatif olacaktır ve sistemimiz Lyapunov teoremine göre kararlı olacaktır. Pratikte ise U değerini belirlemek kolay olmadığı için K_v ve K_z değerleri yeterince büyük olduğu zaman sistemimiz stabil olacaktır.

BÖLÜM 4

BENZETİM

4.1 Benzetim Sonuçları

Geliştirdiğimiz kontrolcümüzü tasarladığımız robotumuzda Matlab-Simulink ve SimMechanics yardımıyla ile matematiksel olarak çalıştırdık. SimMechanics kullanmamızın sebebi gerçeği olabildiğince yakın değerler elde edebilmekti.

Sistemimizi iki hedef yörünge için benzetim yaptık. İlk yörünge hedefimizde sadece eklem açılarını benzetim yapmak istedik ve türevi hiçbir zaman tamamen sıfır olmayacağı ve periyodik olduğu için sin(t) fonksiyonunu seçtik.

Simülasyon süresi olarak 90 saniye seçtik ve ara katman değiştirme algoritmasın hatanın normu [0.6 0.8] arasında ise değişim yapmaması farklı ise değişim yapmasını istedik ve hata istediğimiz aralıktan düşük ise derin öğrenmeye gerek kalmadığına sistem karar versin istedik.

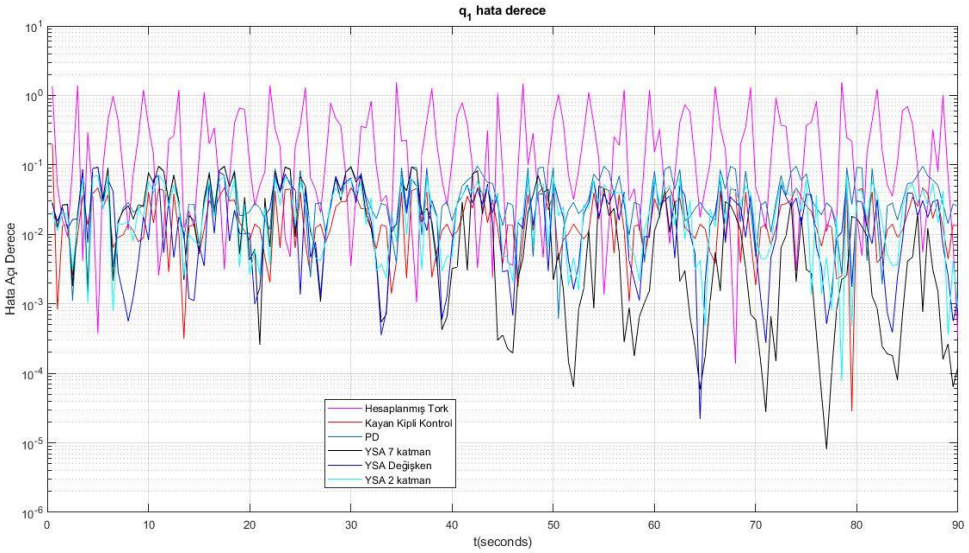
Her iki simülasyon için de her katmanda 15 nöron kullandık ve başlangıç ağırlıklarını matris olarak atarken birim matris atadık bu durumda tüm ağırlık matrislerini içeren Z matrisinin frobenius normunu da maksimum $Z_b \approx 5$ olarak alabiliriz.

Simülasyon için sabitlerimiz

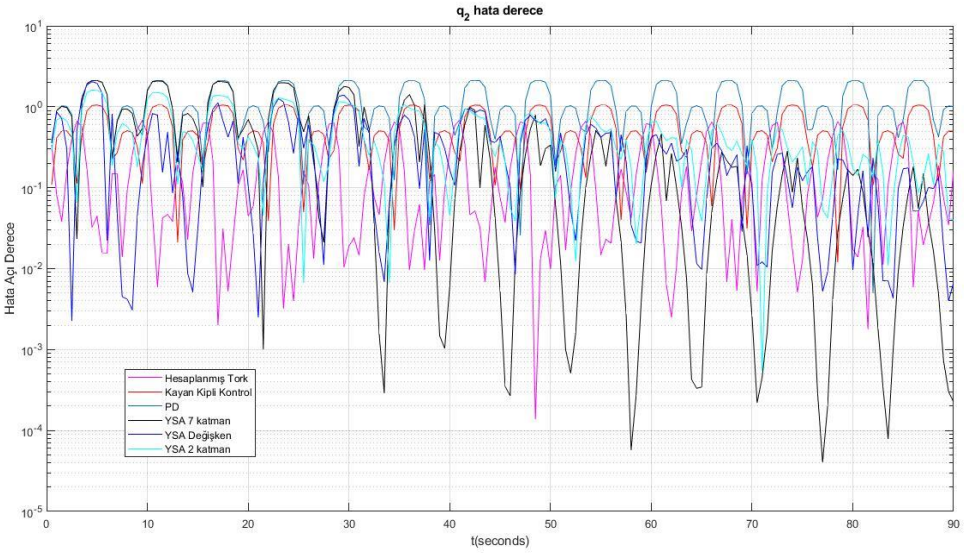
$$K_v = \lambda = 10, \kappa = 0.001, \eta = 0.05, E_{\max} = 0.9, E_{high} = 0.8, E_{low} = 0.6, E_{\min} = 0.3$$

Her iki simülasyon içinde sistemin ana kutbunun -10 olmasını istedik. Bu sebeple Hesaplanmış Tork metodu için $K_v = 20, K_p = 100$ ve doğrusal erişimli kayan kipli kontrol için ise $K_d = 10, K_p = 10$ ve açma kapama tork sinyali 0.1 seçilmiştir.

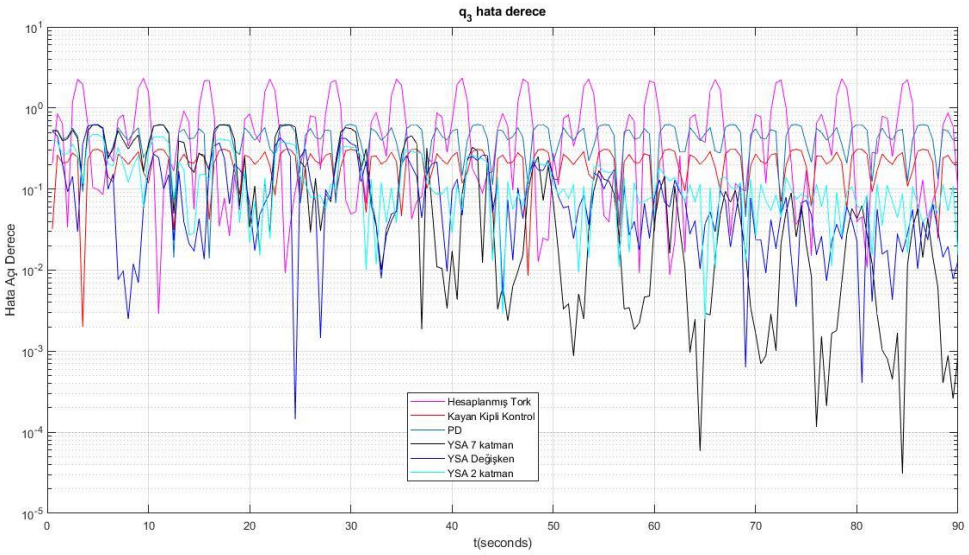
Sistemi biraz daha zorlamak adına tüm denemelerde Tork çıktısına %1 oranında bozucu **Gaussian Beyaz Gürültü** eklenmiştir. Tüm simülasyonlarda Runge-Kutta metodu ve 0.001 saniye adım büyüklüğünü kullandık. Eklem uzayındaki yörünge hedefleri takibi için sonuçlar sırasıyla şekillerde verilecektir. Logaritmik skala kullanılmıştır. ReLU katman sayısının etkisini ölçmek içinse farklı ReLU katmanlarıyla deney yaptık.



