

T.C.
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ
LİSANSÜSTÜ EĐİTİM ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĐİ ANABİLİM DALI

**MOBİL MANİPÜLATÖRLER İÇİN DERİN ÖĐRENME TABANLI ARIZA TEŐHİS
SİSTEMİ**

YÜKSEK LİSANS TEZİ

ZEKİCAN YILMAZ

TEZ DANIŐMANI

DOĐ. DR. TOLGA YÜKSEL

BİLECİK, 2024
10615501

T.C.
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ
LİSANSÜSTÜ EĐİTİM ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĐİ ANABİLİM DALI

**MOBİL MANİPÜLATÖRLER İÇİN DERİN ÖĐRENME TABANLI ARIZA TEŐHİS
SİSTEMİ**

YÜKSEK LİSANS TEZİ

ZEKİCAN YILMAZ

TEZ DANIŐMANI

DOĐ. DR. TOLGA YÜKSEL

BİLECİK, 2024
10615501

BEYAN

Mobil Manipulatörler İçin Derin Öğrenme Tabanlı Arıza Teşhis Sistemi adlı yüksek lisans/doktora/sanatta yeterlik tezi/dönem projesinin hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın, Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.			
DESTEK ALINMIŞTIR	<input type="checkbox"/>	DESTEK ALINMAMIŞTIR	<input checked="" type="checkbox"/>
Destek alındı ise;			
Destekleyen kurum;			
Desteğin Türü		Proje Numarası	
1- BAP (Bilimsel Araştırma Projesi)			
2- TÜBİTAK			
Diğer;.....			
ETİK KURUL onayı var ise;			
ETİK KURUL karar tarih/sayı:	/.....	

Zekican YILMAZ

Tarih

.....

İmza

.....

ÖN SÖZ

Bu tezi hazırlama sürecinde bana rehberlik eden, bilgi ve tecrübelerini benimle paylaşan değerli tez danışmanım Sayın Doç. Dr. Tolga Yüksel'e en içten teşekkürlerimi sunarım. Bilimsel bakış açısını geliştirmemde ve araştırma konumun derinliklerine dalabilmemde gösterdiği sabır, özveri ve verdiği destek için minnettarım.

Hayatımın her aşamasında bana koşulsuz sevgi ve desteklerini sunan kıymetli Matematik Öğretmeni Annem Gülgün Yılmaz'a ve Matematik Öğretmeni Babam Mehmet Yılmaz'a şükranlarımı sunuyorum. Bu akademik yolculuğumda her zaman yanımda oldular, motivasyonumu artırdılar ve başarıya ulaşmam için gerekli olan her türlü fedakarlığı yaptılar. Onların varlığı, her zaman benim için en büyük güç kaynağı oldu.

Ayrıca, bu çalışmanın gerçekleştirilmesi sürecinde bana olanaklar sunan ve destekleyici bir çalışma ortamı sağlayan ASELSAN-BİTES'e de teşekkür etmek istiyorum. İş yerimdeki deneyimlerim, bu tezin hazırlanmasında önemli bir rol oynadı ve araştırmalarımın katkıda bulundu. Bu süreçte desteklerinden dolayı; Aviyonik Programlar Müdürü Sayın Dr. Ali Berkol'a ve Sistem Test Takım Lideri Sayın İdil Gökçe Demirtaş'a, eski şirketimde birlikte çalıştığım yöneticilerim Sayın Yasin Çıbuk'a ve Sayın Hakan Böcekli'ye de bu yolculukta bana büyük bir güç oldukları için teşekkürler etmek isterim.

Bu tez, saygıdeğer danışmanımın rehberliğinde, ailemin sonsuz sevgisi ve desteğiyle ve çalışma hayatımdan aldığım ilhamla şekillendi. Her birine, bana sağladıkları destek için derin minnettarlığımı ifade etmek istiyorum. Bu başarı, onların sayesinde mümkün oldu.

Zekican YILMAZ

2024

ÖZET

MOBİL MANİPÜLATÖRLER İÇİN DERİN ÖĞRENME TABANLI ARIZA TEŞHİS SİSTEMİ

Mobil manipülatörler, birçok endüstriyel alanda kritik görevler üstlenerek, üretimden lojistiğe, sağlık hizmetlerinden afet müdahalelerine kadar geniş bir yelpazede etkinlik göstermektedirler. Dünya çapında endüstrilerin genişlemesiyle birlikte, bu manipülatörlerin büyümesi ve güvenilir bir şekilde çalışmalarının sağlanması giderek daha fazla önem kazanmaktadır.

Bu çalışmada, mobil manipülatörler için arıza teşhisi yapmak üzere derin öğrenme teknikleri temelinde geliştirilen yeni bir metodolojiyi ele almaktadır. Geleneksel yöntemlerin karmaşık kurallar ve uzmanlık bilgisi gerektirmesine karşın, bu çalışma, manipülatörün her bir eklemi üzerinde modellemeler yaparak ve bu eklemlere hareketler uygulayarak veri toplamıştır. Toplanan verilere eklem arızaları eklenmiş ve bu arızaların tanımlanmasında derin öğrenme yöntemleri, özellikle de evrişimli sinir ağları kullanılmıştır. Dokuz farklı eklem üzerinde çeşitli bozulma oranları detaylı bir şekilde incelenmiştir. Mobil robot manipülatörün bir bölgeden diğerine hareketini sağlamak için farklı eklem hareketleri kullanılarak beş ayrı test gerçekleştirilmiştir. Her bir eklem için beş farklı senaryo oluşturularak, bu senaryoların her birinde değişen oranlarda bozulmalar uygulanmıştır, toplamda 255 farklı senaryo değerlendirilmiştir. Elde edilen bu veri setleri, Matlab'ın Toolbox'ı olan Neural Network (NN) ve Uzun-Kısa Süreli Bellek (LSTM) algoritmaları kullanılarak işlenmiş ve bu yöntemlerin arıza tespitindeki etkinliği değerlendirilmiştir. Normal çalışma durumları ile arızalı durumları karşılaştırıldığında, karşılaştırılmıştır

Kuka Youbot'un dinamikleri, bu arıza teşhis çalışmasında kullanılmış olup, eklemlerden elde edilen veriler Yapay Sinir Ağı (YSA) aracılığıyla işlenerek karar verme sürecinde kullanılmıştı, simülasyon aşamasında ise CoppeliaSim kullanılmıştır. Bu yaklaşım, sürdürülebilirlik ve gelişim açısından geleneksel yöntemlere göre daha etkin bir alternatif sunmaktadır.

Anahtar Kelimeler: Mobil Robot Manipülatör, Arıza Tespit Sistemi, Yapay Sinir Ağı.

ABSTRACT

DEEP LEARNING-BASED FAULT DIAGNOSIS SYSTEM FOR MOBILE MANIPULATORS

Mobile manipulators play a crucial role in various industrial fields, ranging from production to logistics, healthcare services to disaster response, demonstrating their effectiveness across a broad spectrum. As industries expand globally, the growth of these manipulators and their reliable operation are becoming increasingly significant.

This study introduces a novel methodology based on deep learning techniques for fault diagnosis in mobile manipulators. Contrasting traditional methods that require complex rules and expert knowledge, this research has conducted modeling on each joint of the manipulator and collected data through the application of movements to these joints. Faults were added to the gathered data, and deep learning methods, particularly convolutional neural networks, were utilized for fault identification. Various degradation rates on nine different joints were meticulously examined. To facilitate movement of the mobile robot manipulator from one area to another, five separate tests using distinct joint movements were performed. Five different scenarios for each joint were created, applying varying rates of faults in each scenario, culminating in a total of 255 distinct scenarios evaluated. These datasets were processed using Neural Network (NN) and Long Short-Term Memory (LSTM) algorithms available in Matlab's Toolbox, assessing the effectiveness of these methods in fault detection. The study compared normal operational data with faulty conditions.

The dynamics of the Kuka Youbot were employed in this fault diagnosis study, with data obtained from the joints processed through an Artificial Neural Network (ANN) for decision-making, while CoppeliaSim was used during the simulation phase. This approach offers a more efficient alternative to traditional methods, enhancing sustainability and development.

Keywords: Mobile Robot Manipulator, Fault Detection System, Artificial Neural Network

İÇİNDEKİLER

	Sayfa
ÖN SÖZ	i
ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	vii
KISALTMALAR VE SİMGELER LİSTESİ	ix
1. GİRİŞ	1
1.1. Literatür Özeti	14
2. MATERYAL.....	16
2.1. Mobil Robot Manipulatörlerin Çalışma Prensibi.....	16
2.2. Arıza Analizi Temelleri	25
2.3. Yapay Sinir Ağları ve Öğrenme Algoritmaları	26
2.3.1. NNTOOL	29
2.3.2. Uzun- Kısa Süreli Bellek LSTM	30
3. METOT	31
3.1. Önerilen Arıza Tespit Sistemi	31
3.2. CoppeliaSim Simülasyon Ortamı.....	33
3.2.1. CoppeliaSim ve Matlab Haberleşmesi.....	34
3.3. Sistem YSA Giriş Sinyallerinin Çıkarımı.....	35
3.4. Veri Setlerinin Oluşturulması.....	37
3.5. Neural Network Toolbox.....	44

3.6. Uzun-Kısa Süreli Bellek (LSTM)	55
3.7. NN ve LSTM ile Yakınsama Karşılaştırılması.....	67
4. SONUÇLAR VE GELECEK ÇALIŞMALAR	79
KAYNAKÇA.....	81

TABLULAR LİSTESİ

	Sayfa
Tablo 2.1. D-H parametrelerinin deęerleri	19
Tablo 2.2. Tekerlek Parametreleri	24
Tablo 3.1. Eklemlerin J ve B Karakteristikleri	40
Tablo 3.2. NNMSE ve LSTM MSE deęerleri	72

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1.1. Taslak tasarım	2
Şekil 1.2. Otonom sınıfa giren mobil robot manipülatör örneği	3
Şekil 1.3. Örnek Model.....	5
Şekil 1.4. Mobil Manipülatör Kinematik Modelleme	6
Şekil 1.5. Örnek PID Kontrolcü Devresi	7
Şekil 1.6. PID Döngüsü	8
Şekil 1.7. Montajlama Örneği	9
Şekil 1.8. Mobil Manipülatörlerin Çeşitli Endüstrilerde Kullanım Alanı	11
Şekil 2.1. Mobil Robot Manipülatör 5 Serbestlik Derecesindeki Eklemleri ve Tekerleklerin Dönme Yönleri	16
Şekil 2.2. KUKA youBot Çerçeveleri	18
Şekil 2.3. KUKA youBot Parametreleri	18
Şekil 2.4. KUKA youBot Platformu ve İlişkili Koordinat Çerçeveleri.....	21
Şekil 2.5. Mecanum Tekerlek.....	23
Şekil 2.6. Mecanum Tekerleğin Koordinat Sisteminde Gösterimi.....	23
Şekil 2.7. Yapay Sinir Hücresi	28
Şekil 3.1. Önerilen Arıza Tespit Sistemi	32
Şekil 3.2. CoppeliaSim Senaryo Simüle Etme Ortamı	35
Şekil 3.3. YSA Giriş Sinyalleri Çıkarımı için Simüle Edilen Görüntü	35
Şekil 3.4. Matlab'a Koordinatların Aktarıldığı Görüntü	36
Şekil 3.5. Mobil Platformdaki Hareket Değişikliği.....	37
Şekil 3.6. PID Kontrolcüsü Örnek Model.....	39
Şekil 3.7. PID'ye Otomatik Ayar'a Giren Verilen Grafiği.....	42
Şekil 3.8. NNT Kullanıcı Arayüzü	44
Şekil 3.9. Create Network or Data Arayüzünün Bir Görseli.....	45
Şekil 3.10. Bölüm 3.4'te oluşturulan veri setlerinden yararlanarak duruş kestiriminde kullanılan yapay sinir ağı yapısını 1. eklem için görsel olarak sergilemektedir	46
Şekil 3.11. NN'nin 1. Eklem için Yakınsaması	47
Şekil 3.12. NN'in 2. Eklem için Yakınsaması.....	48

Şekil 3.13. NN'nin 3. Eklem için Yakınsaması.....	49
Şekil 3.14. NN'nin 4. Eklem için Yakınsaması.....	50
Şekil 3.15. NN'nin 5. Eklem için Yakınsaması.....	51
Şekil 3.16. NN'nin 6. Eklem için Yakınsaması.....	52
Şekil 3.17. NN'nin 7. Eklem için Yakınsaması.....	53
Şekil 3.18. NN'nin 8. Eklem için Yakınsaması.....	54
Şekil 3.19. NN'nin 9. Eklem için Yakınsaması.....	55
Şekil 3.20. LSTM Hücresinin Yapısal Şeması	57
Şekil 3.21. 1. Eklem için LSTM Performansı.....	58
Şekil 3.22. LSTM'nin 1. Eklem için Yakınsaması	59
Şekil 3.23. LSTM'nin 2. Eklem için Yakınsaması	60
Şekil 3.24. LSTM'nin 3. Eklem için Yakınsaması	61
Şekil 3.25. LSTM'nin 4. Eklem için Yakınsaması	62
Şekil 3.26. LSTM'nin 5. Eklem için Yakınsaması	63
Şekil 3.27 LSTM'nin 6. Eklem için Yakınsaması	64
Şekil 3.28. LSTM'nin 7. Eklem için Yakınsaması	65
Şekil 3.29. LSTM'nin 8. Eklem için Yakınsaması	66
Şekil 3.30. LSTM'nin 9. Eklem için Yakınsaması	67
Şekil 3.31. 1. Eklem için NN ve LSTM Karşılaştırması	69
Şekil 3.32. 4. Eklem için NN ve LSTM Karşılaştırılması	71
Şekil 3.33 1. Eklem için NN ve LSTM Yakınsaması	72
Şekil 3.34 2. Eklem için NN ve LSTM Yakınsaması	73
Şekil 3.35 3. Eklem için NN ve LSTM Yakınsaması	73
Şekil 3.36 4. Eklem için NN ve LSTM Yakınsaması	74
Şekil 3.37 5. Eklem için NN ve LSTM Yakınsaması	75
Şekil 3.38 6. Eklem için NN ve LSTM Yakınsaması	75
Şekil 3.39 7. Eklem için NN ve LSTM Yakınsaması	76
Şekil 3.40 8. Eklem için NN ve LSTM Yakınsaması	77
Şekil 3.41 9. Eklem için NN ve LSTM Yakınsaması	77

KISALTMALAR VE SİMGELER LİSTESİ

- ANFIS** : Adaptive Neuro Fuzzy Inference System
- ANN** : Yapay Sinir Ağı
- API** : Aynı zamanda Coppeliasim Uygulama Programlama Arayüzü
- ATS** : Arıza Tespit Sistemi
- CNN** : Evrimsel Sinir Ağları
- D-H** : Denavit-Hartenberg
- D.O.F.** : Degree of Freedom (Serbestlik Derecesi)
- FDI** : Fault Detection and Isolation (Arıza Tespit ve İzolasyonu)
- FL** : Fuzzy Logic (Bulanık Mantık)
- LOGSIG** : Logaritmik-sigmoid
- LSTM** : Little-Short Time Memory (Uzun-Kısa Süreli Bellek)
- MSE** : Mean Square Error (Hataların Ortalama Karesi)
- NN** : Neural Network (Yapay Sinir)
- NNT** : Neural Network Toolbox
- PID** : Proportional Integration Derivative (Oransal-İntegral-Türev)
- RMSE** : Root Mean Square Error (Hataların Ortalama Kare Kökü)
- SSO** : Sosyal Örümcek Optimizasyonu
- TCP/IP** : Transmission Control Protocol
- YSA** : Yapay Sinir Ağı
- net_j** : Nörona Gelen Girdilerin Toplamı
- a** : Link uzunluğu
- B** : Viskoz Sönüm Katsayısı
- d** : Linkin Önceki Ekleme Olan Mesafe
- $F(q')$** : Sistemdeki Belirsizlik
- F_p** : İşlem Hedefi Çerçevesi

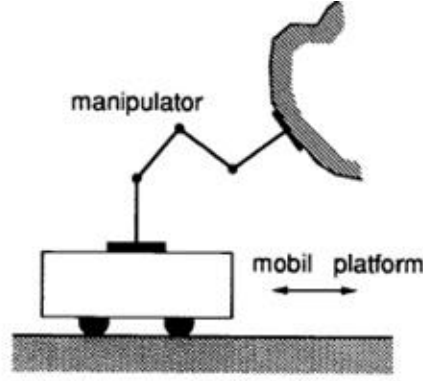
Fr	: Mevcut Robot Çerçevesi
Ft	: Hedef Robot Çerçevesi
G	: Yerçekimi Vektörü
G^-	: Yerçekimi Vektörü
I	: Birim Matrisi
i	: Tekerlek İndeksi
J	: Eylemsizlik Momenti
J_P	: Hız ve Yönün Mobil Platformun Tabanıyla İlişkilendirme Matrisi
l_i	: Tekerleğin Robotun Merkezine Olan Mesafesi
l_{ix}	: Tekerleğin Robotun Merkezine olan x Bileşeni
l_{iy}	: Tekerleğin Robotun Merkezine olan y Bileşeni
o_j	: Nöronun Eşik Değeri
P	: Platform
$q \dot{ }_P$: Mobil Platformun Hız Vektörü
R	: Rotasyon Matrisi
R^n	: Tüm n Boyutlu Noktaların Kümesi
$r\xi$: Duruş Tahmini
T	: Küresel Eylemsizlik
$t\xi$: Hedeflenen Konum
u_P	: Motor Girdileri
W_{ij}	: Girilen Veriler Ağırlıkları
x_b	: Konum Değişkeni
y_b	: Yönelim Değişkeni
z_0	: Baz Koordinat Konumu
α	: İki Ardışık Eksen Arasındaki Açısı
α_i	: Tekerleğin Yere Göre Montaj Açısı
β_i	: Tekerleğin Eğim Açısı

- γ_i : Tekerleğin Dönüş Yönü
 θ : Eklem Açısı
 $\tau_d(t)$: Bilinmeyen Gürültü
 ψ : Arıza
 ω : Açısal Hız

1. GİRİŞ

Mobil manipülatörler, son yıllarda hem endüstriyel hem de akademik çevrelerde önemli bir ilgi alanı oluşturmuştur. Bu dinamik ve çok yönlü robotik sistemler, otomasyon, yapay zekâ ve mekatronik alanlarında önemli gelişmelerle birlikte ortaya çıkmış ve çeşitli sektörlerde kullanımı giderek artmıştır. Son yıllarda mobil manipülatörler hem endüstriyel hem de akademik çevrelerde önemli bir ilgi alanı oluşturmuştur. Otomasyon, yapay zeka ve mekatronik alanlarındaki önemli gelişmelerle birlikte ortaya çıkan bu dinamik ve çok yönlü robotik sistemler, çeşitli sektörlerde giderek artan bir kullanıma sahiptir. Ek olarak, gelişmiş sektörlerin daha da ilerlemesinde önemli bir rol oynayan insansız sistemler, sanayi, endüstri, havacılık, otomasyon sektörlerinin yanı sıra sosyal yaşamda ve hobilerde de hızla yerini almaktadır.

Mobil manipülatörler, robotik ve otomasyon dünyasında devrim niteliğinde bir gelişmedir. Bu sistemler, hareket kabiliyeti ve manipülasyon yeteneklerini bir araya getirerek, çeşitli endüstriyel ve araştırma uygulamalarında kullanılmaktadır. Mobil manipülatörlerin tarihçesini incelenirken, bu teknolojinin temellerinin 20. yüzyılın ortalarına kadar uzandığını gözlemlenmektedir. Erken dönem çalışmalarında modellenmesi, simülasyonu, manipüle edilebilirliği ve hareket üretimi üzerine yoğunlaşmıştır; (Yamamoto ve Yun, 1994: 1326)' de belirtildiği gibi hareket planı için manipülatörün ölçülen eklem konumu bilgisini kullanılması, gibi basit mekanik kollar ve hareketli platformlar olarak tasarlanan bu sistemler, robotik alanında önemli bir dönüm noktası olarak işaretlenmiştir ve Şekil 1.1.'de çalışma ile ilgili taslak tasarım görülmektedir.