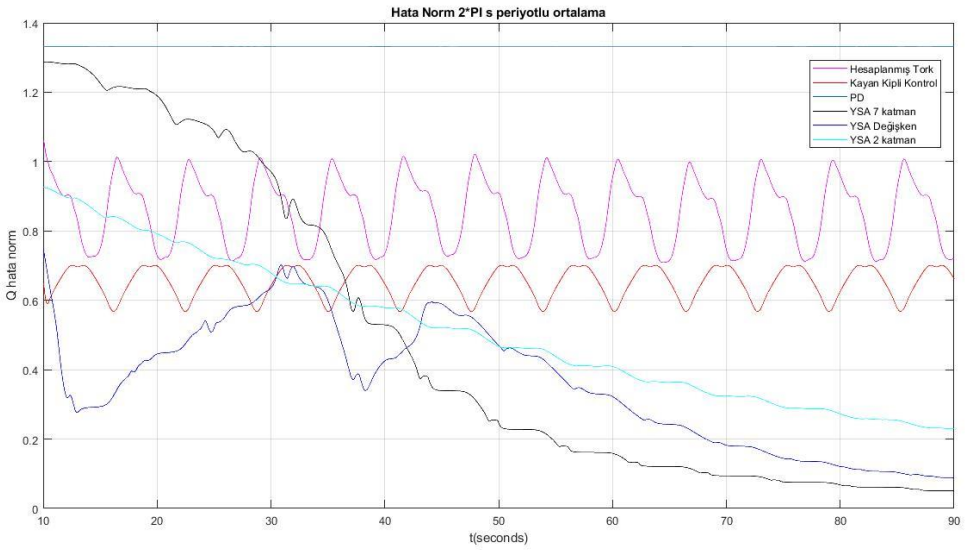
Şekil 4.1 1. Eklem için Hata



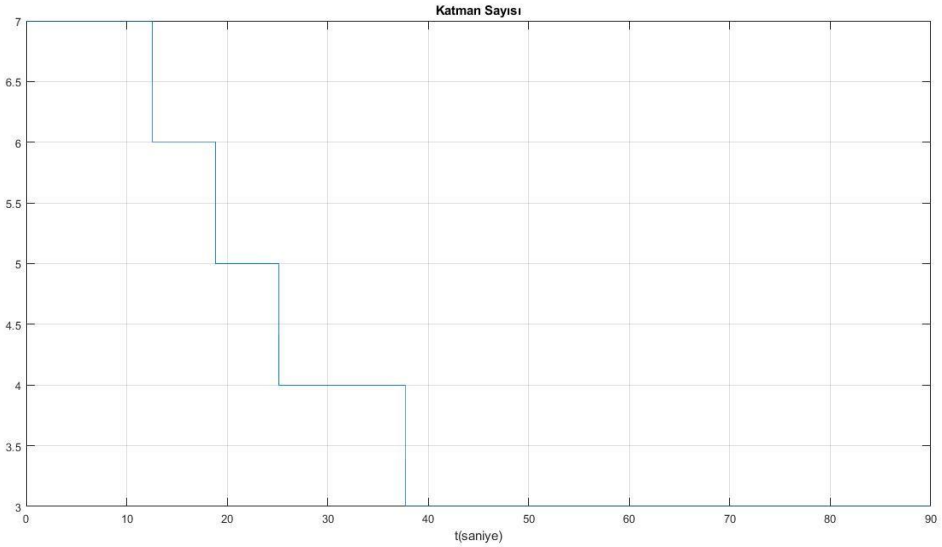
Şekil 4.2 2.Eklem için Hata



Şekil 4.3 3.Eklem için Hata



Şekil 4.4 Hatanın Normu 2π Periyotlu Ortalamaları



Şekil 4.5 YSA Toplam Katman Sayısı

İkincil olarak ise Konum Uzayında O ve Z harflerini yazdıran bir hedef yörünge belirledik. Yörüngeyi belirlerken 5. Derecen polinom cinsinden yörünge atadık ve bunu sönümlü kareler metoduyla ile eklem uzayına çevirdik.5. Derece seçilmesinin sebebi ivmenin 2. Derece polinom gelmesi ve sıfırdan başlayıp yine sıfıra dönerek sistemin tork gereksinimini düşürmesidir.

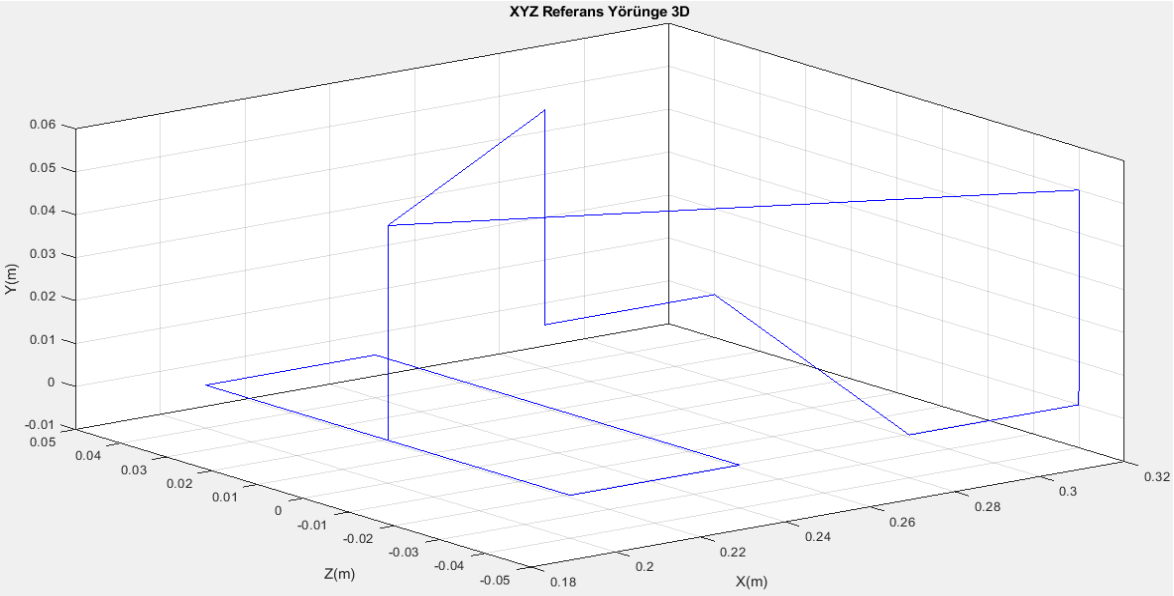
$$n=5, x(t)=\sum_{i=0}^{5} k_i t^i, \dot{x}(t)=\sum_{i=1}^{5} i(k_i t^{i-1}), \ddot{x}(t)=\sum_{i=2}^{5} i(i-1)(k_i t^{i-2}) \quad (3.49)$$

$$A=\begin{bmatrix}1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5\\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4\\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 24t_0^3\end{bmatrix}, B=\begin{bmatrix}1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5\\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4\\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 24t_f^3\end{bmatrix}, K=\begin{bmatrix}k_0\\ k_1\\ k_2\\ k_3\\ k_4\\ k_5\end{bmatrix} \quad (3.50)$$

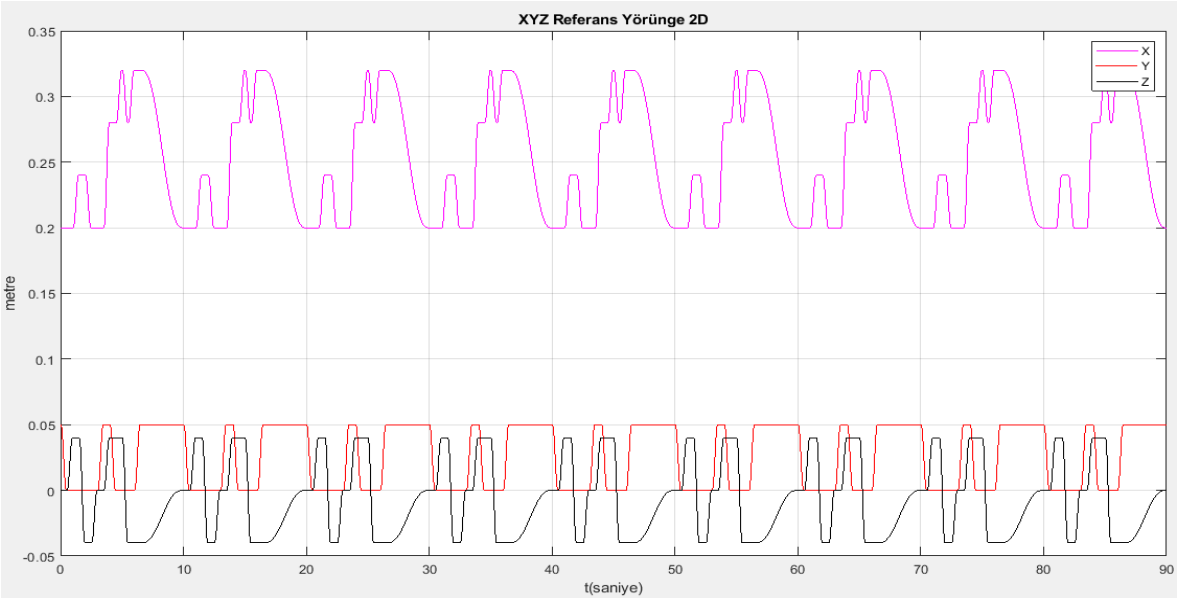
$$K=\left(\begin{bmatrix}A\\ B\end{bmatrix}\right)^{-1}\begin{bmatrix}x_0\\ \dot{x}_0\\ \ddot{x}_0\\ x_f\\ \dot{x}_f\\ \ddot{x}_f\end{bmatrix} \quad (3.51)$$

Metodu ile katsayılar bulunarak hedef yörüngeler daha efektif şekilde atana bilinmektedir ve parametreler önceden belirli olduğu için tekillik sorunu da olmamaktadır.

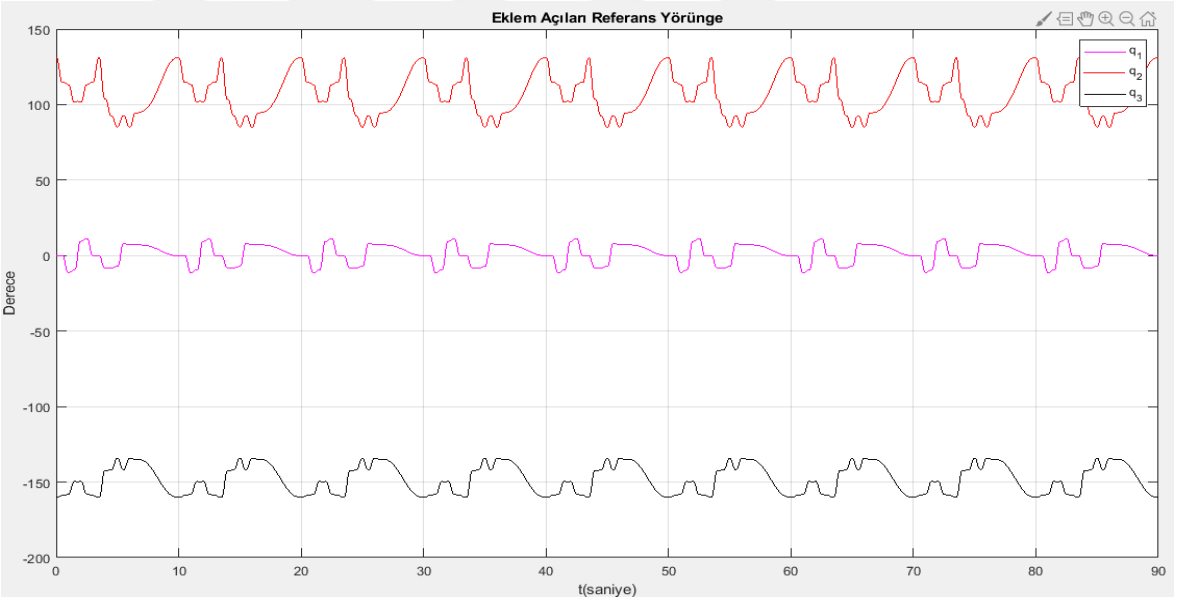
Başlangıç pozisyon olarak X=0.2m Y=0.05m Z=0 seçtik. Aynı görevi 90 saniye içerisinde 9 kez tekrar ettirdik ve her bir sefer için ortalama hata normuna göre katman sayısını güncelledik.