Şekil 1.1. Taslak tasarım

Kaynak: (Yamamoto & Yun, 1994: 1327)

Bu çalışmalar ile robotik sistemlerde hareketlilik ve manipülasyonun entegrasyonunun temeli atılmıştır. Mobil manipulatörler kavramı, hareketli bir robot tabanı ve üzerine monte edilmiş bir manipulatör kolunun birleşiminden oluşan sistemler olarak tanıtılmış ve hareketlilik ile manipülasyonun birleştirilmesiyle robot uygulamalarının sınırlarını genişletme aracı olarak sunulmuştur. Zaman içinde yapay zekâ, senyör teknolojileri ve mekatronik alanlarındaki ilerlemelerle daha karmaşık ve yetenekli hale gelmiştir ve otonom robotların doğuşu da bu şekilde başlamıştır. Otonom robotların; serbestçe engelsiz bir şekilde navigasyon yapabilme potansiyeline ve çevresindeki hareket alanında bulunan herhangi bir engeli aşma zekasına sahipse, bu robot zeki veya otonom olarak kabul edileceği (Raj ve Kos, 2022: 1)'de belirtilse de aslında engel ve yol bilgisi dışında kullanımına yönelik arıza analizi yapan robotlar da otonom robot sınıfına girmektedir. Şekil 1.2.'de de otonom sınıfa giren mobil robot manipulatör örneği gösterilmektedir.



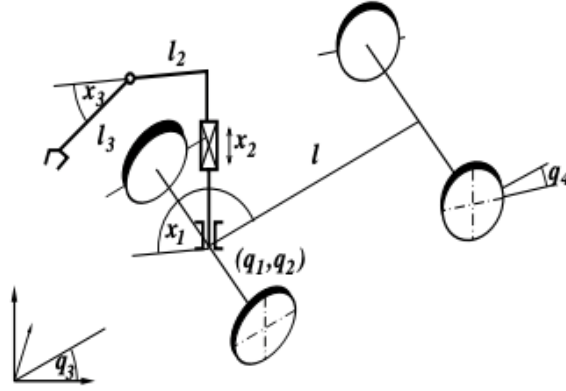
Şekil 1.2. Otonom sınıfa giren mobil robot manipulatör örneği

Mobil manipulatörlerin etkin ve kesintisiz bir şekilde çalışabilmesi, endüstriyel operasyonların sürekliliği için büyük önem arz etmektedir. Bu, özellikle zamanın hayati önem taşıdığı ve hata payının minimumda olması gereken acil durum müdahalelerinde daha da önem kazanmaktadır ve (Yüksel ve Sezgin, 2010: 125)'de de belirtildiği gibi; bir ürün hattındaki arızalı bir robot, esnek bir otomasyon sistemi olsa bile tüm bağlantılı hatlarda bir duraklamaya neden olur. Bu sebeplerden herhangi bir arızanın ve duraklamanın olmaması için eklem hareketlerinin analizi üzerinden duraklama önlenmeye çalışılır.

Eklemler, arızanın belirlenmesi ve analiz edilmesi için kritik bir rol oynamaktadır, çoğu çalışmada bu konu araştırılmıştır; eklem tork kontrolü için gömülü olan eklem tork sensörlerini kullanarak insan-robot çarpışma tespiti üzerine birçok çalışma yapılmıştır (Lee ve Song, 2016: 12). Bu, eklemlerin dinamiklerinin ve çalışma durumlarının detaylı bir şekilde modellenmesini ve analiz edilmesini gerektirir. Oluşturulan model, eklemlerin çeşitli hareketlerini simüle eder ve bu hareketler sırasında meydana gelen değişiklikleri ve olası arızaları belirlemek için kullanılır. Robot manipulatörlerinin güvenilir ve emniyetli çalışmasını sağlamak için, hata teşhisi ve hata toleranslı kontrollerde eklem analizi esastır. Birçok çalışma, robot manipulatörlerinde hata teşhisi ve hata

toleranslı kontrol için eklem analizinin önemini vurgulamıştır, (Uzuner, vd., 2020: 393)'de yapılan çalışmada; beş serbestlik dereceli (DOF) bir seri robot manipülatörünün ters kinematik problemi, sanal eklem yaklaşımı adı verilen bir yöntem aracılığıyla çözüldüğünü belirtmiştir. Sanal eklem yaklaşımı, gerçekteki eklemlerin matematiksel olarak sanal platforma entegre olması ve eklem analizinin yapılması için kullanılır. Bu yaklaşımın bir başka avantajı da sistemlerin gerçek zamanlı olarak monitör edilmesine ve gerekli durumlarda hızlı bir şekilde müdahale edilmesine olanak tanınmasıdır. Bu sayede, arızaların erken teşhisi ve onarımı mümkün hale gelir, bu da operasyonların kesintisiz bir şekilde sürdürülmesine ve olası kayıpların minimize edilmesine katkı sağlar. Bu kayıpların minimize edilmesinde derin öğrenme etkili olmaktadır, özellikle robotlardaki hataların tespiti ve teşhisi için geleneksel yöntemlerin yetersiz kaldığı durumlarda derin öğrenme teknikleri iyi bir potansiyele sahiptir.

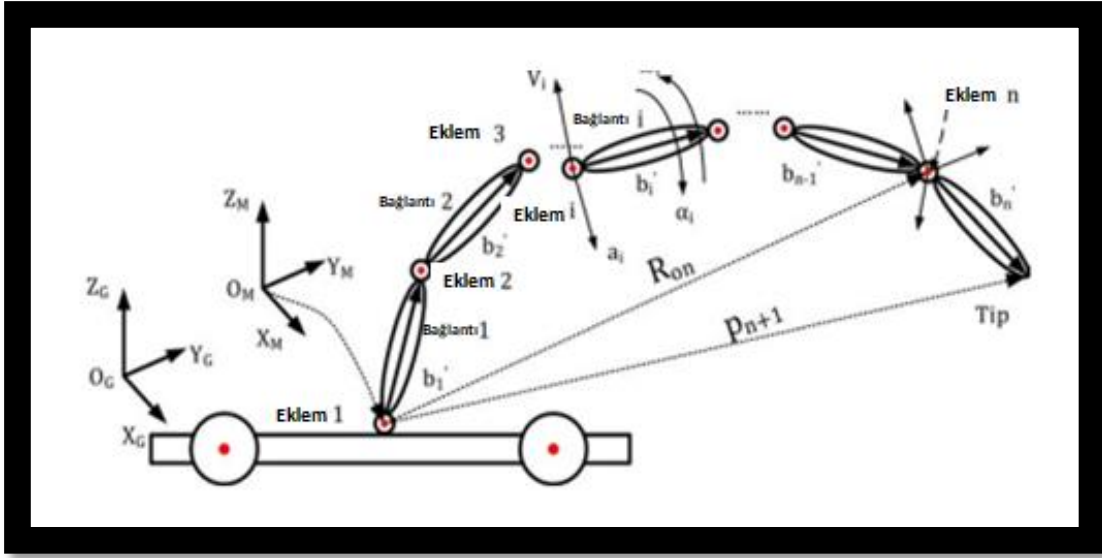
Mobil manipülatörlerle ilişkili arızalar birkaç türe ayrılır. Bunlar kinematik arızalar, dinamik arızalar ve kontrol arızalarını içerir. Kinematik arızalar, manipülatör kolunun ve mobil platformun hareketi ve konumlandırılmasıyla ilgili sorunlara işaret eder. Dinamik arızalar, manipülasyon görevleri sırasında deneyimlenen kuvvetler ve torklar ile mobil manipülatörün stabilitesi ve dengesiyle ilgilidir, Y. Kang ve diğerleri dinamik arızaları şu şekilde açıklamıştır; Aktüatör arızaları (tekerleklerde veya eklemlerde olsun), mobil manipülatörlerin hareketi sırasında ani bir şekilde meydana gelir. Torkun, bir veya daha fazla eklem motorlarına aniden verilmediği durumlarda başarısız olan aktüatörler, robot bileşenlerine zarar verme olasılığı ile sistemi istikrarsızlaştırabilir (Kang, vd., 2012: 266). Kontrol arızaları, mobil manipülatörün hareketini ve etkileşimini yönetmek için kullanılan algoritmalar ve metodolojilerle ilgili sorunları kapsar. Yu ve Cheng tarafından yapılan çalışma, mobil manipülatör sistemlerinin dinamiklerini derinlemesine inceler ve dinamik hatalarla ilişkili zorlukları aydınlatır. Yazarlar, mobil manipülatörlerin dinamik modellenmesini anlamının önemini vurgularlar, bu da kuvvetler, torklar ve genel sistem stabilitesiyle ilgili sorunları ele almak için kritik öneme sahiptir (Yu ve Cheng, 2002: 512). Arızaların matematiksel modellenmesinin, analiz edilmesi açısından önemlidir Şekil 1.3.'de örnek model gösterilmiştir.



Şekil 1.3. Örnek Model

Kaynak: (Tchon ve Jakubiak, 2005: 46)

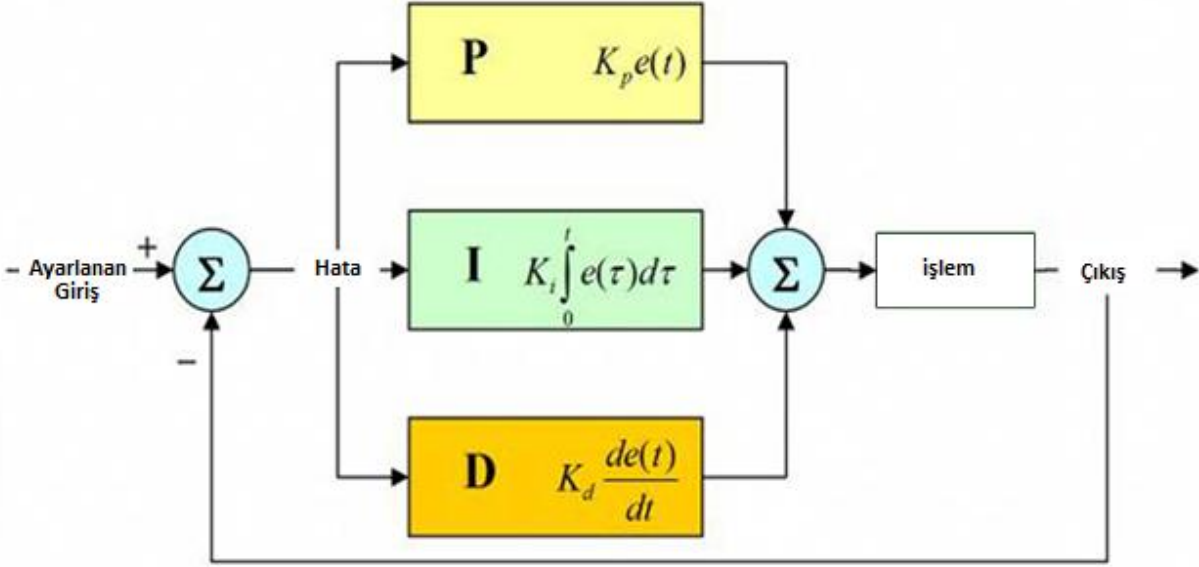
Mobil manipülatörler, kinematik, dinamik ve kontrol hataları gibi çeşitli türde arızalar sergilemektedir. Bu hataları anlamak ve ele almak, mobil manipülatörlerin çeşitli operasyonel senaryolarda performansını ve güvenilirliğini artırmak için hayati öneme sahiptir. Bu sebepten dolayı modellenmesi ve model üzerinden analiz yapılması gerekmektedir çünkü eğer arıza tespit edilirse müdahale edilmesi ve analiz edilmesi bu sebepten dolayı daha basite indirgenmiş olur. Sadece eklemlerdeki arızalar için değil aynı zamanda da tekerlekte oluşan arızalar için de optimizasyon yapılması gerekmektedir. Buna ek olarak da Song ve diğerlerinin yaptığı çalışma, tekerlekli bir manipülatör için mobil platformun optimizasyonuna odaklanmaktadır ve özellikle reaksiyon kuvvetleri ve momentlere karşı devrilme stabilitesini göz önünde bulundurarak, bu durumun kinematik hatalarla ilişkili olduğunu vurgulamaktadır (Song, vd., 2016: 061007-2). Şekil 1.4'te mobil manipülatör kinematik modellemesi gösterilmiştir.



Şekil 1.4. Mobil Manipülator Kinematik Modelleme

Kaynak: (Song, vd., 2016: 061007-5).

Modellemeden sonra en önemli konu ise kontrol modelinin de gerçeğe uygun olarak yapılmış olmasıdır, bunun için de Oransal-İntegral-Türev (PID) kontrolörü, sistem davranışını istenilen referans değere yaklaştırmak için kullanılır. Bu kontrolör, hatayı (istenilen değer ile mevcut değer arasındaki farkı) analiz eder ve buna göre düzeltici bir sinyal üretir. PID modelleme, genellikle sistemlerin istikrarını sağlamak, hızını kontrol etmek ve istenilen konuma ulaşmasını sağlamak için kullanılır. Örnek PID kontrol devresi Şekil 1.5.'te gösterilmiştir. Çınar ve diğerlerinin yaptığı çalışmada da belirtildiği üzere; PID kontrolör parametreleri üretilen modele göre Zeigler–Nichols yöntemi kullanılarak belirlenmiştir (Çınar, vd., 2020: 1628).

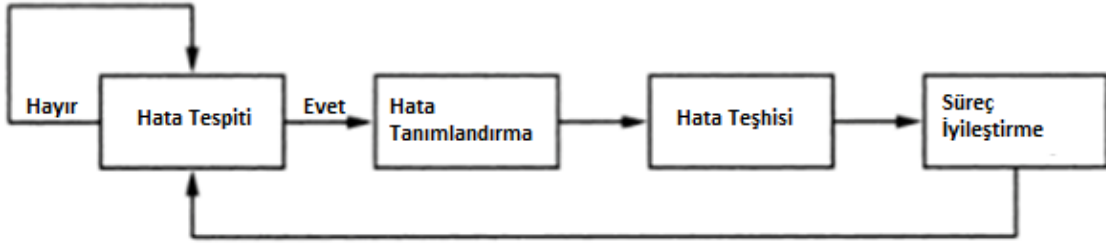


Şekil 1.5. Örnek PID Kontrolcü Devresi

Zeigler–Nichols yöntemi, PID modellemenin kontrol parametrelerinin belirlenmesinde kullanılan yaygın bir yöntemdir. Bu yöntem, kontrolör parametrelerinin belirlenmesi için bir dizi deney ve analiz adımını içerir. İlk olarak, sistem kapalı bir döngüde çalıştırılır ve bu sırada kontrolör devre dışı bırakılır. Daha sonra, sistemde bir adım girişi uygulanır ve sistem cevabı kaydedilir. Bu cevaba göre, belirli kriterlere göre kontrolör parametreleri belirlenir. Zeigler-Nichols yöntemi, sistem cevabının zaman domainindeki karakteristiklerine dayalı olarak kontrolör parametrelerini belirler. Bu karakteristikler genellikle denge zamanı, taşma miktarı ve aşırı sönümlenme gibi performans ölçütlerini içerir. Bu ölçütler, kontrol sisteminin istenilen performansı sağlaması için belirlenir. Bu kontrolör, robotun hareketlerini düzenlemek ve istenen yörüngeye veya pozisyona ulaşmasını sağlamak için kullanılır. Eğer robotun hareketlerinde herhangi bir sapma veya beklenmeyen davranış tespit edilirse, bu durum bir arızanın göstergesi olur. PID kontrolörü bu saptamaları düzeltmek için gerekli ayarlamaları yapar, böylece robotun istikrarını ve doğru çalışmasını sağlar. PID için veriler alınırken; İlk adımda, robotun hareketlerini ve konumunu belirlemek için inverse kinematik yöntemleri kullanılıyor. Bu yöntem, robotun uzuvlarının son noktalarının (end-effectors) konum ve yönelimlerini, eklem açıları gibi girdi parametrelerine dönüştürmek için matematiksel formüller ve algoritmalar içerir. Bu süreç, robotun mevcut durumunu anlamak ve potansiyel hataları belirlemek için kritik öneme sahiptir.

Eklemler haricinde, mobil manipulatörlerde mobil bölüm için de PID kontrolcüsünün tanımlanması gerekmektedir. Literatürde tekerlekli mobil robotlar için PID kontrolünün kullanımını gösteren Khan ve diğerlerinin yaptığı doğadan ilham alan sosyal örümcek algoritması tabanlı bir PID kontrol cihazı kullanılması örnek olur; Daha iyi kontrol performansı için, integral kazancı, diferansiyel kazancı ve oransal kazanç, PID kontrol cihazı tarafından ayarlanmaktadır. Araç yönü değiştirildiğinde robot hızı değişir, bu durumu önlemek için Sosyal Örümcek Optimizasyonu (SSO) algoritması PID kontrol cihazlarında kullanılmaktadır (Khan, vd. , 2023: 1202).

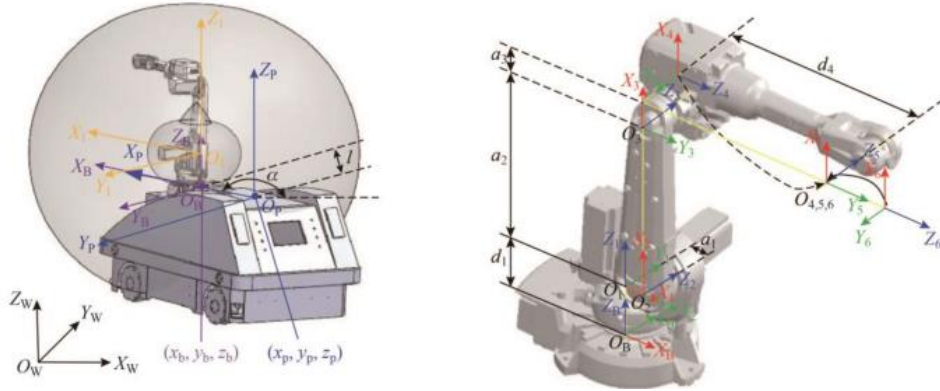
PID kontrolünün hata analizinde etkisi; Chiang ve diğerlerinin kitabında belirttiği ve Şekil 1.6.'da da gözüktüğü gibi hata belirlenmeden önce, optimal olarak PID döngüsü tamamlanmalı (Chiang, vd., 2000: 5)



Şekil 1.6. PID Döngüsü

Mobil manipulatörler, hareketli bir platform ve üzerine monte edilmiş bir veya birden fazla robot kolundan oluşur. Bu sistemler, çeşitli endüstriyel ve araştırma uygulamalarında kullanılmaktadır ve başarılı bir şekilde çalışması için hem mobil platformun hem de manipulatör kollarının hassas bir şekilde kontrol edilmesi gerekmektedir. PID kontrolörleri, mobil manipulatörlerin hareketlerini düzenlemek için sıklıkla kullanılır. Bu kontrolörler, hedeflenen konum veya yörüngeye ulaşmak için gereken gücü hesaplamak üzere sistemden alınan geri bildirimlere dayanır. Örneğin, bir manipulatör kolu belirli bir nesneye doğru hareket ederken, PID kontrolörü, koldaki her bir motorun hızını ve torkunu ayarlayarak kolun doğru yörüngede kalmasını sağlar. PID kontrolörleri, sistemin hızlı ve doğru bir şekilde tepki vermesini, sürekli hataları düzeltmesini ve aşırı tepki vermeyi önlemesini sağlayarak, mobil manipulatörlerin geniş bir uygulama yelpazesinde başarılı bir şekilde kullanılmasına olanak tanır.

Mobil bir manipülatörü etkili bir şekilde monte etmek için, hataları en aza indirmek ve başarılı bir montaj sağlamak amacıyla birkaç kritik husus göz önünde bulundurulmalıdır. Hamner ve diğerlerinin belirttiği gibi; şu anki robotların donanım ve yazılımları çok özel görevler için tasarlanmıştır. Geleceğe bakıldığında, gelecekteki robotlar, geniş çalışma alanları boyunca yük taşımaktan küçük bileşenlerin hassas yerleştirilmesine veya montajına kadar çeşitli görevleri üstlenebilir (Hamner, vd., 2010: 131). Mobil manipülatörlerin montajında başarıya ulaşmak için, ilk adım mobil taban ve manipülatörün hareketlerinin etkili bir şekilde koordine edilmesidir. Bu koordinasyon, manipülatörün erişim alanını genişletir ve istenen uç efektör pozlarını hassas bir şekilde elde etmek için kritik öneme sahiptir. İdeal bir montaj süreci için, bu koordinasyonun yüksek manipülasyon kabiliyetiyle birleştirilmesi ve gelişmiş kontrol stratejilerinin uygulanması gerekmektedir. Özellikle, montaj görevlerinin karmaşıklığı ve tolerans gereksinimleri göz önünde bulundurulduğunda, bu koordine hareketler, hataları azaltarak ve montajın güvenilirliğini artırabilmelidir, Şekil 1.7.'de montaj için örnek birleştirme gösterilmiştir.