Şekil 4.6 3D Konum Uzayı Yörünge Hedefimiz

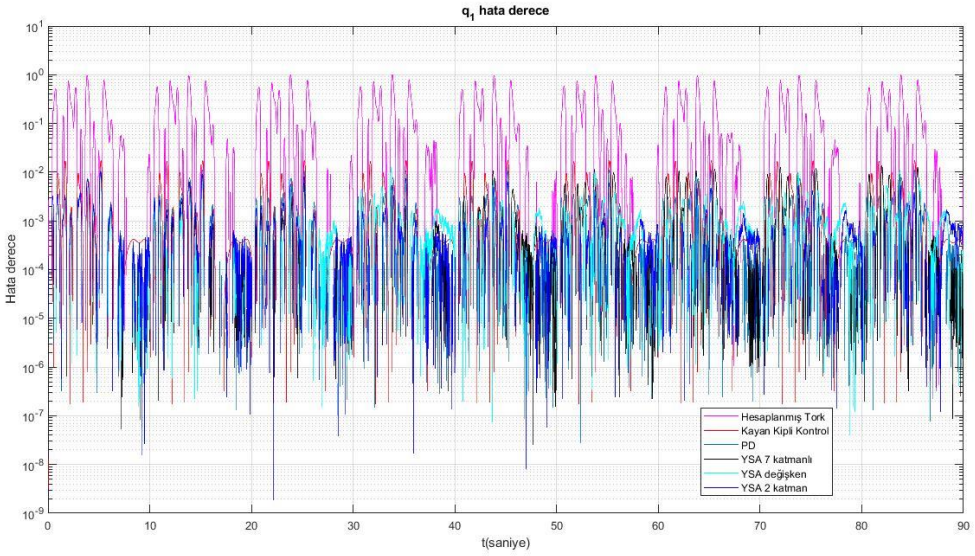


Şekil 4.7 2D Konum Uzayı Yörünge Hedefimiz

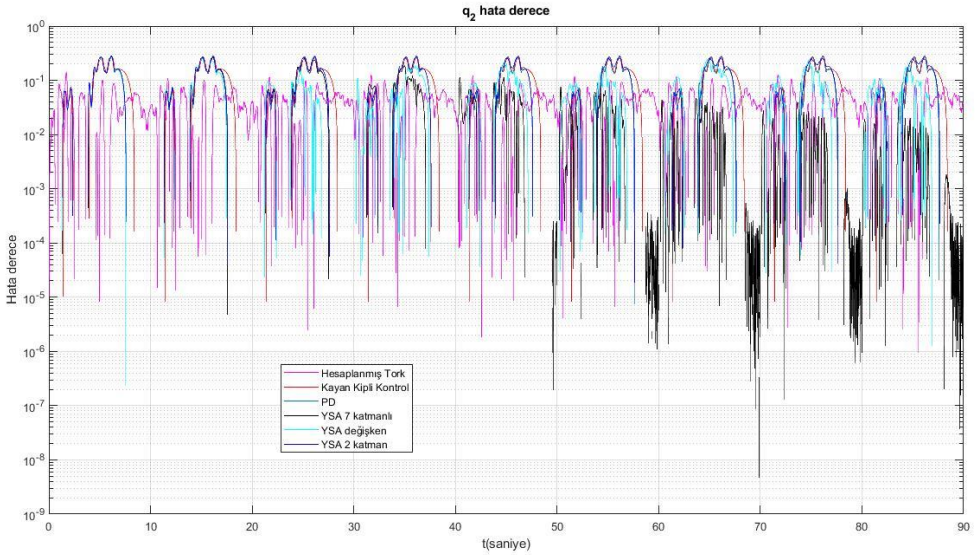


Şekil 4.8 Eklem Açıları Hedef Yörüngemiz

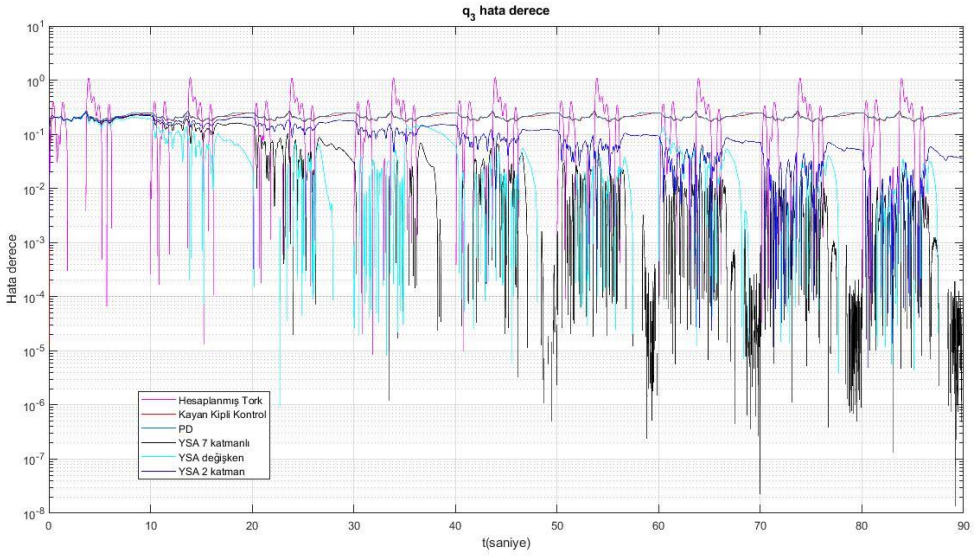
Simülasyon için katman güncelleme sabitlerimizi $E_{\max}=0.25, E_{high}=0.2, E_{low}=0.1, E_{\min}=0.05$ olarak güncelledik ve elde ettiğimiz sonuçlar verilmiştir,