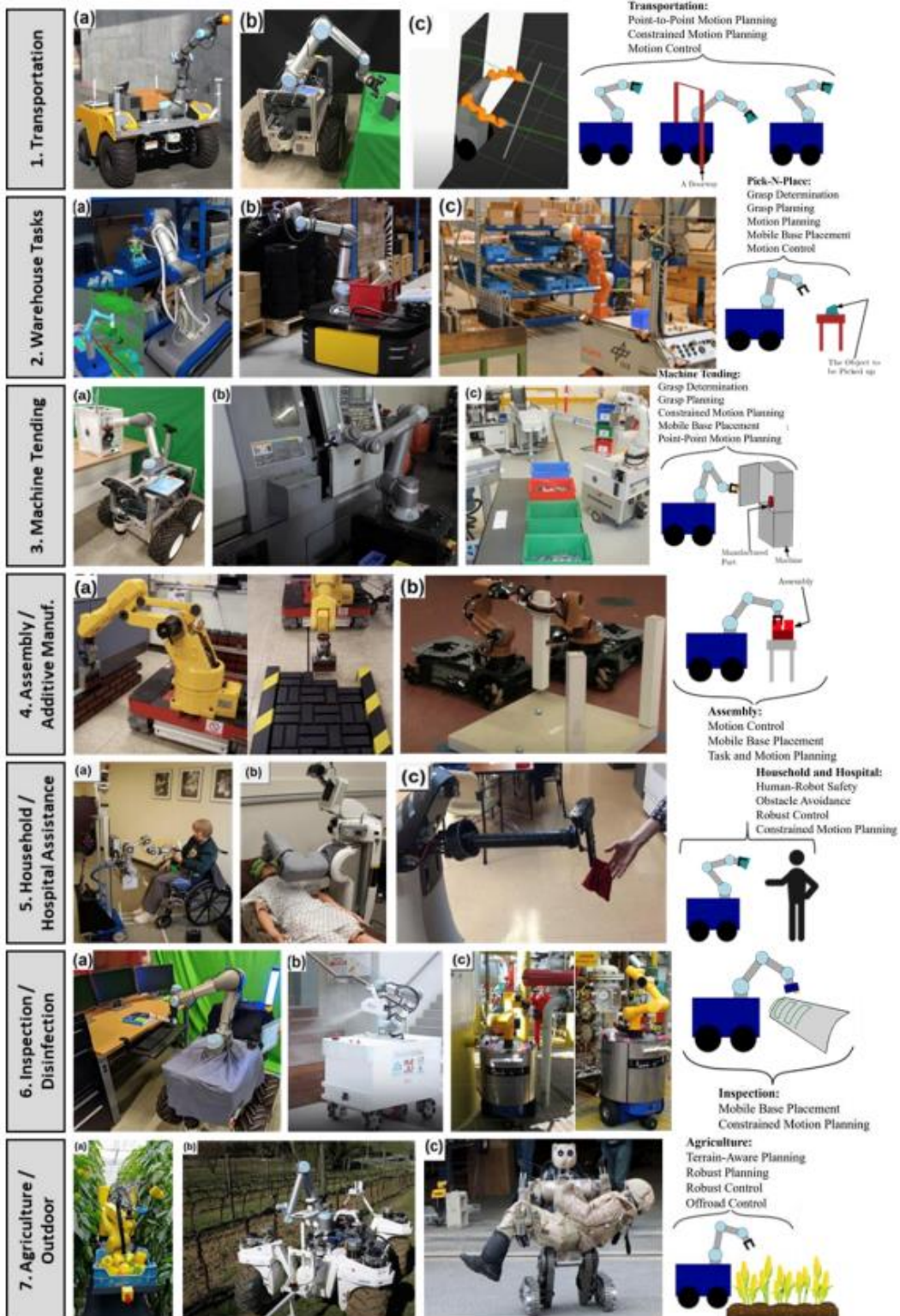


Şekil 1.7. Montajlama Örneği

Kaynak: (Xie, vd, 2023: 95)

Montaj sürecinde karşılaşılan belirsizliklerin ve istisnaların üstesinden gelmek için sofistike reaktif görev kontrol sistemleri kullanılmalıdır. Bu sistemler, mobil manipülatörün çevresel değişikliklere ve beklenmeyen durumlara hızlı ve etkili bir şekilde tepki vermesini sağlar. Bu tür bir kontrol sistemi, manipülatörün montaj sırasında karşılaşılan engelleri aşmasına ve hassas görevleri yüksek güvenilirlikle tamamlamasına olanak tanır. Özellikle, montaj görevlerinde ortaya çıkabilecek geometrik ve mekanik belirsizlikler düşünüldüğünde, reaktif görev kontrolü, montajın doğruluğunu ve verimliliğini artırmak için hayati bir rol oynar.

Bu üç ana stratejiyi birleştirerek, mobil manipülatörlerin montajında hataları en aza indirmek ve başarılı bir montaj sağlamak mümkündür. Koordinasyon ve kontrol, görsel ve kuvvet servo kontrolünün birleştirilmesi ve sofistike reaktif görev kontrolü, bu karmaşık sistemlerin etkin bir şekilde çalışmasını sağlayarak, montaj görevlerinin hassasiyetini ve güvenilirliğini artırır. Bu yaklaşımlar, mobil manipülatörlerin otomotiv montajı gibi karmaşık endüstriyel uygulamalarda etkili bir şekilde kullanılmasının yolunu açar. Thakar ve diğerlerinin de belirttiği gibi; Mobil manipülatör mimarilerinin birçok farklı türü mümkündür ve bu, mobil tabanların doğasına (ray veya XY-gantry monteli sistem, tekerlekli/izli platform ve daha fazlası) ve monte edilen manipülatörün doğasına (eklem sayısı ve türü ile aktüasyonları, örneğin seri vs. paralel) dayanır. Tekerlekli mobil manipülatörler, bir veya daha fazla monte edilmiş manipülatör kolu bulunan bir tekerlekli mobil robot tabanından oluşur (Thakar, vd., 2023: 020801). Şekil 1.8’de çeşitli endüstrilerde kullanılan robotlar gösterilmiştir. Bu alanlar mobil manipülatörlerin endüstrilerde hangi alanda olduğunu belirtmiştir; her biri, robotların taşıma, depo görevleri, makine besleme, montaj, ev içi yardım, denetim ve dezenfeksiyon, tarım ve dış mekan faaliyetleri gibi görevleri yerine getirdiği farklı senaryoları temsil eder. Resimler, robotların bu görevleri yerine getirirken kullandığı belirli teknikleri ve hareket planlamalarını da açıklamaktadır, bu da robotların karmaşık ve dinamik ortamlarda verimli bir şekilde çalışabilmelerini sağlayan yeteneklerini sergilemektedir.



Şekil 1.8. Mobil Manipülörlerin Çeşitli Endüstrilerde Kullanım Alanı

Kaynak: (Thakar, vd., 2023: 020801)

Mobil manipülatörlerin çeşitliliği ve bu sistemlerin tasarımında karşılaşılan önemli seçenekler vurgulanmıştır. Mobil manipülatörler, genellikle hareket kabiliyetleri ve görevleri yerine getirme şekilleri açısından farklılık gösterir. Bu farklılıklar, temel olarak iki ana bileşene dayanır: mobil taban ve monte edilen manipülatör. Mobil taban, raylı sistemler, XY-rampa sistemleri, tekerlekli veya izli platformlar gibi çeşitli yapılandırmalarda olur. Montajlı manipülatör ise eklemlerinin sayısı, türü ve çalışma şekli (seri veya paralel gibi) açısından farklılık gösterir. Tekerlekli mobil manipülatörler, en yaygın kullanılan türlerden biridir ve genellikle bir veya daha fazla robot kolunun tekerlekli bir platforma monte edilmesiyle karakterize edilir. Bu çeşitlilik, mobil manipülatörlerin geniş bir uygulama yelpazesine uyum sağlamasına olanak tanır, ancak aynı zamanda kontrol sistemlerinin tasarımını ve uygulamasını karmaşıklaştırır.

Mobil robot manipülatörlerde arıza çeşitleri Yüksel ve Sezgin'in yaptığı çalışmada şu şekilde geçmektedir; ilgili kullanılan yöntemlere geçmeden arızaların sınıflandırılması ile ilgili üç tip sınıflandırmadan bahsedilmektedir; aktüatör arızası, bileşen arızası, algılayıcı arızası (Yüksel ve Sezgin, 2009: 80). İlk sınıflandırma aktüatör arızaları; McIntyre ve diğerlerinin çalışmasına göre; Robot manipülatörlerindeki aktüatör arızaları, genellikle üç ana formda görülür: Birinci olarak, kilitli eklemler, burada bir eklem torka bağlı olmaksızın sabit bir konumda takılı kalır. İkinci olarak, serbest salınan eklemler, bu durumda eklem sıfır tork nedeniyle serbestçe hareket eder. Üçüncü olarak, kısmi aktüatör arızaları, bu durumda aktüatörün tork üretiminde belirli bir yüzde oranında azalma görülür (McIntyre, vd., 2004: 4982). Bileşen arızası, mobil manipülatörlerde, mobil platform üzerine monte edilmiş manipülatörün bileşenlerinden birinin arızalanması durumunu ifade eder. Bu tür arızalar, mobil manipülatörün hareket kabiliyetini, dengeyi ve manipülasyon yeteneklerini etkiler. Algılayıcı arızası, mobil manipülatörlerde, manipülatörün hareket ve çevresel etkileşimleri algılamak için kullandığı sensörlerin arızalanması durumunu ifade eder. Bu tür arızalar, mobil manipülatörün çevresel koşulları algılamasını, nesnelere etkileşimini ve güvenliğini etkiler.

Bu tezde, mobil manipülatörlerin eklemlerinde meydana gelen arızaları tespit etmek için derin öğrenme yöntemlerini esas alır. Bu yöntemlerin kullanımı, arıza teşhisinin daha hızlı ve etkin bir şekilde gerçekleştirilmesine olanak tanırken, aynı zamanda sistemlerin adaptasyon ve öğrenme yeteneklerini de arttırmaktadır. Arıza teşhisi için kullanılan yöntem, eklemlerden elde edilen veriyi analiz eder ve bu veriyi kullanarak arızaları belirlemek için Yapay Sinir Ağı (YSA) modeli oluşturur; makine öğrenimi veya yapay zekâ tekniklerini kullanmak, en iyi modelleri geliştirmek

için büyük miktarda veriyi analiz eden yöntemler önerilmiştir, (Park vd., 2018: 2110)'de belirtildiği gibi bu şekilde açıklamıştır. Mobil manipülatörden elde edilen seri setinden elde edilen bilgilerin yapay zekâ ile eğitilerek kullanılması, derin öğrenme ile arıza teşhisi olarak adlandırılır. Mobil manipülatörlerdeki arıza tespitinde derin öğrenmenin önemi, arıza teşhis sistemlerinin doğruluğunu, verimliliğini ve uyum yeteneğini artırma kapasitesinde yatmaktadır. Evrişimsel sinir ağları (CNN) ve uzun kısa süreli hafıza (LSTM) ağları gibi derin öğrenme teknikleri, endüstriyel robotik, mobil robot navigasyonu ve motor sürücü kontrol sistemleri dahil olmak üzere çeşitli alanlarda arızaları tespit etme ve izole etme konusunda önemli potansiyel göstermiştir. Wang ve diğerlerinin de araştırdığı gibi; derin öğrenme teknolojisi, veriye dayalı arıza tespitinde şimdiye kadar kapsamlı bir şekilde araştırılmıştır. Tarihsel veri setine dayalı olarak, dinamik olarak konumlandırılan sinirlerin itici arızasını tespit etmek üzere derin bir CNN modeli tasarlanmıştır (Wang vd., 2023: 144). Derin öğrenmede, bir robotun veya mobil manipülatörün hareketini kontrol etmek için elde edilen veriler doğrudan bir robot manipülatörün veya robotun eklemlerinden elde edilen veriler ile elde edilir. Bu durumda robotun veya veri elde edilen mobil manipülatörün hareketi, veri setini hareket algılama kabiliyetini aktive eder. Elde edilen veri setinin eğitilmesi için ise Sinir Ağı (NN) ve LSTM eğitim algoritmalarının kullanılmalıdır. Bu tez çalışmasında NN ve LSTM eğitim algoritmaları kullanılmıştır ve karşılaştırılması yapılmıştır. Bölüm 1'de mobil manipülatörlerin tarihçesi, kullanım alanları, montajlanması ve kullanılan sinir ağları hakkında genel bir yapılmıştır. Daha sonra gelen Bölüm 1.1'de mobil manipülatörlerde oluşan arızalardan ve çözüm yöntemlerinden bahsedilmiştir.

Bölüm 2'de tezin içeriğinde olan konuların teorik temelleri anlatılmıştır. Bunlar; mobil manipülatörün çalışma prensibi, arıza tespitinin temelleri, yapay sinir ağlarıdır.

Bölüm 3'te kullanılan mobil manipülatörün özellikleri, Coppelia simülasyon ortamı, Coppelia ve Matlab haberleşmesi, Sistem YSA Giriş Sinyallerinin Çıkarımı, YSA ve Neural Network Toolbox, ve yörünge planlama ve görüş alanı koruyucu, YSA duruş kestirimci girişlerine rastgele gürültü eklenmesi ve analizi anlatılmıştır.

Bölüm 4'te ise, YSA mimarisini kullanarak duruş kestiriminde ve iki farklı sinir ağı modeli karşılaştırılarak, arızaların tespit edilmesinin özetine, bulguların derin öğrenmenin mobil manipülatör arıza tespitinde etkili oluş şekline ve gelecek çalışmalara değinilmiştir.

1.1. Literatür Özeti

Mobil manipulatörlerde derin öğrenme tabanlı ilk çalışmalardan birisi (Han ve Kopacek, 1997)'de yapılmıştır. Han ve Kopacek çalışmalarında; yapay sinir ağları (YSA) ve Fuzzy Robot Control (Fuzzy sistemler) gibi gelişmiş kontrol tekniklerinin mobil manipulatörlerde nasıl kullanıldığını inceler, bu sistemlerin dinamik ve doğrusal olmayan davranışlarını etkili bir şekilde yönetme yeteneklerine odaklanır. Mobil manipulatörlerin artan kullanımı ve bu sistemlerin bilinmeyen ortamlarda akıllı hareketlilik ve işleme yeteneği gereksinimine dikkat çeker, ayrıca, mobil robotların navigasyonu ve çarpışma önleme gibi zorlukları ele alır, bu konularda YSA ve fuzzy sistemlerin uygulamalarını ve entegrasyonunu tartışır ve Son olarak, ultrasonik sensörlerin ve YSA ile bulanık kontrolün entegrasyonunun, mobil manipulatörlerin etkinliğini ve güvenilirliğini artırmak için nasıl kullanılabileceğini belirtmişlerdir ancak yapay sinir ağlarının kullanımına değinilmemiştir. Sonrasındaki önemli çalışmalardan birisi olan Skoundrianos ve Tzafestas tarafından da (SKOUNDRIANOS ve TZAFESTAS, 2004) şu şekilde belirtilmiştir; robotun tekerleklerine uygulanan voltaj ile tekerleklerin hızı arasındaki ilişkiyi tanımlayan bir model üzerinde durur. Bu ilişki, tekerlekler arasındaki çapraz bağımlılık nedeniyle doğrusal olmayan bir karakter gösterir. Sinir ağı yapısı, ağ çıktısının bir sonraki değerini önceki çıktı ve geçmiş girdi değerlerine bağlı olarak lineer bir tahminle belirler. Bu yapı, hata tespiti ve teşhisinde kritik olan kalıntı sinyallerinin oluşturulmasında kullanılır, Makalenin sonucunda, önerilen yöntemin hızlı ve verimli çalıştığı, yanlış alarmların önüne geçildiği belirtilir. Yöntemin başarısı, modelleme sürecinin performansına dayanmaktadır. Yerel model ağlarının kullanımı, işletim rejimine doğrudan bir referans sağlayarak modelin daha iyi anlaşılmasını ve dolayısıyla daha kolay manipüle edilmesini sağlar. Bu makale sadece tekerlek bazında inceleme yapmıştır ama eklemler için de aynı özellik geçerlidir. 2010 yıllarında (Yüksel ve Sezgin, 2010:125)'da ise Neural Network (NN)'lerin ve Bulanık Mantık (FL) sistemlerinin hata tespiti ve izolasyonu için kullanımına odaklanılır. ANFIS (Adaptive Neuro Fuzzy Inference System), NN ve FL'nin birleşimi, etkili bir sistem modellemesi için kullanılır ve bu yaklaşımın avantajları ve uygulamaları tartışılır, Bu sonuçlar, robot manipulatörler için etkili ve güvenilir (FL) FDI şemalarının geliştirilmesinde önemli bir adımı temsil etmektedir. Aynı zamanda da; 2012 yılında yayınlanan Akhtar'ın kitabında (Akhtar, 2012); makalenin sunduğu FDI yaklaşımlarını destekler. Makaledeki FDI şemaları, mobil manipulatörlerin hata tespiti ve izolasyonuna odaklanırken, bu kitap, hataların özellikle manipulatörlerin nesnelere etkileşimi sırasında ortaya çıkan bilinmeyen dış hatalarına yönelik bir

özüm sunar. Her iki kaynak da, mobil manipölätörlerdeki hataların tespiti ve yönetimi için sofistike ve entegre edilmiş yaklaşımların önemini vurgular. Makalede sunulan model tabanlı FDI şemaları ve kitapta açıklanan bilinmeyen dış hatalara karşı güvenilirlik artırma yaklaşımı, mobil manipölätörlerin daha etkin ve güvenilir bir şekilde çalışmasını sağlamak için birbirini tamamlayıcı stratejiler sunar.

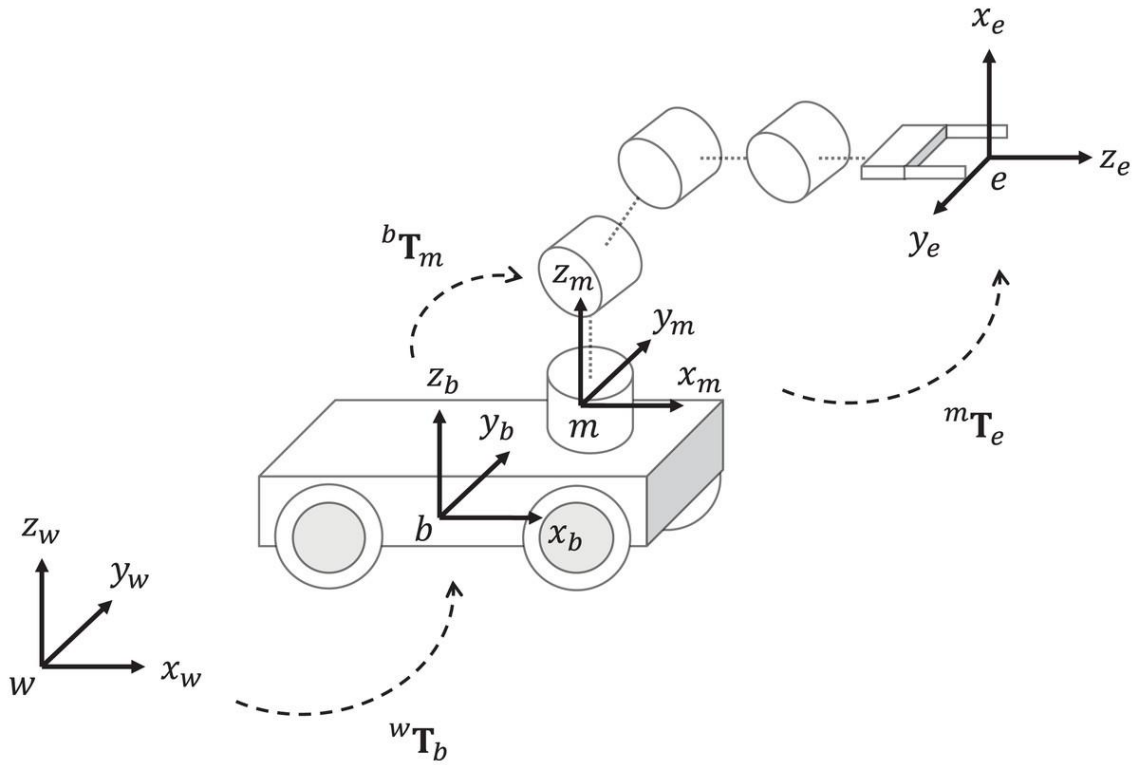
2022’de ise Pan ve diğerlerinin yaptığı çalışmada (Pan, vd., 2023) robot kol hata tespiti ve tanılamasını karmaşık ortamlarda makine görüşüne dayalı olarak ele almıştır. Önerilen yöntem, çoklu bilgi birleştirme ve dropout tabanlı makine öğrenimi kullanarak hata tespiti ve tanılamasını gerçekleştirir. Bu süreçte, robot kolların hareketlerini izleyen ve hataları belirleyen bir makine öğrenimi modeli kullanılır. Model, görsel sensörler ve iç sensörlerden gelen verileri birleştirerek hataları tanımlar ve sınıflandırır. Makalenin odak noktası, karmaşık ortamlarda robot kol arızalarını etkin bir şekilde tespit etmek ve tanılamaktır.

2. MATERYAL

Bu bölümde tezin içeriğinde olan konuların teorik temelleri anlatılmıştır. Bu konular şunlardır; mobil robot manipülatörlerin çalışma prensibi, arıza bulma temelleri, yapay sinir ağı ve arıza tespiti.

2.1. Mobil Robot Manipülatörlerin Çalışma Prensibi

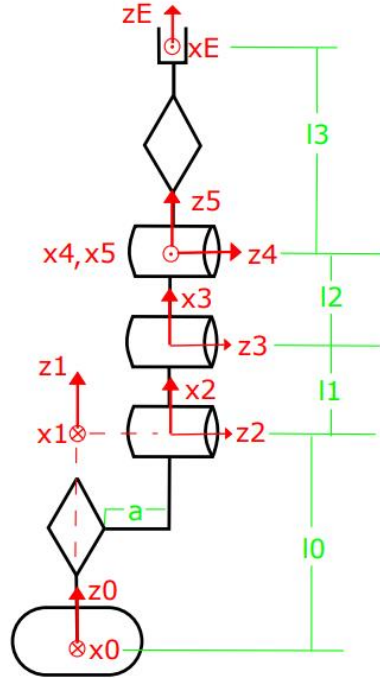
Mobil robot manipülatörünün çalışma prensibi, hareket kabiliyeti ile manipülasyon yeteneklerinin entegrasyonunu içerir. Bu, robotun çevresinde gezinmesini ve aynı zamanda çeşitli manipülasyon görevlerini gerçekleştirmesini sağlar Şekil 2.1.'de mobil robot manipülatörün eklemleri ve dönme yönleri gösterilmiştir.



Şekil 2.1. Mobil Robot Manipülatör 5 Serbestlik Derecesindeki Eklemleri ve Tekerleklerin Dönme Yönleri

Kaynak: (Hernandez-Barragan, vd., 2021: 419)

İleri kinematik, bir robotun kinematik denklemlerini kullanarak, eklem parametrelerinin belirli değerlerinden uç etkileyicinin pozisyonunu hesaplama işlemine atıfta bulunur. Bu nedenle, açık zincirli, dönme eklemleri bir kol için eklem açıları, kartezyen uzayındaki uç etkileyici pozisyonuna dönüştüren haritayı elde etmek, öncelikle " karmaşık " bir geometrik problem ve ikincil olarak bir cebirsel problem olarak kabul edilir. Bu probleme sistemli yaklaşımlar bulunmaktadır. Özellikle, robotik alanında en popüler yöntemlerden biri Denavit-Hartenberg (D-H) kuralıdır; genel olarak, bir açık zincir manipülatör $n + 1$ bağlantıdan oluşur ve bu bağlantılar n eklemlerle birbirine bağlanmıştır. Bu yapı, manipülatörün hareket kabiliyetini ve konumlandırmasını belirler. Varsayılan olarak, sıfırıncı bağlantı (0^{th}) manipülatörün tabanına veya zemine bağlıdır. Bu, manipülatörün başlangıç referans noktasını oluşturur ve kinematik analizler için temel bir yapı sağlar. Eklem, iki ardışık bağlantıyı birbirine bağladığı için, bir bağlantı üzerindeki koordinat çerçevesinin pozisyonunu ve yönelimini homojen bir dönüşümle açıklamak mümkündür. Denavit-Hartenberg (D-H) kuralının amacı, bu homojen dönüşümlerin türetimini kolaylaştırmak ve bu dönüşümlerin matris çarpımı yoluyla ileri kinematik haritasını yinelemeli olarak bulmaktır. D-H kuralı, bir çerçeve i ile önceki çerçeve ile $i - 1$ olan pozisyonunu tam olarak belirlemek için dört parametre kullanır. Bu dört parametre, çerçevenin konumunu ve yönelimini detaylı bir şekilde tanımlar, böylece robot kolun her bir bölümünün mekansal ilişkilerini net bir şekilde belirleyebilir. Bu parametreler, robotun her eklemindeki hareketin ve konumunun hesaplanmasında kullanılır, böylece robotun genel hareket ve konumlandırma yetenekleri belirlenebilir. a_i , çerçeveler arasındaki doğrusal uzaklığı ifade eder ve robot kolunun her bir eklemi arasındaki mekansal konumlandırmayı belirlemek için kullanılır. D-H kuralında kullanılan d_i parametresi, O_i noktasının z_{i-1} eksenine göre koordinatını belirtir. Bu parametre, bir önceki çerçeve ile mevcut çerçeve arasındaki doğrusal mesafeyi ifade eder ve robot kolunun uzaysal konumlandırmasını tanımlamada kullanılır. D-H kuralındaki α_i parametresi, z_{i-1} ve z_i eksenleri arasındaki açıyı tanımlar ve bu açı x_i eksenine etrafında ölçülür. Dönüş saat yönünün tersine yapıldığında bu açı pozitif olarak alınır. Bu, robot kolunun eklemleri arasındaki açısal ilişkiyi belirleyen önemli bir faktördür. D-H kuralındaki θ_i parametresi, x_{i-1} ve x_i eksenleri arasındaki açıyı tanımlar ve bu açı z_{i-1} eksenine etrafında ölçülür. Dönüş saat yönünün tersine yapıldığında bu açı pozitif olarak kabul edilir. Bu parametre, robot kolunun eklemleri arasındaki dönüşsel ilişkiyi belirler ve kinematik yapıyı detaylandırmada kullanılır. D-H kurallarına göre çerçeve bağlantısı yapılırken, KUKA youBot için çerçeveler Şekil 2.2.'de gösterilmiştir



Şekil 2.2. KUKA youBot Çerçevesi

Kuka youBot'un parametrelerinin değerleri Şekil 2.3.'te belirtilmiştir.



Şekil 2.3. KUKA youBot Parametreleri

Kaynak: (Hernandez-Barragan, vd., 2021: 419)

D-H parametrelerinin deęerleri ařaęıdaki Tablo 2.1.'de gsterilmiřtir;

Tablo 2.1. D-H parametrelerinin deęerleri

Eklem	a (mm)	α (rad)	d (mm)	θ (rad)
1	33	$\frac{\pi}{2}$	147	θ_1
2	155	0	0	θ_2
3	135	0	0	θ_3
4	0	$\frac{\pi}{2}$	0	θ_4
5	0	0	217,5	θ_5

a Parametresi, link uzunluęu yani, kolun iki dikey z-ekseni arasındaki mesafeyi ifade eder. Modifiye edilmiř D-H kuralına gre, ereve i 'den $i-1$ 'e homojen dnřm, belirli bir matris formlasyonu kullanılarak etkili bir řekilde gerekleřtirilir. Bu dnřm, robot kolunun her bir eklemi arasındaki uzaysal iliřkiyi ve hareket kabiliyetini matematiksel olarak ifade eder ve robotun hareketlerini hesaplariken kritik bir rol oynar. Denklem (2.1)'de gsterilmiřtir

$${}_{i-1}T_i = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 & a_{i-1} \\ \sin(q_i)\cos(\alpha_{i-1}) & \cos(q_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(q_i)\sin(\alpha_{i-1}) & \cos(q_i)\sin(\alpha_{i-1}) & -\cos(\alpha_{i-1}) & -d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.1)$$

KUKA Youbot'un 5 Serbestlik Dereceli (D.O.F.) maniplatr iin, u etkileyici ile taban arasındaki dnřm, maniplatrn u noktasının pozisyonunu ve ynlendirmesini, onun tabanına gre matematiksel olarak ifade eder. Genellikle, bir dizi homojen dnřm ierir ve bu dnřimler, maniplatrn her bir eklem hareketini sırasıyla uygular. Denklem 2.2'de olduęu gibi:

$${}^0T_E = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 \quad (2.2)$$

Bu noktada, son homojen dnřmn sadece bir eviri iřlemi olduęu belirtilmelidir. Ayrıca, analizin ilerleyen blmlerinde elde edilen denklemleri kolayca iřleyebilmek iin ereve baęlantılarına bazı ayarlamalar yapılmıřtır. Tablo 2.1'den alınan parametrelerle yer deęiřtirme yapıldıktan sonra, sırasıyla ereveden ereveye homojen dnřimler ařaęıdaki gibi elde edilir.

$${}^0T_1(\theta_1) = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0.033\cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0.33(\sin(\theta_1)) \\ 0 & 1 & 0 & 0.147 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.3)$$

$${}^1T_2(\theta_2) = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0.155\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0.155(\sin(\theta_2)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$${}^2T_3(\theta_3) = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0.135\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0.135(\sin(\theta_3)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

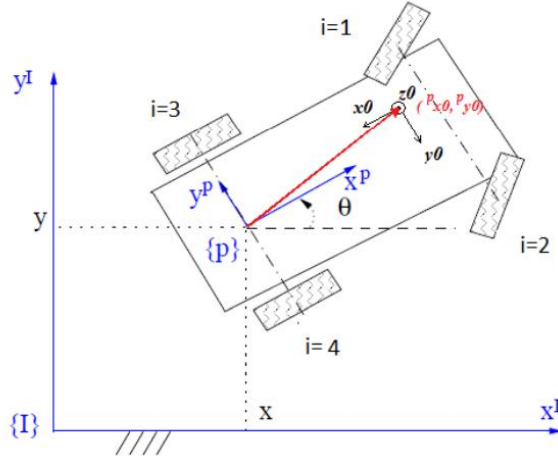
$${}^3T_4(\theta_4) = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$${}^4T_5(\theta_5) = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0.2175 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$q = [x_b \ y_b \ \theta_b \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^T \quad (2.8)$$

Eklem θ_1 - θ_5 değerleri, manipülatörün eklem konfigürasyonudur; x_b , y_b , θ_b mobil platformun sırasıyla konum ve yönelim değişkenleridir ve bu şekilde koordinatları bulunmuş olur.

Şu an itibarıyla, kol için tam bir kinematik analiz gerçekleştirilmiştir. Kombine sistem analizine devam edebilmek için, mobil platform için ayrı bir analizin önceden tamamlanmış olması gerekmektedir. Mobil platform analizi 6. – 7. – 8. ve 9. Eklem olarak analiz edilmiştir ve koordinat çerçevesi Şekil 2.4.'te gösterilmiştir;



Şekil 2.4. KUKA youBot Platformu ve İlişkili Koordinat Çerçevesleri

Notasyonlarda 'p' harfi platformu, 'I' harfi ise küresel-eylemsiz çerçeveyi temsil eder. Ek olarak, analizi kolaylaştırmak için platform çerçevesinin orijininin platformun merkeziyle örtüştüğü varsayılmıştır. Mobil platform için rotasyon matrisleri Denklem 2.8'de gösterilmiştir;

$${}^P\mathbf{R}_0 = \mathbf{R}_{z_0 \equiv z_P} \Big|_{\theta_z = 180^\circ} = \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\theta_z = 180^\circ} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Platformun koordinat sisteminden küresel-eylemsiz koordinat sistemine olan dönüşüm matrisi, platformun mekânsal yönlendirilmesini ve global çerçevedeki konumunu belirlemek için kullanılır. Bu matris, platformun dönüş açılarını ve eksenel hizalanmasını ifade ederek, platformun küresel koordinatlara göre nasıl hareket ettiğini ve yönlendirildiğini matematiksel olarak denklem 2.9'daki gibi tanımlar:

$${}^I\mathbf{R}_P = \begin{bmatrix} \cos \theta_P & \sin \theta_P & 0 \\ -\sin \theta_P & \cos \theta_P & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Bu durumda, manipülatör taban çerçevesinden küresel-eylemsiz çerçeveye olan dönüşüm matrisi, manipülatörün küresel çerçevedeki konumunu ve yönlendirmesini tanımlar. Bu matris, manipülatörün tabanından küresel referans çerçevesine olan mekânsal ilişkiyi ve dönüşümü matematiksel olarak denklem 2.10'daki gibi ifade eder;

$${}^I\mathbf{R}_P = {}^I\mathbf{R}_P {}^P\mathbf{R}_0 = \begin{bmatrix} \cos \theta_P & \sin \theta_P & 0 \\ -\sin \theta_P & \cos \theta_P & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Manipülâtör taban çerçevesinin küresel-eylemsiz çerçeveye göre konumunu tam olarak tanımlamak için, belirli varsayımlar yapılır. Bu, manipülâtörün küresel referans çerçevesindeki mekansal konumunu ve yönlendirmesini belirlemek için gerekli bir adımdır. Bu varsayımlar, manipülâtörün tabanının konumunu ve eylemsiz çerçeveye göre oryantasyonunu net bir şekilde tanımlar. Manipülâtör taban çerçevesinin küresel-eylemsiz çerçeveye göre konumunu tanımlarken, bu konumun küresel-eylemsiz z-ekseni boyunca sabit olduğunu varsayılır ve bu değer I_{Z_0} ile ifade edilir. Bu varsayım, manipülâtörün küresel referans çerçevesindeki dikey konumunu belirlemede kullanılır. $\dot{\mathbf{q}}_P$ mobil platformun hız vektörüdür. Bu model, platformun hareket dinamiklerini ve kinematik özelliklerini matematiksel olarak ifade eder Jacobian matrisi (\mathbf{J}_P) hızını ve yönlendirmesini manipülâtörün tabanının hareketleriyle ilişkilendirir ve platformun hareket kabiliyetlerini anlamada önemli bir rol oynar. Kontrol girdileri (\mathbf{u}_P) motor komutları veya aktüatör girdileridir. Denklem 2.11 ile platformun hareket kabiliyetinin anlaşılması sağlanır

$$\dot{\mathbf{q}}_P = \mathbf{J}_P \mathbf{u}_P \quad (2.11)$$

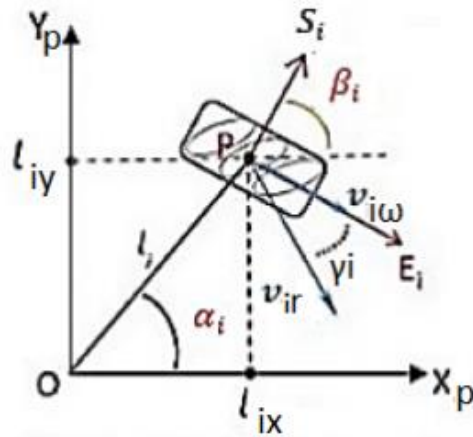
Bu durumda $\mathbf{q}_P = [x_P \ y_P \ \theta_P]^T$ platformun serbestlik derecelerinin vektörünü ifade eder. Bu matris platformun serbestlik derecelerinin bir fonksiyonudur.

KUKA youBot platformu, Mecanum tekerlekleri ile donatılmıştır. Bu geleneksel tekerlekler, çaplarına bağlı bir dizi silindirle gelir. Sol ve sağ elle çalışan silindirlerle tekerlekleri değiştirerek, her tekerleğin tekerlek tabanı çaprazına yaklaşık olarak dik açılarda kuvvet uyguladığı şekilde, araç stabil hale getirilir ve her tekerleğin dönme hızı ve yönünü değiştirerek herhangi bir yönde hareket ettirir ve döndürür. Dört tekerleği aynı yönde hareket ettirmek, aracın ileri veya geri hareketini sağlar. Bir tarafın tekerleklerini diğer tarafinkilerin tersi yönde çalıştırmak, aracın dönmesine neden olur. Bir diyagonal üzerindeki tekerlekleri, diğer diyagonal üzerindeki tersi yönde çalıştırmak ise yan harekete sebep olur. Bu tekerlek hareketlerinin kombinasyonları, aracın herhangi bir yönde ve herhangi bir araç dönüşüyle hareket etmesini sağlar. Şekil 2.5'te Mecanum tekerlek gösterilmiştir.



Şekil 2.5. Mecanum Tekerlek

Şekil 2.6.'da ise mecanum tekerleğin koordinat sisteminde gösterimi verilmiştir;



Şekil 2.6. Mecanum Tekerleğin Koordinat Sisteminde Gösterimi

Omnidireksiyonel bir platformun modellenmesi, platformun doğrusal ve açısal hızlarının Mecanum tekerleklerinin dönme hızlarına bağlı olduğu basit bir varsayıma dayanır. Doğal olarak, bu ilişkileri türetmek için uygulanabilecek geometrik yöntemler vardır. Bu yöntemler, belirli geometrik niceliklere dayalı olarak \mathbf{J}_P matrisinin değerlendirilmesine dayanır.

i , tekerlek indeksidir, 1'den 4'e kadar numaralandırılır. α_i , Tekerleğin yere göre montaj açısı. β_i , tekerlek dönüş yönüne göre tekerleğin eğim açısıdır. γ_i , tekerleğin dönüş yönüdür. l_i , tekerleğin robotun merkezine olan mesafesi. l_{ix} ve l_{iy} ise tekerleğin robotun merkezine olan x ve y bileşenleridir. Tekerleklerin tamamen aynı ve yarıçapları r olduğunu varsayılırsa, bu açılar için tipik parametreler Tablo 2.2'de sunulmuştur.