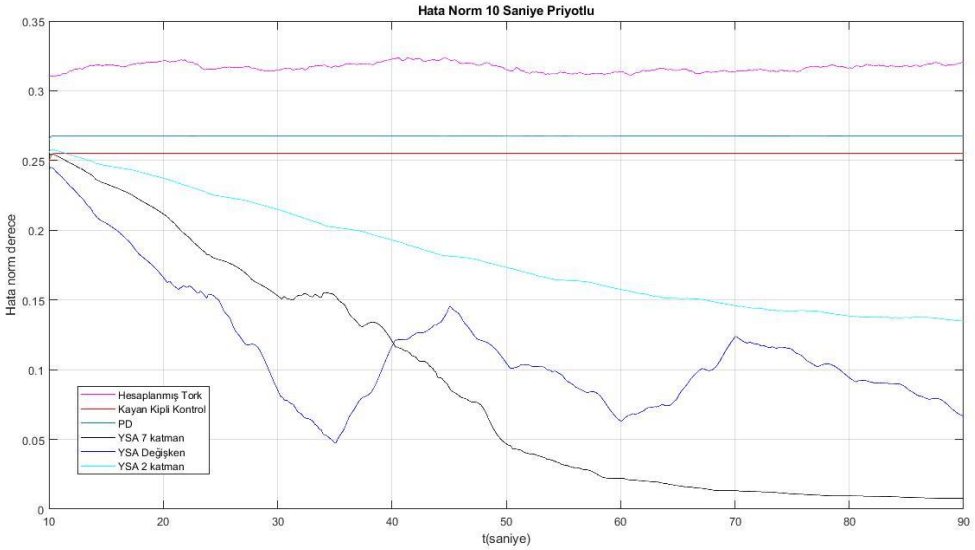
Şekil 4.9 1.Eklem için Hata



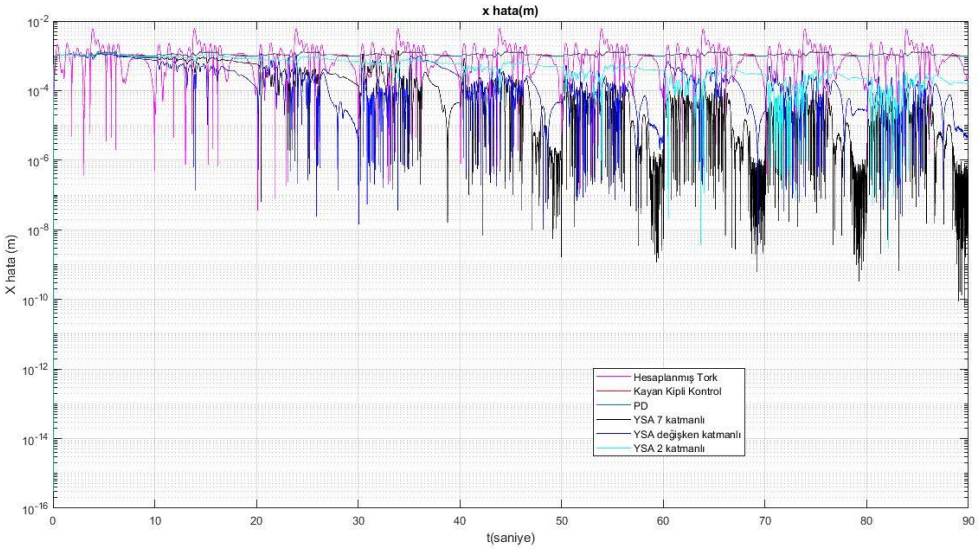
Şekil 4.10 2.Eklem için Hata



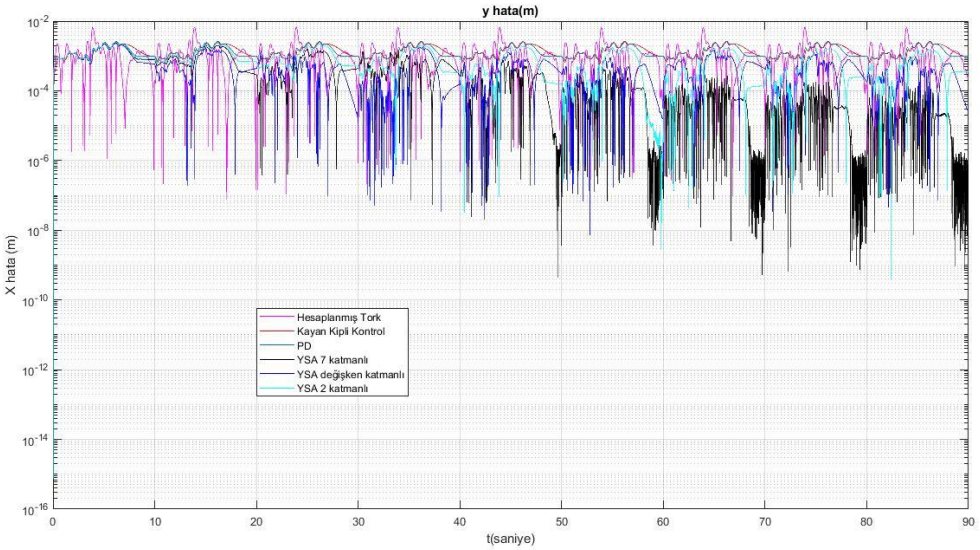
Şekil 4.11 3.Eklem için Hata



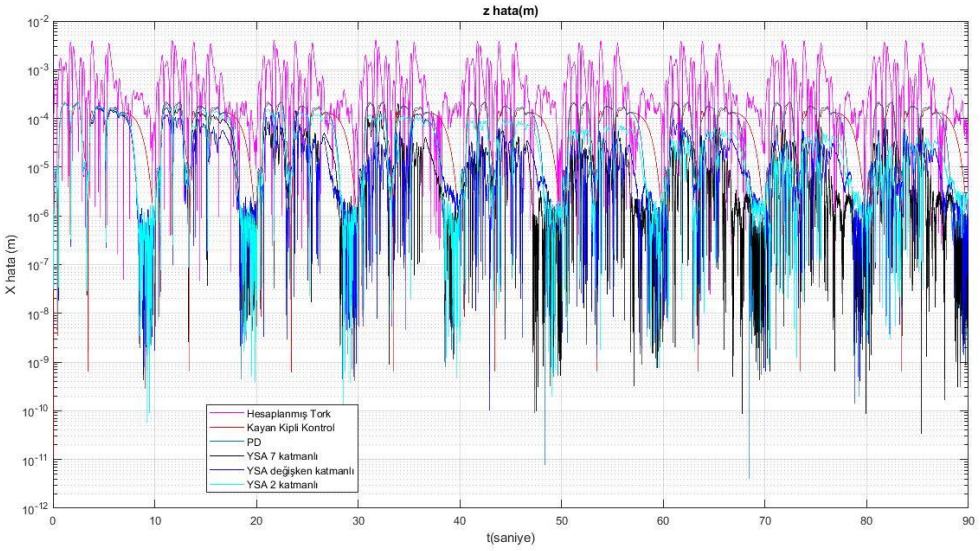
Şekil 4.12 Eklem Açıları için Hataların Ortalama Normu