Tablo 2.2. Tekerlek Parametreleri

i	α_i	β_i	γ_i	l_i	l_{ix}	l_{iy}
1	$\pi/4$	$\pi/2$	$-\pi/4$	l	l_x	l_y
2	$-\pi/4$	$-\pi/2$	$\pi/4$	l	l_x	l_y
3	$3\pi/4$	$\pi/2$	$\pi/4$	l	l_x	l_y
4	$-3\pi/4$	$-\pi/2$	$-\pi/4$	l	l_x	l_y

Basit hız vektörü ve geometrik ilişkileri kullanarak ya da yukarıdaki değerleri kinematik modellere yerleştirerek, kinematik denklemi denklem 2.12'deki gibi türetilir. Bu işlem, platformun ve tekerleklerin hareketlerinin ve açısal hızın (ω) matematiksel modelini oluşturur ve platformun hareket kabiliyetlerini ve sınırlamalarını anlamada önemli bir rol oynar.

$$\dot{\mathbf{q}}_P = \mathbf{J}_{P,o} \mathbf{u}_{P,o} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} [\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4]^T \quad (2.12)$$

2.2. Arıza Analizi Temelleri

Bir n serbestlik dereceli robot manipülatörün hareketi, diferansiyel denklemlerle tanımlanır. Bu, manipülatörün kinematik ve dinamik özelliklerini matematiksel olarak ifade eden bir model sağlar ve robotun hareketlerini analiz etmek için temel oluşturur. Bu tür bir model, robotun hareket kabiliyetlerini anlamak ve kontrol sistemlerini tasarlamak için önemlidir (Denklem 2.12)

$$\ddot{q} = M^{-1}(q)[\tau - V_m(q, \dot{q})\dot{q} - G(q) - F(\dot{q}) - \tau_d(t)] + \beta(t - T)\psi(q, \dot{q}, \tau) \quad (2.13)$$

Denklem 2.13'de; $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ eklem pozisyonların vektörlerini temsil eder, robot manipülatörün hareketi, hızlar ve ivmeler ile tanımlanan diferansiyel denklemlerle ifade edilir. Bu, manipülatörün mekanik davranışını ve hareket dinamiklerini anlamada kullanılan temel bir yaklaşımdır. Hızlar, manipülatörün anlık hareket durumunu, ivmeler ise bu hareketin zaman içinde nasıl değiştiğini gösterir. $\tau \in \mathbb{R}^n$ tork vektörünün girişi. $M(q)$ ise robot manipülatörün kinematik modelinde, atalet matrisi (tersi bulunan) önemli bir rol oynar. Bu matris, robotun hareketine direnç gösteren atalet kuvvetlerini ifade eder ve manipülatörün her bir bağlantısının atalet özelliklerini matematiksel olarak modellemeye yardımcı olur. Bu atalet matrisi, robotun dinamik davranışını anlamak ve kontrol sistemlerini tasarlamak için kullanılır. $V_m(q, \dot{q})$, robot manipülatörün hareket dinamiklerinde merkezci ve içinde bulunduğu dönen bir referans çerçevesinde gözlemlendiğinde ortaya çıkan görünür kuvvetleri içerir. Bu terimler, manipülatörün hareketi sırasında ortaya çıkan ek kuvvetleri ve etkileri ifade eder ve robotun dinamik modelinin doğruluğunu artırmada önemlidir. $G(q)$; yer çekimi vektörünü ifade, $F(\dot{q})$ hareket denklemlerinde bilinmeyen statik ve dinamik sürtünme terimlerini ve çeşitli belirsizlik kaynaklarını içerir. Bu terimler, manipülatörün hareketi sırasında karşılaşılan sürtünme kuvvetlerini ve diğer etkileyici faktörleri temsil eder. $\tau_d(t)$; bilinmeyen ve sınırlı katkıda bulunan bozulmaları ve gürültüyü temsil eder. Bu, manipülatörün hareketlerini etkileyebilecek rastgele ve öngörülemez faktörleri kapsar ve bu faktörlerin robotun performansı ve hareket kontrolü üzerindeki etkilerini modellemek için kullanılır. Vemuri ve Polycarpou'nun çalışmasına göre; $\tau_d(t)$ ve $F(\dot{q})$ terimleri birlikte kullanıldığında, sistemdeki modelleme belirsizliklerini temsil eder (Vemuri ve Polycarpou, 2004: 420). Bu, robot manipülatörün hareket modelinin tam olarak ifade edilemeyen yönlerini ve olası sapmalarını kapsar, bu da kontrol sistemlerinin tasarımında ve performans analizinde etkin olarak kullanılan yöntemdir. ψ ; vektörü, robot manipülatöründeki arızaları temsil eder. Bu vektör, manipülatördeki

potansiyel hataların veya arızaların neden olduğu sapmaları ve anormallikleri ifade eder ve bu bilgi, arızaların tespit edilmesi ve giderilmesinde kullanılır. $\beta(t - T)$; aman profili, robot manipülatöründeki arızanın zamanla nasıl değiştiğini veya geliştiğini temsil eder. Bu, arızanın başlangıcını, süresini ve şiddetini içerir ve arıza tespiti ve analizi için önemli bir veridir. T , ise arızanın bilinmeyen meydana geliş zamanını temsil eder. Bu, arızanın ne zaman başladığını belirsiz olduğu durumlarda kullanılan bir değişkendir ve arıza tespiti ve analizi için önemli bir bilgi sağlar.

Bir robotik sistemdeki arıza, manipülatörün dinamiklerini tahmin edilemez bir şekilde değiştirir. Arıza koşullarının doğru bir şekilde tanımlanması, çoğu zaman, arızaların doğrusal olmayan modellemesini gerektirir. Bu, (2.13) numaralı denkleminde ψ ile ifade edilen durumdur. Doğrusal olmayan modelleme yeteneği, arıza nedeniyle ortaya çıkan ψ sapmasının, eklemlerin konumlarına, hızlarına ve giriş torkuna bağlı doğrusal olmayan bir fonksiyon olarak izin verilmesiyle yansıtılır. Bu, dinamik sistemdeki arızanın doğrusal olmayan modellemesi, yalnızca ekleme arızalarını (robotik sistemdeki sapmalar) değil, aynı zamanda çarpma arızalarını (manipülatör parametre değişiklikleri) ve ek durumlar yaratan arızaları (örneğin, bağlantı kopması) temsil etmeye olanak tanır. Örneğin, bir arıza nedeniyle T zamanında yer çekimi vektörü G , \bar{G} olarak değişirse, bu yer çekimi terimindeki değişiklik, denklem 2.14'te gösterildiği gibi temsil eder. Bu, arızanın dinamik sistem üzerindeki etkisini ve manipülatörün tepkisini anlamak için kullanılan bir modelleme yöntemidir.

$$\psi(q, \dot{q}, \tau) = M^{-1}(q)[\bar{G}(q) - G(q)] \quad (2.14)$$

Bu durumda, denklem (2.13), robotik sistemdeki bir arızayı, manipülatörün dinamiklerinde bilinmeyen bir değişiklik olarak modellemektedir. Bu değişiklik, zamanın bir fonksiyonu olarak değil, sistem durumunun ve sistem girdilerinin bir fonksiyonu olarak temsil edilir. Bu, çoğu geleneksel arıza modelleme yaklaşımlarının aksine, arızayı zamana bağlı katkılı veya çarpımsal bir miktar olarak temsil eden bir yaklaşımdır.

2.3. Yapay Sinir Ağları ve Öğrenme Algoritmaları

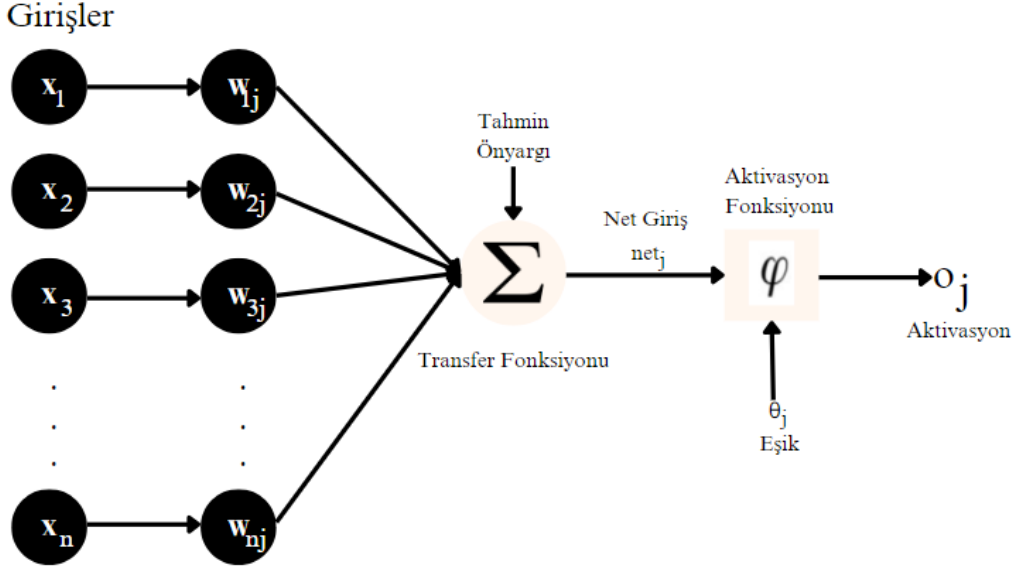
Yapay sinir ağları (YSA), bilgisayar bilimleri ve yapay zekâ alanında yaygın olarak kullanılan bir modeldir. YSA'lar, insan beyninin sinir ağlarından esinlenerek tasarlanmış ve bilgi işleme, desen tanıma, sınıflandırma, regresyon ve tahmin gibi çeşitli görevlerde kullanılmıştır.

YSA'lar, girdi verilerini işleyerek öğrenme süreciyle belirli bir görevi yerine getirmek üzere eğitilir ve genellikle katmanlar, nöronlar ve ağırlıklardan oluşan bir yapıya sahiptir. Ergezer ve diğerlerinin yaptığı çalışmada; Yapay sinir ağları temelde tamamen insan beyni örneklenerek geliştirilmiş bir teknolojidir. Bilindiği gibi; öğrenme, hatırlama, düşünme gibi tüm insan davranışlarının temelinde sinir hücreleri bulunmaktadır (Ergazer, vd., 2003: 15). YSA, insan beyninin karmaşık yapısından ilham alarak geliştirilen yenilikçi bir teknolojidir. İnsan beyninin öğrenme, hatırlama ve düşünme gibi temel işlevlerini gerçekleştiren sinir hücrelerinin özelliklerini model alarak, bu sistemler, karmaşık veri setlerinden öğrenme ve karar verme yeteneklerini taklit eder.

Nöral ağı işleyişi sırasında, ağ girdi verilerine ve beklenen çıktıya göre bağlantılarının ağırlıklarını ayarlamayı öğrenen bir eğitim sürecinden geçer. Bu, genellikle ağı tahminleri ile gerçek çıktılar arasındaki farkı en aza indiren bir optimizasyon algoritması, örneğin geri yayılım, ile gerçekleştirilir. Eğitildikten sonra, yapay sinir ağları, yeni girdi verilerini işleyerek, öğrenilen ağırlıkları ve aktivasyon fonksiyonlarını uygulayarak bir çıktı üretir. Bu süreç, yapay sinir ağlarının öğrenilen bilgilere dayalı olarak kararlar almasını ve kalıpları genelleştirmesini sağlar ve YSA, desen tanıma, veri madenciliği, robotik ve yapay zekâ gibi birçok alanda kullanılmaktadır. Kendi kendine öğrenme, kendi kendine organize olma ve gerçek zamanlı öğrenme gibi özellikleri sayesinde, geleneksel yapay zekâ ve bilgi işleme teknolojilerinin sınırlamalarının üstesinden gelirler. YSA, 1980'lerden bugüne kadar yapay zekanın en popüler konularından biri (Wu ve Feng, 2018: 1651). Bu popüler konu mobil manipülatörler için ise mobil platform ve monte edilmiş manipülatörün farklı dinamiklerini etkili bir şekilde koordine edilebilmesine destek olur. YSA, sistem dinamiklerini doğrudan tahmin ederek yapısal belirsizlikleri öğrenme yeteneğiyle aşar. Bu yaklaşım, kontrolün sağlamlığını artırır ve sistemdeki belirsizliklere ve dış etkilere karşı dayanıklılığı iyileştirir. Bu özellikler, mobil manipülatörlerin hassas ve etkin bir şekilde kontrol edilmesini sağlar, böylece karmaşık görevleri gerçekleştirme kabiliyetlerini artırır (Xu, vd., 2009: 791). Bir hücreye n adet veri girişi verilsin (X_n veri girişi), girilen verilerin ağırlıkları (W_{ij}) ile çarpılarak tüm veriler toplandıktan sonra önyargı eklenerek elde edilen sonuç net yargıyı verir. θ_j nöronun eşik değeridir. Net girdi aktivasyon fonksiyonundan geçirilerek bir veri çıktısı elde edilir. Bu işlemler denklem 2.16 ve 2.17'da verilmiştir. Şekil 2.7.'de bir yapay sinir hücresi görülmektedir.

$$net_j = \sum_{i=1}^n X_i W_{ij} - \theta_j \quad (2.16)$$

$$o_j = f(\text{net})_j \quad (2.17)$$



Şekil 2.7. Yapay Sinir Hücresi

Yapay sinir ağları (YSA) türlerinin sayısı, alan içindeki çeşitli uygulamaları ve mimarileri yansıtacak şekilde geniştir. Yaygın türler arasında ileri beslemeli sinir ağları, geri dönüşümlü sinir ağları, evrişimli sinir ağları ve radial temel fonksiyon ağları bulunmaktadır. Ayrıca, uzun kısa süreli hafıza ağları, derin inanç ağları ve kendiliğinden örgütlenen haritalar gibi özelleştirilmiş mimariler de vardır. Her tür, belirli zorlukları ve görevleri ele almak için tasarlanmış olup, sinir ağı modellerinin zengin manzarasına katkıda bulunur. Yapay sinir ağları (YSA), özellikle görüntü ve ses işleme, doğal dil işleme, tıbbi tanı ve sağlık hizmetleri, finans ve algoritmik ticaret, robotik kontrol, tahmin ve kestirim sistemleri ile ürün öneri ve kişiselleştirme alanlarında yaygın olarak kullanılmaktadır. Bu teknoloji, karmaşık veri desenlerini tanıma ve analiz etme yeteneği sayesinde, bu alanlarda önemli ilerlemelere yol açmıştır. Bu çalışmada öğrenme algoritması olarak; Matlab'ın toolbox'ı olan NNTOOL ve LSTM kullanılmıştır.

MSE (Mean Squared Error) ve RMSE (Root Mean Squared Error) istatistik ve makine öğrenmesinde sıklıkla kullanılan iki hata metriğidir. Her ikisi de model tahminlerinin gerçek değerlerden ne kadar farklı olduğunu ölçmek için kullanılır. MSE, tahmin edilen değerler ile gerçek değerler arasındaki farkların karelerinin ortalamasıdır, RMSE, MSE'nin kareköküdür ve bu nedenle

hataların ölçęi orijinal verilerle aynıdır. RMSE, model hatalarının ne kadar büyük olduğunu anlamak için daha sezgisel bir ölçü sağlar çünkü hataların birimi orijinal veri birimleriyle aynıdır.

MSE'nin gösterimi denklem 2.15'te ve RMSE'nin gösterimi de denklem 2.16'da verilmiştir; n verilerin sayısı, Y_i gerçek değerler, \hat{Y}_i tahmin edilen değerleri temsil eder.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.15)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (2.16)$$

MSE hataların karelerini alarak büyük hatalara daha fazla ağırlık verir. RMSE ise özellikle büyük hataların varlığında model hatalarının ne kadar büyük olduğunu anlamak için daha sezgisel bir ölçü sağlar.

2.3.1. NNTOOL

NNTool, MATLAB yazılımında bulunan bir grafiksel kullanıcı arayüzüdür ve Yapay Sinir Ağları (YSA) ile çalışmak için tasarlanmıştır. Bu araç, akademik ve araştırma amaçlı kullanımlar için sinir ağlarını kolaylıkla oluşturmak, eğitmek ve test etmek için geliştirilmiştir. NNTool, kullanıcıların veri setlerini yüklemelerine, farklı yapay sinir ağları modelleri oluşturabilmelerine ve bu modelleri eğitmelerine olanak tanır. Ayrıca, eğitilen modelleri test edebilme ve performanslarını değerlendirme imkânı sunar. Kullanıcılar, bu arayüz aracılığıyla sinir ağlarının mimarilerini (örneğin, katman sayısı, nöron sayısı) ve eğitim parametrelerini (örneğin, öğrenme oranı, momentum) özelleştirir. NNTool; çeşitli öğrenme algoritmalarını destekler, bu çalışma için kullanılan ise Levenberg-Marquardt Algoritması'dır.

Levenberg-Marquardt algoritması, yapay sinir ağları ve derin öğrenme alanında yaygın olarak kullanılan bir optimizasyon algoritmasıdır. Bu algoritma, genellikle ağırlık eğitiminde kullanılarak ağırlık hata fonksiyonunu minimize etmek için ağırlıkların güncellenmesinde kullanılır. Levenberg-Marquardt algoritması, genellikle hızlı bir şekilde yakınsama sağlayarak eğitim sürecini hızlandırır. Yu ve Wilamowski kitabında şu şekilde belirtmiştir; Levenberg-Marquardt algoritması, yapay sinir ağları ve derin öğrenme alanında yaygın olarak kullanılan bir optimizasyon algoritmasıdır. Bu algoritma, genellikle ağırlık eğitiminde kullanılarak ağırlık hata fonksiyonunu minimize etmek için ağırlıkların güncellenmesinde kullanılır. Levenberg-Marquardt algoritması,

genellikle hızlı bir şekilde yakınsama sağlayarak eğitim sürecini hızlandırır (Wilamowski ve Yu, 2011:12-6).

2.3.2. Uzun- Kısa Süreli Bellek LSTM

LSTM (Uzun-Kısa Süreli Bellek), özellikle zaman serisi verileri, doğal dil işleme ve diğer sekanssal veri kümeleri üzerinde çalışmak için tasarlanmış, gelişmiş bir Yapay Sinir Ağı (YSA) türüdür. LSTM, geri beslemeli sinir ağlarının bir alt sınıfıdır ve uzun süreli bağımlılıkları öğrenme ve saklama konusunda önemli avantajlar sunar.

Geleneksel yapay sinir ağları, "vanishing gradient" (kaybolan gradyan) problemi nedeniyle uzun dönemli bağımlılıkları öğrenmekte zorlanır. LSTM, bu problemi çözmek için tasarlanmıştır. Yapısında, veriler arasındaki uzun süreli ilişkileri korumak ve yönetmek için özel hücreler (LSTM hücreleri) bulunur. Bu hücreler, bilgiyi uzun süreler boyunca saklayabilir ve gerektiğinde unutma, güncelleme veya saklama gibi işlemleri gerçekleştirir.

LSTM hücreleri, üç ana bileşenden oluşur: unutma kapısı (forget gate), giriş kapısı (input gate) ve çıkış kapısı (output gate). Bu kapılar, hücrenin durumunu ve çıktısını kontrol eder, böylece ağın hangi bilgiyi saklayacağını veya atacağını belirler.

LSTM, karmaşık veri kümelerini modellemek için kullanıldığı gibi, ses tanıma, müzik üretimi, dil modelleme, metin üretimi ve makine çevirisi gibi birçok uygulamada da başarıyla kullanılmıştır. Bu özellikleri sayesinde, zamanla ilişkili verilerin işlenmesinde kritik bir rol oynar ve günümüzde derin öğrenme ve yapay zeka alanlarında yaygın olarak kullanılan bir teknolojidir.

Bu çalışmada ise mobil robot manipülatörlerde hata tespiti için yapay sinir ağı mimarisi kullanılmıştır. Bölüm 3'te yapay sinir ağı yapısı sistem YSA giriş sinyallerinin çıkarımından sonra tanıtılacaktır.

3. METOT

Bu bölümde mobil robot manipülatör için önerilen arıza tespiti yöntemi, Coppelia simülasyon ortamı, YSA giriş sinyallerinin çıkarımı, veri setinin oluşturulması, yapay sinir ağı, Neural Network Toolbox (NN), LSTM, Coppelia-Matlab haberleşmesi ve YSA ile duruş kestirimi anlatılmıştır.