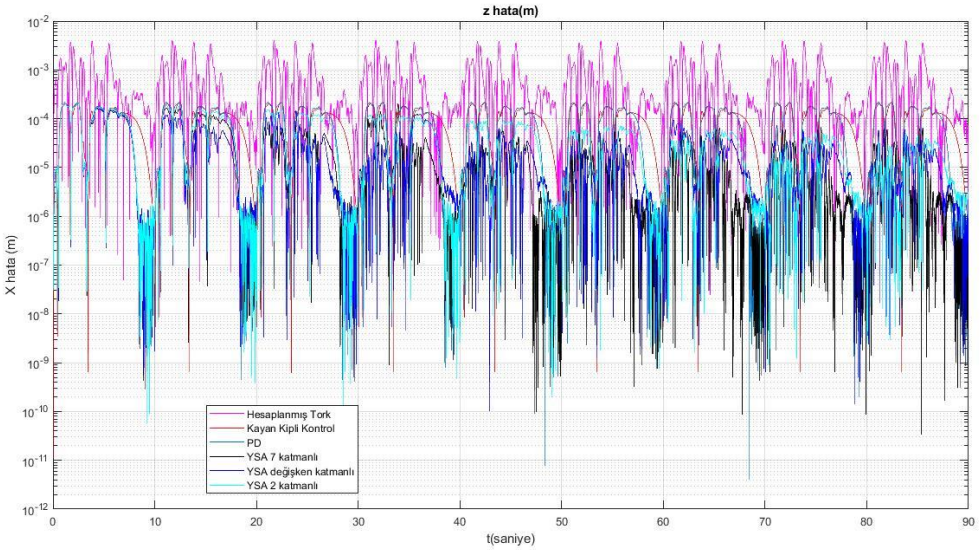
Şekil 4.13 X Koordinat Hata



Şekil 4.14 Y Koordinat Hata

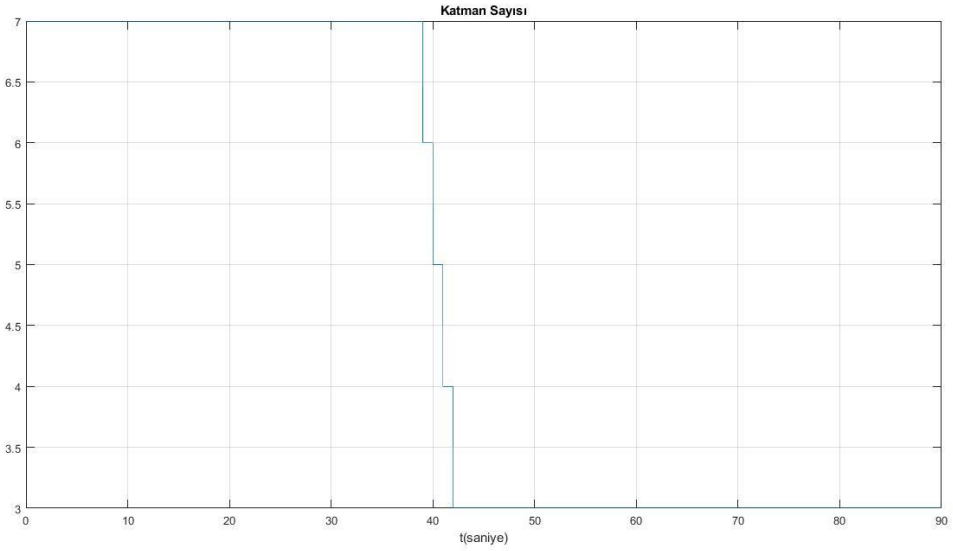


Şekil 4.15 Z Koordinat Hata



Şekil 4.16 Konum Uzayındaki Hatanın Ortalama Normu

Katman sayısının değiştirme algoritması öğrenme sürecinde toplam katmanı 7 de tutmuş ve gereken koşullar sağlanınca katman sayısını kademeli olarak minimuma çekmiştir.



Şekil 4.17 Katman Sayısı

BÖLÜM 5

SONUÇ VE ÖNERİLER

Tüm kontrol algoritmaları eşit şartlarda olabilmeleri için sistemim ana kutup rakamı -10 olarak seçildi. Algoritmamız eşit şartlar altında diğer kontrolcülerle karşılaştırıldığında daha başarılı sonuçlar ortaya koymuştur. Bunun başlıca sebebi kontrolcümüzün modelin içerisindeki doğrusal olmayan yapıyı YSA yardımıyla öğrenerek sistemin bilinmeyenlerini daha iyi içinde eritebilmesidir. Aynı zamanda doğrusal olmadığı için herhangi bir doğru sallaştırmaya da ihtiyaç duymayacağı ve modelin dinamiklerinin öncede hesaplanmasını gerektirmeyeceği için bu konuda da zaman kazandıracaktır.

Tüm katmanların, son katman hariç ReLU çıkışlı olması doğru sallaşma riskini taşımaktaydı. Bunun ana sebebi ReLU'nun sıfırdan büyük olması durumunda girdi çıktı arasında eğimi bir olan doğru gibi davranmasıydı. Yani ikiden fazla ara katmanın harcanan işlem vaktine karşılık iki katmanla aynı performansı verme ihtimali vardı.Buna karşılık ReLU işlem hızı olarak çok hızlı olması ispatlanabilir olması ve negatif çıktı vermeme avantajlarını taşımaktaydı. Bu yaptığımız test gösterdi ki ReLU katmanı sayısının artması sonuca direkt olarak olumlu etki yapmaktadır ve doğru sallaşma sorunu 7 katmana kadar problem oluşturmamaktadır.

Buna ek olarak 1996 yılında Franck Lewis ve Aydın Yeşildirek ortaya konan teorideki 2 katman sayısını artırmanın getirdiği en büyük dezavantajı katman sayısı ile beraber O(n) kadar işlem sayısının artmasıydı. Yapılan testlerimiz gösterdi ki geliştirdiğimiz adaptif ara katman sayısı ayarlama metodu vasıtasıyla simülasyonda geçen sürenin

azaltılması mümkün olabilmektedir (Yaklaşık %30 civarı). Bu da gerçek hayatta bize örnekleme zamanını artırabilme şansı vererek daha başarılı sistem kontrolü olanağı sağlayacaktır. Aynı teorideki aktivasyon fonksiyonlarını değiştirmek ve bunu en matematiksel olarak basit öğrenme metotlarından biri olan Hebbian metoduyla da birleştirmek hesaplama süremizde kısalmalar sağladı.

Algoritmamızın bir diğer özelliği özellikle periyodik işlerde daha fazla fark yaratmasıdır bunun başlıca sebebi öğrenme ve uyum sağlama için az da olsa bir zaman gerekmesidir. Bu yönü SMC, Hesaplanmış Tork ve PD 'ye göre önemli dezavantaj olabilmektedir. Buna ek olarak algoritmamız çok fazla matris hesabı gerektirmektedir bu da düşük güçlü bilgisayarlar veya mikro denetleyiciler için problem olabilmektedir. Bu problemi de gizli katman sayısını ayarlayarak bir nebze aşmış bulunuyoruz. Algoritmamız YSA'nın ve derin öğrenmenin gücünü ve PD kontrolcünün basitliği ile birleştirmiş ve danışmansız, veri seti olmadan kendi kendine öğrenerek oldukça başarılı olmuştur.