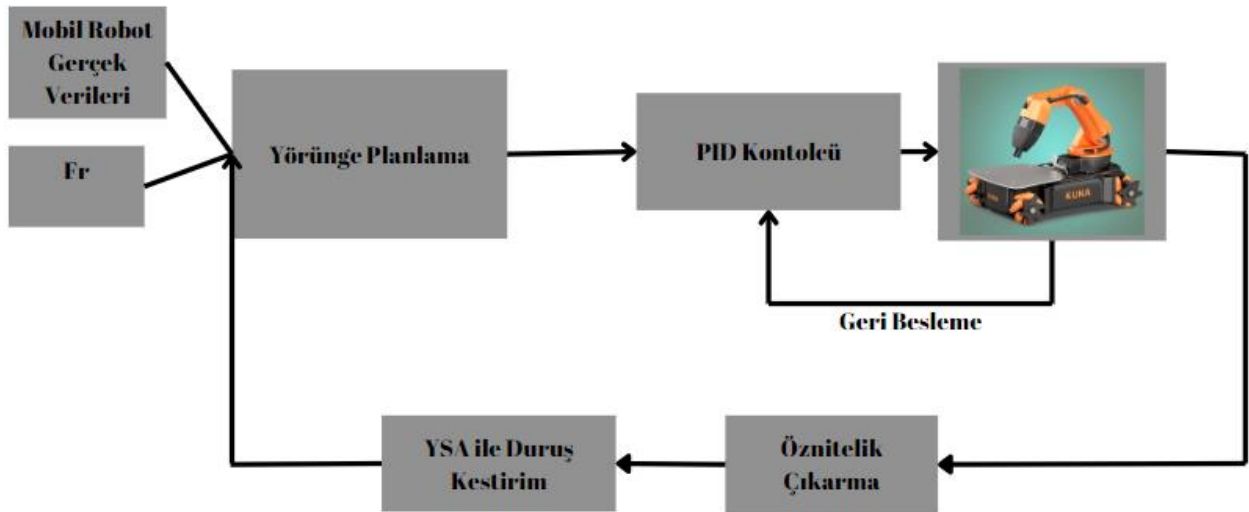
3.1. Önerilen Arıza Tespit Sistemi

Mobil manipülatörlerde arıza tespiti, robotun mekanik ve elektronik sistemlerindeki potansiyel sorunları belirlemek için kullanılan bir süreçtir. Arıza Tespit Sistemi (ATS), model tabanlı bir tekniktir, çünkü bu sistem, robotun veya makinenin beklenen davranışını ve performansını temsil eden bir matematiksel model kullanır. Bu model, normal çalışma koşulları altında cihazın nasıl davranması gerektiğini öngörür. Sistem gerçekteki performansı bu modele karşılaştırarak, herhangi bir sapmayı veya anormalliği tespit eder. Robotun eklemlerinin konumlarından gelen verilerle ATS döngüsü tamamlanır. Bu aşamada elde edilen eklemlerin konum verileri, mobil robotun 5+4 eklemi için yapılır. 5 eklem şu şekildedir; baz (veya taban) eklemi, omuz eklemi, dirsek eklemi, bilek eğilme/bükülme eklemi ve bilek dönme eklem ve +4 olarak adlandırılan eklemler ise robotun tekerleklerinin konum hareketidir, yani toplanda 9 eklem olarak alınmıştır. Sistemde 9 eklem de koordinatlarının zamana bağlı hareketinin eklenmesi lazım. Sisteme üç koordinat çerçevesinin tanımlanması gerekir; Mevcut Robot Çerçevesi (F_r), Robot manipülatörünün şu anki duruşunu ve konumunu temsil eden koordinat sistemi. Bu çerçeve, robotun anlık hareketlerini ve mekânsal konumunu analiz etmek için kullanılır. Hedef Robot Çerçevesi (F_t), Robotun ulaşması gereken nihai duruş ve konum. Bu çerçeve, robotun hedeflediği son durumu gösterir ve arıza tespiti için ideal duruş ile mevcut duruş arasındaki sapmaları belirlemek için kullanılır. İşlem Hedefi Çerçevesi (F_p), robotun gerçekleştirmesi gereken spesifik görev veya işleme odaklanan bir koordinat sistemi. Bu çerçeve, manipülatörün hedeflediği işlemlerin doğruluğunu ve etkinliğini analiz etmek ve bu süreçlerdeki potansiyel arızaları tespit etmek için kullanılır.

Mobil manipülatörde arıza tespitinde, eklemlerin kalibrasyonu ve modelleme kaynaklı hataların varlığında daha iyi yerel kararlılığa ve yakınsamaya sahiptir. Bununla birlikte, sistemler genellikle küresel yakınsama sorunlarıyla karşılaşır, yani belirli başlangıç koşulları veya parametreler altında optimal veya istenen sonuca ulaşma konusunda zorluklar yaşanır. Bu durum,

sistemlerin tüm olası durumlar için istikrarlı ve tutarlı çözümlere ulaşamaması anlamına gelir ve bu, özellikle karmaşık veya değişken sistemlerde önemli bir kısıtlama olarak ortaya çıkar. Örneğin; bir manipülatörün otomatik olarak bir nesneyi tanıma ve ona doğru hareket etme görevi ele alınır. Eğer manipülatörün algılama ve hareket sistemi yeterince genelleştirilmemişse, belirli nesnelere veya belirli çevresel koşullar altında başarısız olur. Örneğin, manipülatörün nesneyi tanıma algoritması, sadece belirli aydınlatma koşullarında veya belirli bir arka plana karşı iyi çalışıyorsa, bu, küresel yakınsama sorunlarının bir örneği olur. Bu tür durumlar, manipülatörün çeşitli ortamlarda ve koşullarda etkili ve tutarlı performans göstermesini engeller.

Bu çalışmada mobil robot manipülatörün (KUKA youBot) tüm gerçek verileri kullanılmıştır. Bu sayede gerçek ile modelin eşleşip eşleşmediği kontrol ediliyor. Şekil 3.1.'de ATS sisteminde r_{ξ} 'ye göre şu anki durumu ve konumu tahmin edilir. Ulaşması gereken konum t_{ξ} 'ye göre ulaşması istenilen konum belirlenir. Yörünge planlama aşamasında, YSA'da başlangıç ve bitiş noktaları sağlanmış ve bu iki nokta arasındaki yörünge, üçüncü dereceden bir polinom kullanılarak oluşturulmuştur.



Şekil 3.1. Önerilen Arıza Tespit Sistemi

3.2. CoppeliaSim Simülasyon Ortamı

Simülasyon aşaması, robotik sistemlerin ve algoritmaların geliştirilmesi ve test edilmesi için yaygın olarak kullanılan bir platform olan Coppeliasim'de gerçekleştirilmiştir. Bu simülasyon ortamı, karmaşık mekanik yapıların ve hareket kontrol algoritmalarının doğruluk ve etkinlik açısından değerlendirilmesine olanak tanıyarak, gerçek dünya koşullarını yakından taklit edebilen bir test yatağı sunar. Coppeliasim'in sunduğu gelişmiş fizik motoru ve çeşitli robot modelleri, KUKA youBot'u kapsamlı bir şekilde incelenmesi ve geliştirilmesi için temel oluşturmuştur.

Coppeliasim, önceden V-REP olarak bilinen, robotik simülasyonlar için güçlü ve esnek bir platformdur. Fiziksel robotik sistemlerin ve algoritmaların prototipleme, test etme ve doğrulama süreçlerini desteklemek için tasarlanmıştır. Simülasyon ortamı, gerçekçi fizik motorları ile donatılmıştır ve kullanıcıların, sensörler, aktüatörler ve diğer robotik bileşenlerle etkileşimli ve dinamik ortamlar oluşturmasına olanak tanır. Coppeliasim, çoklu robotların koordineli hareketlerini ve karmaşık mekanizmaların işleyişini simüle etme yeteneğine sahiptir, ayrıca gerçek zamanlı kontrol algoritmalarının ve yapay zekâ modellerinin entegrasyonunu mümkün kılar. Coppeliasim, Lua programlama dilini birincil dil olarak kullanır. Lua, kullanım kolaylığı ve performansı nedeniyle tercih edilir. Aynı zamanda Coppeliasim Uygulama Programlama Arayüzü (API)'ya sahiptir. API; yazılım bileşenleri arasında belirli bir protokol veya araç seti sağlayarak, farklı programların birbirleriyle etkileşimde bulunmalarını ve veri alışverişi yapmalarını mümkün kılar. Bununla birlikte, Coppeliasim Python ile de uyumlu bir yapıya sahiptir ve genellikle dışarıdan Python betikleri aracılığıyla robotik simülasyonları kontrol etmek için kullanılır. Ayrıca, Coppeliasim'in API'si aracılığıyla C++, Java, Matlab gibi diğer dillerle de entegrasyon sağlanır ve bu dillerden simülasyonu kontrol etmek mümkündür. Bu çok dilli destek, kullanıcıların kendi ihtiyaçlarına göre özelleştirilmiş simülasyonlar yaratmasına olanak tanır ve Coppeliasim'i farklı disiplinlerdeki araştırmacılar ve mühendisler için erişilebilir bir araç haline getirir. Coppeliasim'de API, kullanıcıların simülasyon ortamında programatik olarak etkileşimde bulunmalarına olanak tanır. İki temel API türü bulunmaktadır; Uzak API (Remote API) ve Komut Dizisi Oluşturma API (scripting API).

Uzak API (Remote API); ağ üzerinden Coppeliasim'e bağlanarak simülasyonu kontrol etmek için kullanılır. Uzak API sayesinde, kullanıcılar Matlab, Python, Java, C++ gibi dillerde yazdıkları harici programlar aracılığıyla simülasyonu yönetir. Uzak API, simülasyonla ilgili

komutları göndermek ve simülasyon durumunu sorgulamak için bir ağ soketi üzerinden haberleşme kurar. Komut Dizisi Oluşturma API; coppeliasim, Lua programlama dili için güçlü bir betikleme desteği sunar. Lua betikleri doğrudan simülasyon ortamında çalışır ve robotlar, sensörler ve diğer simülasyon nesnelere üzerinde doğrudan kontrol sağlar.

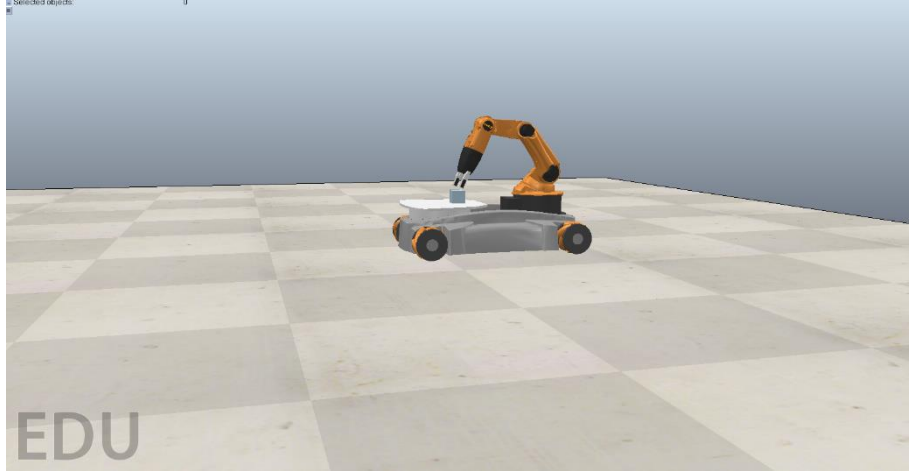
Örneğin; Rodriguez-Tirado ve diğerleri tarafından yapılan çalışmada; CoppeliaSim yazılımı kullanılarak bir labirentin 3 boyutlu modeli oluşturulmuş ve robotun bu labirenti geçişi simüle edilmiştir (Rodriguez-Tirado, vd., 2020: 154).

3.2.1.CoppeliaSim ve Matlab Haberleşmesi

CoppeliaSim, belirlenen simülasyonu harici bir uygulamadan kontrol edilmesini ve veri akışının yapılması için bir API'ya sahiptir. Bu çalışma, MATLAB'ın analitik ve işleme gücünün, Coppeliasim'in simülasyon kabiliyetleri ile birleştirilmesi sayesinde, karmaşık robotik sisteminin test edilmesi ile oluşturulmuştur.

Coppeliasim ve MATLAB arasındaki entegrasyon, Coppeliasim'in Uzak API özelliği kullanılarak gerçekleştirilir. Bu işlem, öncelikle Coppeliasim tarafında Uzak API sunucusunun etkinleştirilmesini ve ardından MATLAB tarafında bir istemci olarak bu sunucuya bağlanılmasını içerir. Coppeliasim'de, simülasyon sahnesi içerisinde bir Uzak API sunucusu nesnesi eklenir ve belirli bir port üzerinden dinlemeye ayarlanır. MATLAB tarafında ise, Coppeliasim'in sağladığı Uzak API kütüphanesi içindeki fonksiyonlar ve arayüzler kullanılarak, belirlenen porta TCP/IP üzerinden bağlantı kurulur. Bu bağlantı kurulduktan sonra, Matlab'dan giriş olarak verilen; eklem hareketlerinin ve mobil robot hareketlerinin motor hızları CoppeliaSim tarafından işlenir ve LUA tarafından uygulanır. Matlab'dan uygun motor hızlarını alan LUA, simülasyondaki tanımlanan eklemlere uygun motor girişlerini yollar. LUA ivmeölçer ve jiroskop verilerini okur ve tamamlayıcı filtre uygulayarak yuvarlanma ve eğim açılarını hesaplar. İşlenen verilerin görselleştirilmesi hem Simulink/Matlab hem de CoppeliaSim uygulamasında yapılmaktadır (Khalilov, 2016).

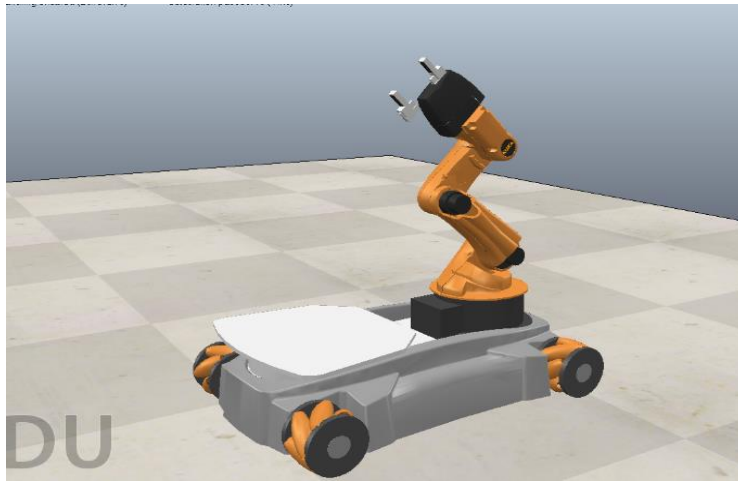
Tamamlanan senaryoya ait bir CoppeliaSim görüntüsü Şekil 3.2.'de verilmiştir. Burada mobil robot manipülatör olarak Robotics Toolbox'ta tanımlı olan Kukka youBot kullanılmıştır. youBot'un beş eksenli manipülatörünü, dört bağımsız tekerleği ve bunlara bağlı motorları içerir. Her eksen ve tekerlek, Coppeliasim'in sağladığı kontrol mekanizmaları aracılığıyla ayrı ayrı hareket ettirir.



Şekil 3.2. Coppeliasim Senaryo Simüle Etme Ortamı

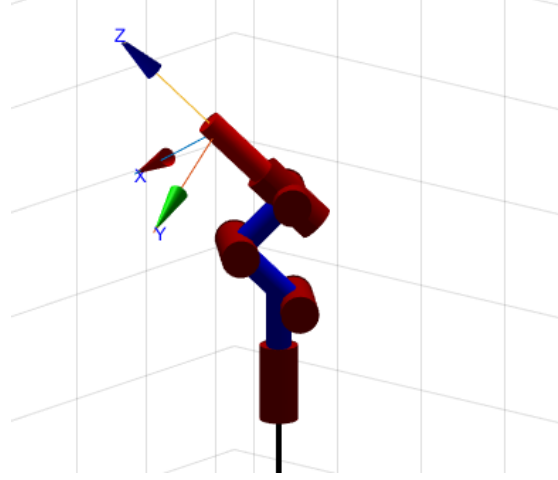
3.3. Sistem YSA Giriş Sinyallerinin Çıkarımı

Önerilen sistemde YSA giriş sinyallerini içeren; eklem açısı bilgisi gibi YSA için veri seti oluştururken, bu süreç özellikle önemlidir çünkü YSA'lar genellikle girdi olarak işlenmiş, anlamlı YSA giriş sinyallerine ihtiyaç duyar. Şekil 3.3'te YSA giriş sinyalleri çıkarımı için simüle edilen görüntü verilmiştir.



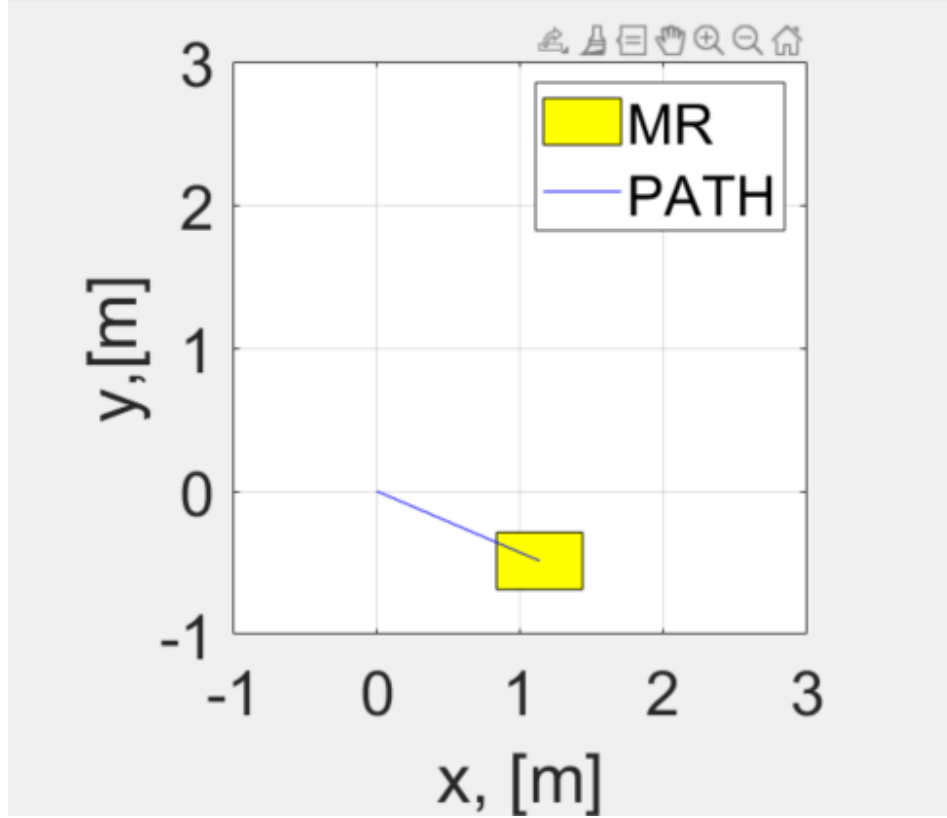
Şekil 3.3. YSA Giriş Sinyalleri Çıkarımı için Simüle Edilen Görüntü

Daha sonra, simüle edilen adım sonrasında Matlab’da ilgili koordinatlar girilmiştir ve simüle edilen koordinatlar Matlab’a aktarılmıştır. Matlab’a koordinatların aktarıldığı görüntü Şekil 3.4.’te verilmiştir.



Şekil 3.4. Matlab’a Koordinatların Aktarıldığı Görüntü

Mobil platform için de konumlanan mobil platformdaki hareket değişimi Şekil 3.5’te gösterilmiştir. Şekil 3.5.’te mobil robot manipülatörün son konumu gösterilmiştir ve bilgiler CoppeliaSim’den alınıp, robotun tekerleklerinin açısal hızlarına dayanarak robotun hareketini ve yörüngesi hesaplanmıştır. Matlab’da koordinatları aktarılmış haldedir. Dört tekerleğin açısal hızları sabit olarak tanımlanır ve bu hızlar, tekerlek konfigürasyon matrisi ile çarpılarak robotun genel hız vektörüne dönüştürülür. Daha sonra, zaman türevi koordinatları hesaplanır ve Euler metodu kullanılarak robotun pozisyonu aktarılır.



Şekil 3.5. Mobil Platformdaki Hareket Değişikliği

3.4. Veri Setlerinin Oluşturulması

Veri setleri oluşturulurken bölüm 3.2’de bahsedilen YSA giriş sinyallerinin çıkarımı yapıldıktan sonra, 5 serbestlik derecesine ait eklemin ve 4 tekerleğin ayrı olarak YSA giriş sinyalleri bulunmuştur. İlk olarak, robot kolunun her bir ekleminin açısal konumları (q_1 ila q_5) radyan cinsinden belirlenmiştir. Bu açılar, robot kolunun özgün duruşunu temsil etmektedir. Daha sonra, bu açı değerleri 'joint_pos2' adlı bir diziye atanır. Bu dizi, robot kolunun eklemlerinin hedef konumlarını içermektedir. Robot kolunun her bir Link nesnesi, bu açı değerleri ve her bir eklemlerle ilgili fiziksel özellikler (eklem uzunlukları ve büküm açıları) kullanılarak oluşturulmuştur. Link nesneleri, robotun seri bağlantılı yapısını ve her eklem arasındaki ilişkiyi modellemek için kullanılır.

Robot kolunun başlangıç duruşu ile sıfır açılar olarak belirlenmiş ve bu, yörünge hesabı için başlangıç vektörü olarak kullanılmıştır. Yörünge hesabı, robot kolunun başlangıç duruşundan

hedef duruşuna olan geçişini hesaplar ve bu geçiş sırasındaki ara konumları, hızları ve ivmeleri üretir.

İleri kinematik fonksiyonu kullanılarak, robot kolunun uç işlevcisinin uzaydaki pozisyonu hesaplanır ve bu pozisyon 'koordinat' değişkeninde saklanır. 'x', 'y', 'z' değişkenleri, uç işlevcisinin koordinat sistemdeki konumlarını temsil eder. Gerçek dünya koordinatlarına dönüştürmek için küçük ofsetler eklenerek ('x_reel', 'y_reel', 'z_reel') robot kolunun gerçek dünya koordinatları elde edilir.

Son olarak, yörünge hesabı için üretilen veriler, gelecekteki analiz ve simülasyon çalışmaları için '.mat' formatında kaydedilir. Her eklem için hız verileri MATLAB dosyaları olarak dışa aktarılır. Bu veriler, daha sonra Simulink modelinde veya diğer simülasyon ortamlarında kullanılmak üzere hazır hale getirilmiştir. Her bir eklem için oluşturulan zaman boyunca açısal hız simulink'e eklenmek üzere hazırdır.

Mobilite kısmı için ise; her simülasyon döngüsünde robotun anlık yön açısı hesaplanır. Bu açı, robotun yerel koordinatlardan küresel koordinatlara dönüşümünü sağlayan Jacobian matrisinin oluşturulmasında kullanılır. Tekerleklerin açısal hızları sabit değerlerle tanımlanmış ve bu hızlar, tekerlek konfigürasyon matrisi aracılığıyla robotun lineer ve açısal hız vektörüne dönüştürülmüştür. Ayrıca, tekerleklerin her birinin x ve y koordinatları, robotun hareketi sırasında her köşesindeki tekerlek konumlarını temsil edecek şekilde ayrı ayrı hesaplanmış ve bu bilgiler, Simulink modeline aktarılacak biçimde kaydedilmiştir. Bu entegrasyon, robotun kontrol sistemlerinin daha ileri düzeyde analiz ve tasarımı için bir temel oluşturur.

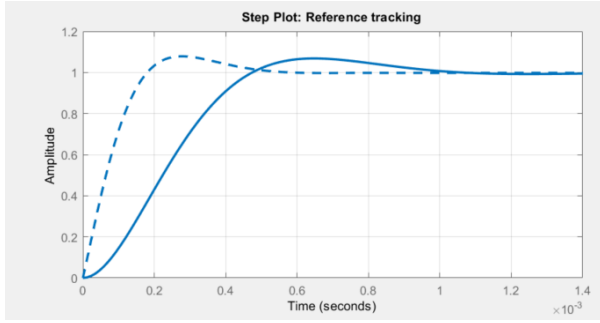
Simulink için hazır olan veri setleri simulink'e aktarılır. Simulink'te ise; her bir eklem için oluşturulan veri setleri bir PID kontrolcüsüne eklenir, örnek model Şekil 3.6'da gösterilmiştir. Model, bir PID kontrol mekanizması kullanarak, robotun eklem hareketlerini hassas bir şekilde yönetmeyi amaçlamaktadır. Modelin başlangıcında, kontrol döngüsü için gerekli olan girdi sinyali sağlanmaktadır. Bu referans sinyal, robot eklemının istenen pozisyonunu temsil eder. Ardından, bir toplama bloğu aracılığıyla, referans sinyali ile mevcut sistem çıktısı arasındaki fark hesaplanır ve bu hata sinyali PID kontrolörüne iletilir.

Tablo 3.1. Eklemlerin J ve B Karakteristikleri

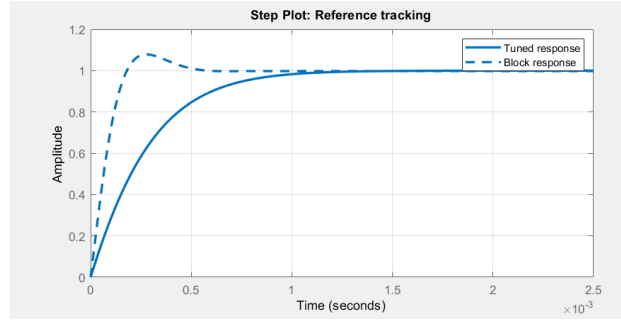
Eklemlerin Adı	J Değeri	B Değeri
1. Eklemler	0.006	40
2. Eklemler	0.006	40
3. Eklemler	0.006	20
4. Eklemler	0.0814	8
5. Eklemler	0.0734	20
6. Eklemler (Sol ön tekerlek)	0.006	40
7. Eklemler (Sol arka tekerlek)	0.006	40
8. Eklemler (Sağ ön tekerlek)	0.006	40
9. Eklemler (Sağ arka tekerlek)	0.006	40

Otomatik PID'ye giren ve çıkan verilerin grafiği Şekil 3.7.'de gösterilmiştir. Grafik, kontrolörün referans bir sinyali takip etme kabiliyetini iki farklı yanıt ile karşılaştırıyor. Mavi kesikli çizgi (Block response), otomatik ayar öncesindeki orijinal PID parametrelerine sahip sistem yanıtını gösterirken, katı mavi çizgi (Tuned response), otomatik ayar sonrası elde edilen optimize edilmiş sistem yanıtını temsil eder. Grafikte görülen 'Step Plot: Reference tracking' başlığı, bu grafiğin referans bir sinyale karşı sistem yanıtını izlediğini gösterir. Adım cevabı, sistemin bir adım sinyaline nasıl tepki verdiğini ve zaman içindeki genliğini (Amplitude) gösterir. X eksenini zamanı (saniye cinsinden), Y eksenini ise sistem çıkışının genliğini ifade eder. Otomatik ayar işlemi, PID kontrolörünün daha hızlı ve daha az salınım yaparak istenen değere ulaşmasını sağlamıştır.

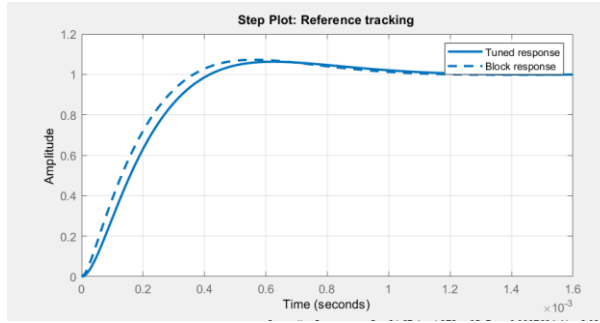
Optimize edilmiş yanıt, orijinal blok yanıtına kıyasla daha az aşım (overshoot) ve daha kısa bir yerleşme süresine (settling time) sahip olduğu için, sistem performansında önemli bir iyileşme gösterir. Bu, kontrol sistemlerinin performansını artırmak için kritik bir süreçtir ve kontrolörün daha hızlı ve daha doğru bir şekilde referans sinyali takip etmesini sağlar, tüm eklemler için otomatik ayar uygulanmıştır.



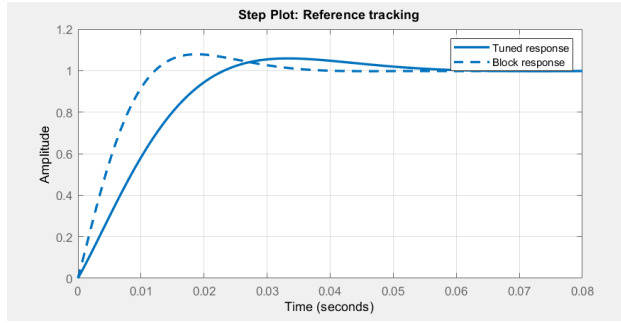
a)



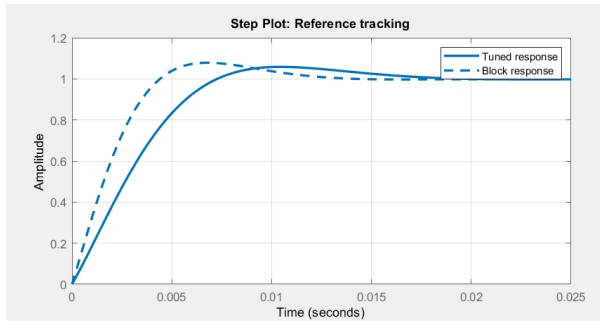
b)



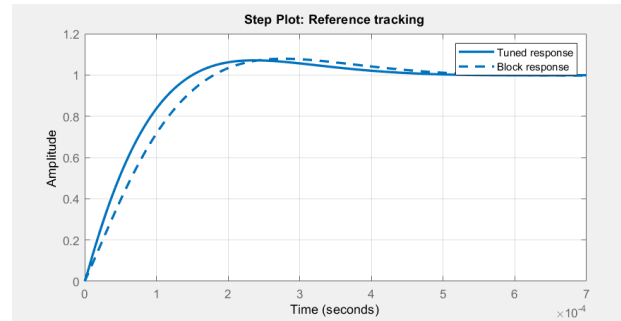
c)



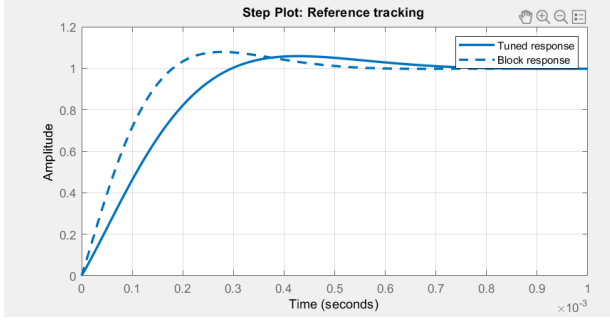
d)



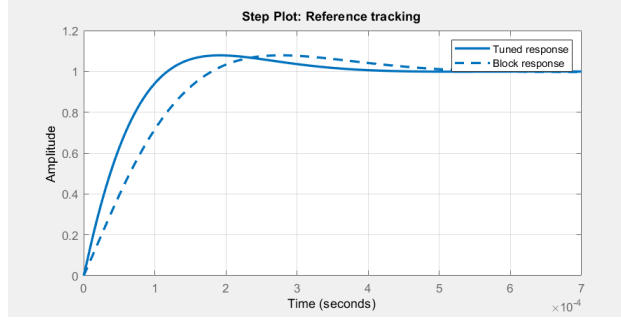
e)



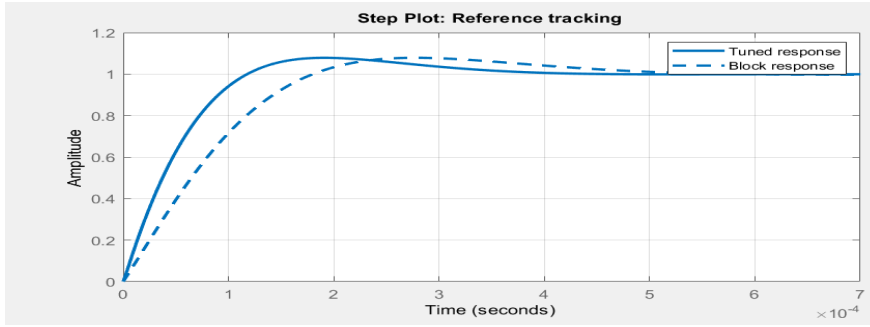
f)



g)



h)



i)

Şekil 3.7. PID'ye Otomatik Ayar'a Giren Verilen Grafiği

- | | | | | |
|-------------|-------------|-------------|-------------|-------------|
| a) 1. Eklem | c) 3. Eklem | e) 5. Eklem | g) 7. Eklem | i) 9. Eklem |
| b) 2. Eklem | d) 4. Eklem | f) 6. Eklem | h) 8. Eklem | |

Şekil 3.7'de a'da bulunan 1. Eklem için, otomatik ayar öncesinde sistem, hedef değere ulaşmak için daha uzun bir süre ve daha fazla aşım gösteriyor. Bu durum, orijinal PID parametrelerinin sistemin dinamiklerine tam olarak uymadığı anlamındadır. Autotune sonrası, sistem çok daha hızlı bir şekilde istenilen referans değere ulaşıyor ve aşım neredeyse ortadan kalkmış görünüyor ve yerleşme süresi de önemli ölçüde kısalıyor. Şekil 3.7'de b'de bulunan 2. Eklem için ise; oransal ve türevsel etkiler artırılmış, böylece sistem daha hızlı yanıt verirken, integral etkisi azaltılmış ki bu da sistem yanıtını kararlaştırmış ve aşımı önlemiştir. Şekil 3.7'de c'de bulunan 3. Eklem için, stabilize olma süresi kısaltılmıştır. Bu, otomatik ayarlama işlemi sırasında, kontrolörün yanıtını hızlandırmak ve stabilitesini artırmak için P ve D parametrelerinin artırıldığını ve I parametresinin düzgün bir geçiş sağlayacak şekilde ayarlandığını gösterir. Şekil 3.7'de d'de bulunan 4. Eklem için ise; stabilize olma süresinin kısalması ve aşımın neredeyse sıfır

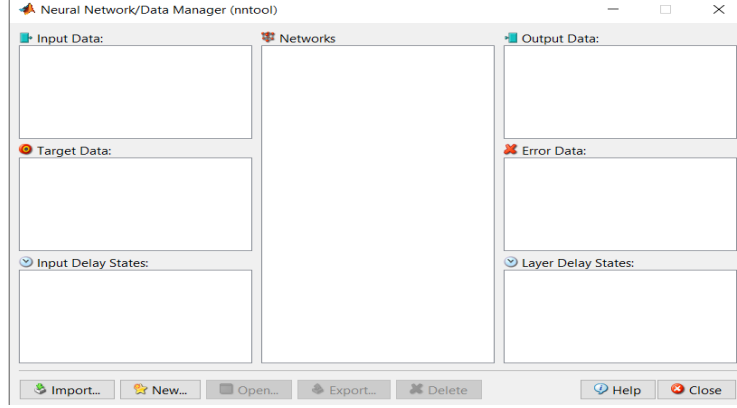
olması bu parametrelerin sistemin istenen performansını sağlamak için doğru bir şekilde ayarlandığını gösterir. Şekil 3.7'de e'de bulunan 5. Eklem için, sistem yanıtını hızlandırarak ve istenmeyen aşımı azaltarak, kontrol performansını önemli ölçüde iyileştirmiştir. Şekil 3.7'de f'de bulunan 6. Eklem, için; kontrolör parametrelerinin başlangıçta sistem dinamiklerine uygun olmadığını ve bu nedenle hedefe ulaşma sürecinin istenenden daha uzun sürdüğünü gözlemlenmektedir. Auto tune sonrasında daha dik bir yükselme eğrisi sergileyerek hedef değere çok daha hızlı ulaşıyor. Bu, otomatik ayarlama işlemiyle PID parametrelerinin, sistem performansını iyileştirecek şekilde ayarlandığını göstermektedir. Şekil 3.7'de g'de bulunan 7. Eklem, için; hedef değere çok daha hızlı ulaşıyor ve aşımı neredeyse yok. Bu, PID parametrelerinin otomatik ayarlama işlemi ile sistemin tepkisini hızlandırmak ve kararlılığını artırmak için etkin bir şekilde ayarlandığını gösterir. Şekil 3.7'de h'de bulunan 8. Eklem, için; sistemin daha hızlı yanıt vermesini ve aşımı önleyerek daha kararlı olmasını sağlamıştır. Şekil 3.7'de ı'da bulunan 9. Eklem, için ise; hedef değere daha hızlı ulaşmakta ve neredeyse hiç aşım olmadan kararlı bir duruma geçmektedir.

Çıkış olarak alınan veri setinde ise; giriş verisi referans sinyali olarak hedef değeri sağlayan giriş bloğundan gelen veridir. Bu veri, kontrol sisteminin neyi başarmaya çalıştığını yani hedeflenen çıktıyı temsil eder. PID çıkışından elde edilen veri ise; PID kontrolöründen gelen sinyal, sistemin hedefe ulaşmak için ne tür bir düzeltme yaptığını gösterir. Bu, kontrolörün hata sinyaline yanıt olarak ne kadar güç uyguladığını gösterir. Transfer fonksiyonu çıkışı; sistemin dinamik davranışını temsil eder. Bu satır, kontrol sinyalinin sistemin fiziksel yapısı tarafından nasıl işlendiğini gösterir. İntegral bloğu çıkışı; hızdan pozisyona geçiş yapmak için kullanılır. Bu satır, kontrol sinyalinin zaman içinde birikimini temsil eder ve böylece sistemin pozisyonunu verir. PID içindeyken arıza simülasyonu yaratmak amacıyla %30, %40, %50, %60, %70 oranında her bir eklem ve tekerlek için arızalar veriliyor ve bu arızaların YSA duruş kestirimci veri setine dahil ediliyor. Eğer, veriler hatasız ise verilen yüzde aynı şekilde alınıyor fakat veride sapma var ise sapma yeni veri setine dahil oluyor.

Önerilen sistemde toplamda 225 farklı deneme yapılmıştır. Giriş verisi olarak her denemede 45 adet 2x10'lik bir giriş matrisi elde edilmiştir. 225 deneme 5 senaryo halinde ayrılmıştır. Çıkış verisi olarak 5x7088'lik çıkış verileri elde edilmiştir ve toplamda her 9 eklem için 5 farklı senaryo içermektedir. Önerilen sistemde, YSA tarafından kullanılan duruş kestirimcisinin giriş verileri için oluşturulan veri setlerine ilişkin ayrıntılı bilgiler, sonraki bölümlerde açıklanmıştır.

3.5. Neural Network Toolbox

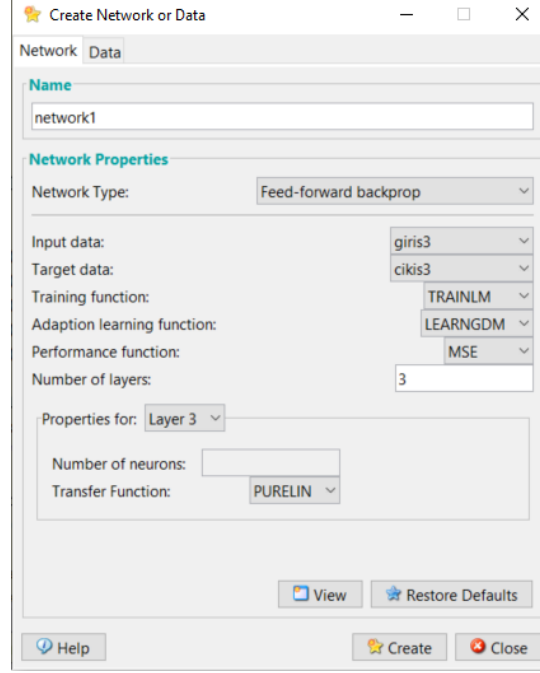
Bu sistemde duruş kestirimi için kullanılan sinir ağlarından birisi Matlab arayüzü olan Neural Network Toolbox (NNT) kullanılmıştır. NNT'yi çağırarak için komut ekranına "nntool" yazılması gerekmektedir ve Şekil 3.8.'de NNT'nin kullanıcı arayüzü verilmiştir.



Şekil 3.8. NNT Kullanıcı Arayüzü

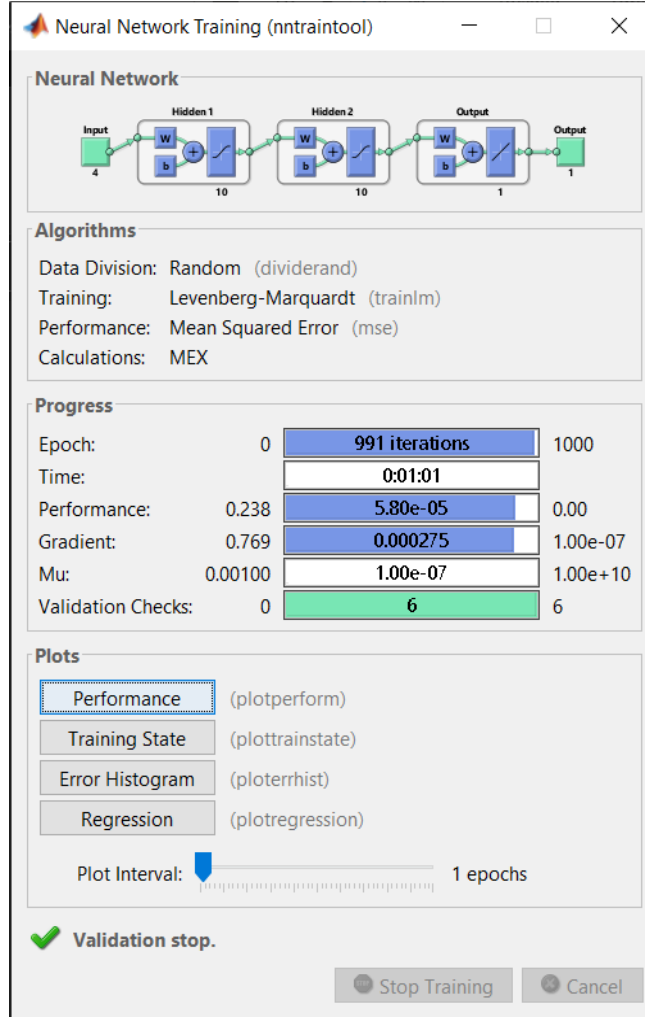
Yapay sinir ağının inşa sürecine başlamak üzere, ilk olarak, Import seçeneği aracılığıyla giriş ve çıkış verilerinin arayüze dahil edilmesi gerekmektedir. Bu seçenek işaretlendiğinde, Import to Network/Data Manager arayüzü kullanıcıya sunulur. MATLAB çalışma alanında bulunan vektör veya matris formatındaki giriş ve çıkış verileri, "import from MATLAB workspace" opsiyonu kullanılarak sisteme aktarılır. Bu aşamada, giriş verileri 'input data', çıkış verileri ise 'target data' olarak tanımlanır ve arayüzün sağ alt kısmında yer alan Import butonu ile süreç tamamlanır. Yapay sinir ağını oluşturmak için sonrasında New butonu seçilir.