Bu araştırmada gelecekte ek olarak yapılabilecek şeylere baktığımızda

- Kontrolcülerimiz dış dünyada robotun üzerinde denene bilinir ve sonuçlar arası kıyas yapıla bilinir.
- Kontrolcümüzün ara katmanlarındaki aktivasyon fonksiyonları Tanh ve Sigmoid ile ekleme çıkarmalar yaparak performans ve örnekleme hızı karşılaştırılabilir.
- Daha kompleks robotlar, otonom robotlar gibi dinamik modelini çıkarmanın çok zor olduğu yerlerde faydalanıla bilinir
- Daha çeşitli durumlar için başka öğrenme algoritmaları denenerek sabit hata düşürülmesi denenebilir veya nöronları budama işlemi yapılarak örnekleme zamanın artması test edile bilinir.

Genel olarak baktığımızda danışmansız YSA'nın robot kontrolü ve başka dinamik sistemler üzerinde kullanımı başarılı sonuçlar vermektedir. Yeni ve modifiye edilmiş algoritmalar ile eski algoritmaların problemleri azaltılarak veya kabul edile bilinir seviyeye çekilerek bu algoritmaların yaygınlaşması ilerde sağlana bilinir.

KAYNAKLAR

[1] Ishiguro, A. Furuhashi , T. ve Okuma, S.Y., Uchikawa, (1992). "A Neural Network Compensator for Uncertainties of Robotics Manipulators", IEEE Transactions on Industrial Electronics, 39: 565-570.

[2] Lewis, F.L. Yesildirek, A. ve Kai, L., (1996). "Multilayer neural-net robot controller with guaranteed tracking performance", IEEE Transactions on Neural Networks, 7: 388-399.

[3] Lewis, F.L. Yeşildirek , A. ve Sarangapani, J., (1998). Neural Network Control of Robot Manipulators and Nonlinear Systems, Bristol, PA, USA: Taylor \& Francis, Inc.

[4] Sun, F. Sun , Z. ve Woo, P.-Y., (2001). "Neural network-based adaptive controller design of robotic manipulators with an observer", IEEE Transactions on Neural Networks, 12: 54-67.

[5] Ye, J., (2008). "Adaptive Control of Nonlinear PID-based Analog Neural Networks for a Nonholonomic Mobile Robot", Neurocomput., 71: 1561-1565.

[6] Velagic, J. Osmic, N. ve Lacevic, B., (2008). "Neural Network Controller for Mobile Robot Motion Control", International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, 2: 2471-2476.

[7] Braganza, D. Dawson, D.M. Walker, I.D. ve Nath , N., (2007). "A Neural Network Controller for Continuum Robots", IEEE Transactions on Robotics, 23: 1270-1277.

[8] Buss, S.R., (2009). Introduction to Inverse Kinematics with Jacobian Transpose,Pseudoinverse and Damped Least Square Methods, University of California.

[9] Murray , R.M.L., Zexiang ve Sastry, S.S., (1994). Robot Dynamics and Control, ed. A Mathematical Introduction to Robotic Manipulation. CRC Press, 171.

[10] Chung , W. Fu, L.-C. ve Hsu, S.-H., (2008). Motion Control, B. Siciliano ve Khatib, O., ed. Springer Handbook of Robotics. Berlin: Springer, Berlin,

Heidelberg, 133-159.

[11] Asif, U. ve Iqbal, J., (2011). A Robotic System with Hybrid Motion Cueing Controller for Ineria Tensor Approximation in Micro-Manipulations, Electrical &Mechanical Engineering: National University of Sciences and Technology.

[12] Perruquetti, W. ve Barbot , J.P., (2002). Sliding Mode Control In Engineering: MARCEL DEKKER.

[13] Rozaimi, G. Yahaya, M.S. Mohd Fua'ad, R. Abd Wahab, I.M.H. ve Zulfatman, (2011). "Performance Comparison between Sliding Mode Control with PID Sliding Surface and PID Controller for an Electro-hydraulic Positioning System", Malasia.

[14] Köse, E. Abacı , K. ve Aksoy, S., (2011). Kayan Kipli Kontrolde Farklı Erişim Alt Yaklaşımlarının Analizleri, 6th International Advanced Technologies Symposium. Elazığ.

[15] Hinton, G.E. Osindero, S. ve Teh, Y.-W., (2006). "A fast learning algorithm for deep belief nets", Neural computation, 18: 1527-1554.

[16] Electronics, P.R., 99:1 Metal Gearmotor 25Dx54L mm HP 12V with 48 CPR Encoder, Las Vegas , Nevada.

EK-A

TEZ KAPSAMINDA KULLANILAN MATEMATİKSEL DEĞERLER

A-1 Robot Sabitleri

Robotun uzunluk ağırlık ve atalet matrisler verilmiştir.

$$l = \begin{bmatrix} l_2 \\ l_{2c} \\ l_3 \\ l_{3c} \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.2143163 \\ 0.45335 \\ 0.2266382 \end{bmatrix} m \tag{7.1}$$

$$m = \begin{bmatrix} m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} 0.2763 \\ 0.2251 \end{bmatrix} kg \tag{7.2}$$

$$I_1 = \begin{bmatrix} 2537.6 & -17.2 & -8.84 \\ -17.2 & 2134.9 & 316.1 \\ -8.84 & 316.1 & 592.7 \end{bmatrix} gcm^2 \tag{7.3}$$

$$I_2 = \begin{bmatrix} 3594 & -21 & -5358 \\ -21 & 37463 & 3 \\ -5358 & 3 & 35141 \end{bmatrix} gcm^2 \tag{7.4}$$

$$I_3 = \begin{bmatrix} 1656 & -39 & 638 \\ -39 & 42446 & 1 \\ 638 & 1 & 41605 \end{bmatrix} gcm^2 \tag{7.5}$$

A-2 Robot Dinamik Matrisleri ve ReLU

H matrisi aşağıda verilmiştir.

$$H=\begin{bmatrix} H_{11} & 0 & 0 \\ 0 & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \tag{7.6}$$

$$H_{11}=I_{1_{yy}}+I_{2_{yy}}+I_{3_{yy}}+m_3(l_2\cos(q_2)+l_{3c}\cos(q_2+q_3))^2+m_2l_{2c}^2\cos^2(q_2) \tag{7.7}$$

$$H_{22}=(I_{2_{xx}}+I_{3_{xx}}-I_{2_{zz}}-I_{3_{zz}})\sin^2(q_1)+I_{2_{zz}}+I_{3_{zz}}+\sin(2q_1)(I_{3_{xz}}+I_{2_{xz}})+m_3(l_2^2+2l_2l_{3c}\cos(q_3)+l_{3c}^2)+l_{2c}^2m_2 \tag{7.8}$$

$$H_{23}=H_{32}=(I_{3_{zz}}-I_{3_{xx}})\cos^2(q_1)+m_3l_{3c}(l_{3c}+l_2\cos(q_3))+I_{3_{xx}}+I_{3_{xz}}\sin(2q_1) \tag{7.9}$$

$$H_{33}=(I_{3_{zz}}-I_{3_{xx}})\cos^2(q_1)+m_3l_{3c}^2+I_{3_{xx}}+I_{3_{xz}}\sin(2q_1) \tag{7.10}$$

V matrisi aşağıda verilmiştir.

$$O_1=\sin(2(q_2+q_3)),O_2=\sin(2q_2+q_3) \tag{7.11}$$

$$C=\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{7.12}$$

$$C_{11}=\frac{-(I_{3c}{}^2m_3O_1+l_2l_{3c}m_3O_2+l_2l_{3c}m_3sin(q_3))\dot{q}_3-(m_3l_2{}^2sin(q_2)+2*m_3l_2l_{3c}O_2+m_2l_{2c}{}^2sin(q_2)+m_3l_{3c}{}^2O_1)\dot{q}_2}{2} \tag{7.13}$$

$$C_{12_a}=\frac{(((I_{3_{zz}}\text{-}I_{3_{xx}})\sin(2q_1)\text{-}2I_{3_{xz}}\cos(2q_1))\dot{q}_3+((I_{2_{zz}}+I_{3_{zz}}\text{-}I_{2_{xx}}\text{-}I_{3_{xx}})\sin(2q_1)\text{-}2\cos(2q_1)(I_{2_{xz}}+I_{3_{xz}}))\dot{q}_2}{2} \tag{7.14}$$

$$C_{12_b}=\frac{m_3l_2{}^2sin(2q_2)+2m_3l_2l_{3c}O_2+m_2l_{2c}{}^2sin(2q_2)+m_3l_{3c}{}^2O_1)\dot{q}_1}{2} \tag{7.15}$$

$$C_{12}=C_{12_a}+C_{12_b} \tag{7.16}$$

$$C_{13}=\frac{(l_{3c}{}^2m_3O_1+l_2l_{3c}m_3O_2+l_2l_{3c}m_3sin(q_3))\dot{q}_1-((-I_{3_{zz}}+I_{3_{xx}})sin(2q_1)+2I_{3_{xz}}cos(2q_1))(\dot{q}_2+\dot{q}_3)}{-2} \tag{7.17}$$

$$C_{21}=-C_{12} \tag{7.18}$$

$$C_{22}=\frac{2\cos(q_1)(I_{2_{xz}}+I_{3_{xz}})+(\text{-}I_{2_{zz}}\text{-}I_{3_{zz}}+I_{2_{xx}}+I_{3_{xx}})\sin(2q_1))\dot{q}_1\text{-}2m_3l_2l_{3c}\sin(q_3)\dot{q}_3}{2} \tag{7.19}$$

$$C_{23}=\frac{((\text{-}I_{3_{zz}}+I_{3_{xx}})\sin(2q_1)+2I_{3_{xz}}\cos(2q_1))\dot{q}_1\text{-}2m_3l_2l_{3c}\sin(q_3)(\dot{q}_2+\dot{q}_3)}{2} \tag{7.20}$$

$$C_{31}=-C_{13} \tag{7.21}$$

$$C_{32}=C_{23}+m_3l_2l_{3c}\sin(q_3)(2\dot{q}_2+\dot{q}_3) \tag{7.22}$$

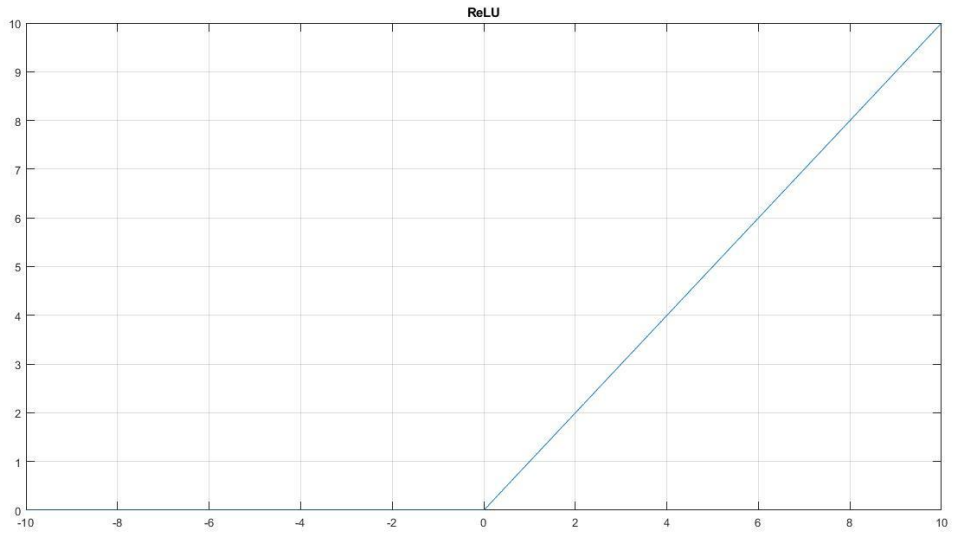
$$C_{33}=\frac{(-I_{3_{zz}}+I_{3_{xx}})\sin(2q_1)+2I_{3_{xz}}\cos(2q_1)}{2} \tag{7.23}$$

$$\left\|H_{\max}\right\|_2 \approx 0.1489\Lambda\left\|C_{\max}\right\|_2 \approx 0.0013 \tag{7.24}$$

$$G=\begin{bmatrix} 0 \\ m_2 g l_{2c}\cos(q_2)+m_3 g(l_2\cos(q_2)+l_3(\cos(q_2+q_3)) \\ m_3 g l_{3c}\cos(q_2+q_3) \end{bmatrix}, g=9.79\,{}^{m}\!/_{s^2} \tag{7.25}$$

ReLU fonksiyonu aşağıda verilmiştir.

$$y=\mathrm{Re}LU(x)=\begin{cases} x>0 \Rightarrow y=x \\ x\le 0 \Rightarrow y=0 \end{cases} \tag{7.26}$$



Şekil A.1 ReLU Fonksiyon Grafiği

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Özgür ALTINIŞIK

Doğum Tarihi ve Yeri : 21 Mayıs 1991, Sakarya

Yabancı Dili : İngilizce

E-posta : ozgural@sabanciuniv.edu

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Mekatronik Mühendisliği	Sabancı Üniversitesi	2014
Lise	Fen Lisesi	Eskişehir Fatih Fen Lisesi	2009

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2017-Halen	Bursa Teknik Üniversitesi	Araştırma Görevlisi
2015-2015	Profen İletişim	Kontrol Mühendisi

YAYINLARI

Bildiri

1. Altınışık, Ö. and Yeşildirek, A. (2018). “R-r-r Robot Control With Adaptive Layer Number Neural-Network + Pd Controller”, Proceedings of IMSMATEC 2018, 10-12 April 2018, İzmir, 361-367.