Şekil 3.9'da Create Network or Data arayüzünün bir görseli yer almakta. Bu arayüzde, kullanıcı giriş verilerini 'input data', çıkış verilerini 'target data' alanlarına yerleştiriyor. Ağ eğitimi için Levenberg-Marquardt algoritması (TRAINLM) kullanılıyor ve uyarılma öğrenme fonksiyonu olarak LEARNGDM tercih ediliyor. Bu çalışmada, katman sayısı üç olarak belirlenmiş ve ilk iki katman gizli katman olarak adlandırılmış. Her iki gizli katman için nöron sayısı 10 olarak ayarlanmış ve aktarım fonksiyonu olarak LOGSIG kullanılmış. Çıkış katmanının katman sayısına herhangi bir müdahalede bulunulmamış ve aktarım fonksiyonu olarak PURELIN seçilmiş. Ağ yapılandırması tamamlandıktan sonra 'Create' butonu ile yapay sinir ağı oluşturulmuş.



Şekil 3.9. Create Network or Data Arayüzünün Bir Görseli

YSA duruş kestirimcisine ilişkin giriş ve çıkış matrislerinin, tüm eklemlere ait verileri içerecek şekilde giriş ve çıkış verileri olarak sağlandığı belirtilmiştir. Şekil 3.10, Bölüm 3.4'te oluşturulan veri setlerinden yararlanarak duruş kestiriminde kullanılan yapay sinir ağı yapısını 1. eklem için görsel olarak sergilemektedir. Şekil 3.10'da gösterilen eğitim penceresi, dört giriş birimi ve iki gizli katman içeren bir yapay sinir ağı modelini tanıtmaktadır. Gizli katmanlar sırasıyla on nörona sahiptir ve model tek bir çıkış birimi ile sonuçlanmaktadır. Performans skoru olarak $5.30e-05$ değeri elde edilmiş, bu düşük bir hata oranı göstermekte ve modelin yüksek bir doğrulukla eğitildiğini işaret etmektedir. Gradyan değeri 0.769 'dan 0.000275 'e düşmüştür, bu da modelin hata yüzeyinde düşük gradyan değerlerine ulaştığını gösterir, yani model hedef fonksiyonun minimumuna yakınsamaktadır. Eğitim sürecinde 'Mu' değeri, yani düzenleme parametresi, 0.00100 'dan $1.00e-07$ 'ye ayarlanarak modelin daha iyi genelleme yapması sağlanmıştır. Doğrulama kontrolleri altı kez yapılmış ve bu sınır aşıldığında eğitim otomatik olarak durmuştur.

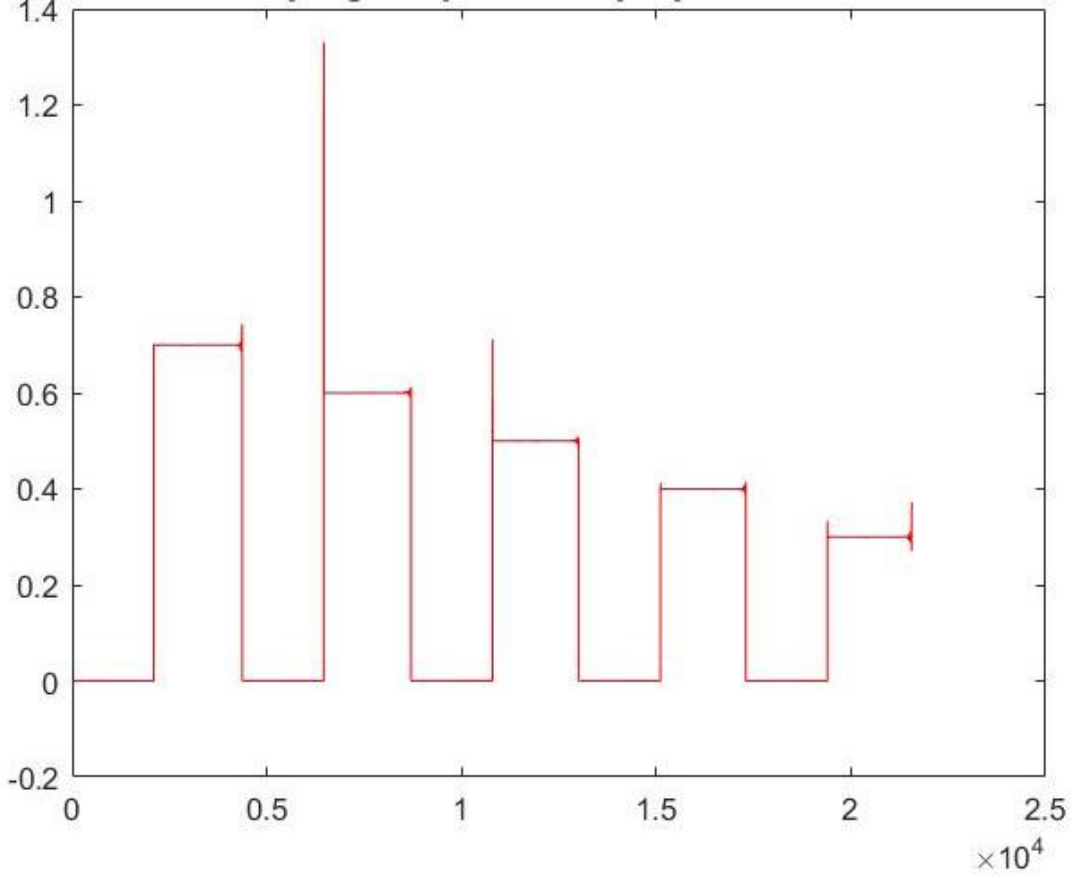


Şekil 3.10. Bölüm 3.4'te oluşturulan veri setlerinden yararlanarak duruş kestiriminde kullanılan yapay sinir ağı yapısını 1. eklem için görsel olarak sergilemektedir

Bu arayüz, kullanıcıya yapay sinir ağı modelinin eğitim sürecini interaktif bir şekilde gözlemleme ve değerlendirme imkânı sunar. NNTOOL'un sağladığı bu işlevsellik, önerilen sistemde ele alınan yapay sinir ağı modelinin geliştirilmesi ve eğitimi sürecinde önemli bir araç olarak hizmet etmiştir. Bu aracın kullanımı, modelin performansını artırma ve öğrenme sürecini hızlandırma potansiyeline sahiptir.

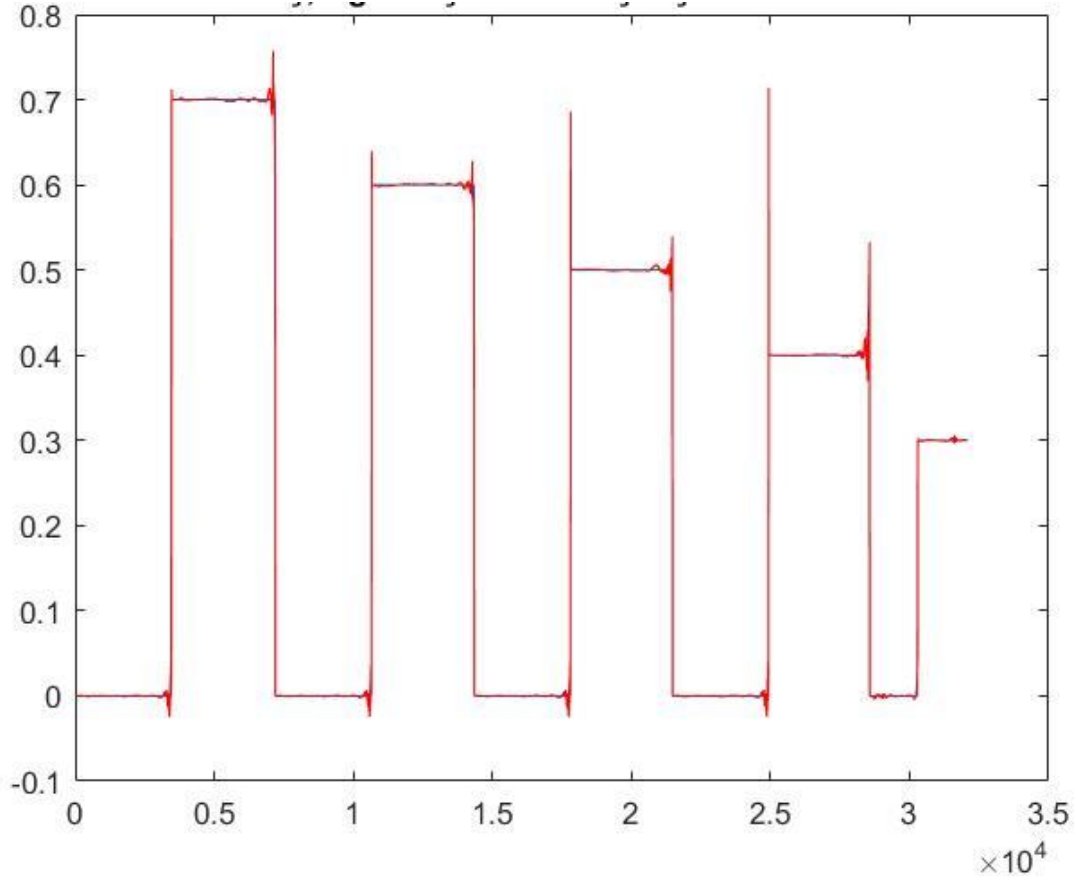
1. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği Şekil 3.11.'de verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Grafikteki genliklerin düzgün bir şekilde dağılmadığı ve bazı

yüksek tepe noktalarının varlığı, modelin belirli girişlere karşı aşırı tepki gösterebileceğinden ya da eğitim veri setindeki gürültüden etkilenebileceğinden dolayıdır.



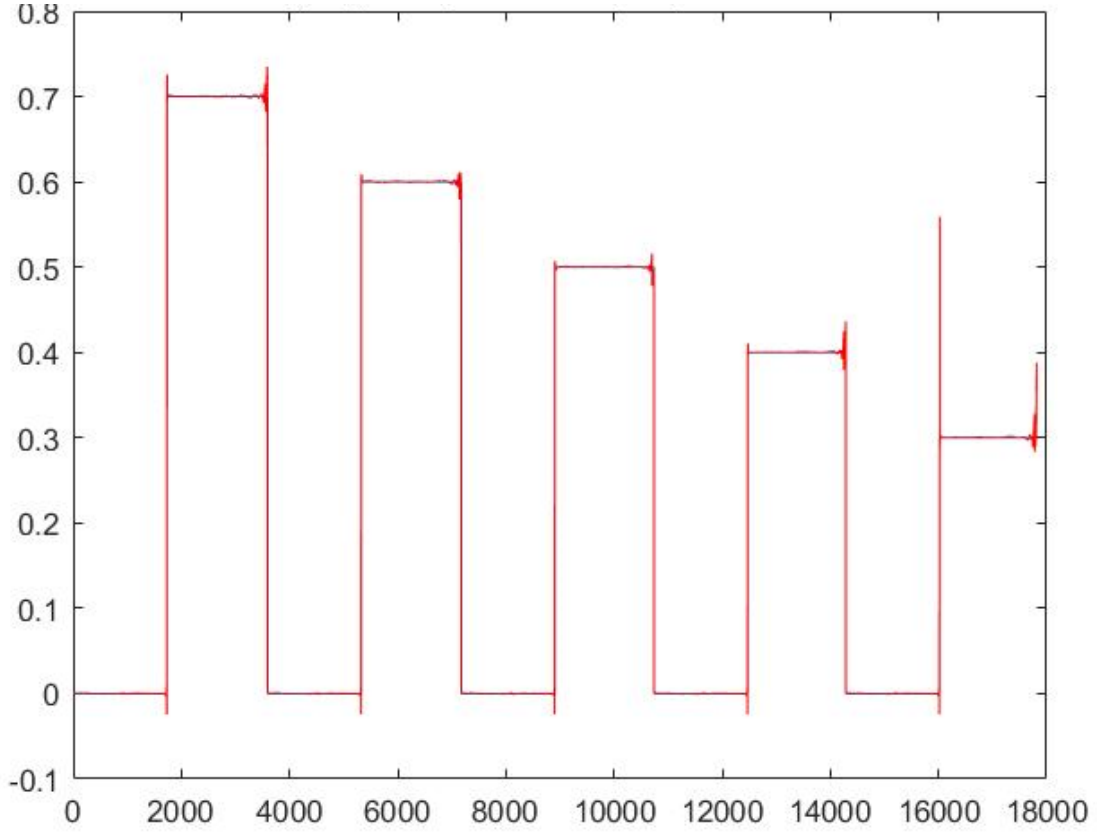
Şekil 3.11. NN'nin 1. Eklem için Yakınsaması

Şekil 3.12.'de 2. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Burada yakınsamanın olduğunu fakat eğitim setinin sonlarına doğru yakınsamadan sapmalar görülmektedir.



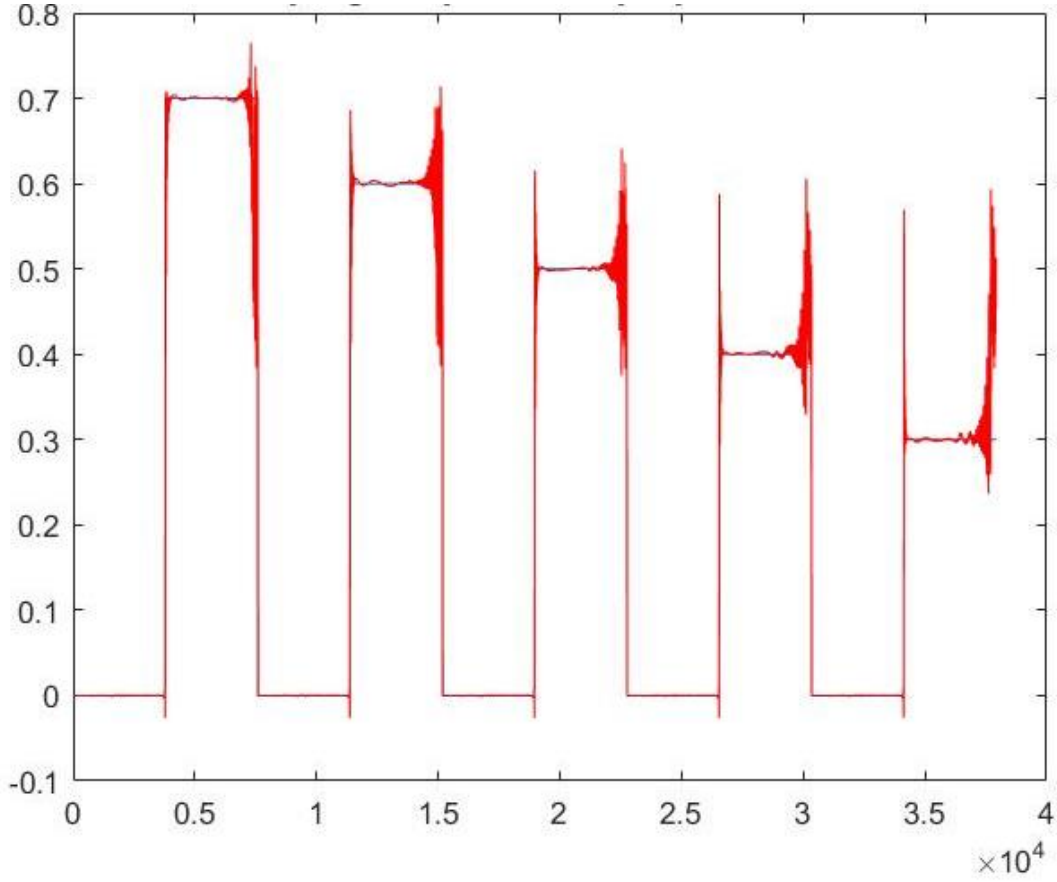
Şekil 3.12. NN'in 2. Eklem için Yakınsaması

Şekil 3.13.'te 3. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. 3. Eklem için; grafikteki çıkışlar hedef değerlerden önemli ölçüde farklılıklar göstermemektedir.



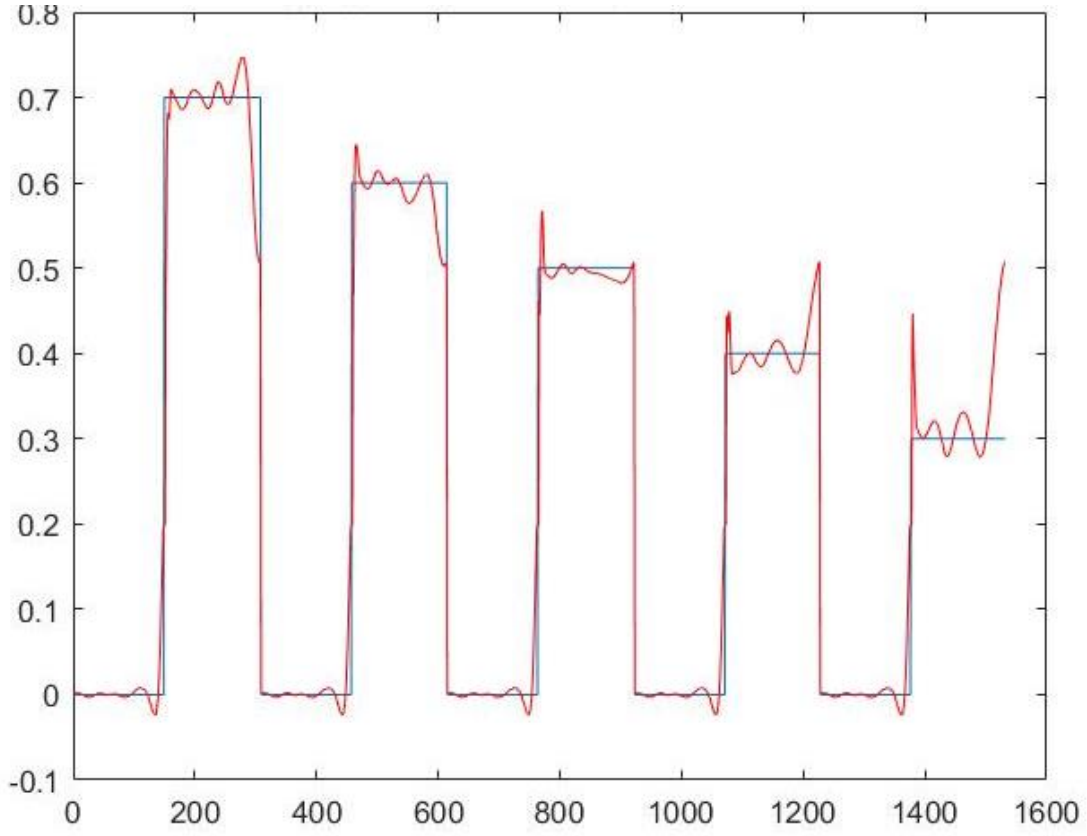
Şekil 3.13. NN'nin 3. Eklem için Yakınsaması

Şekil 3.14.'te 4. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Yakınsamada Keskin tepe noktaları ve düşüşler mevcuttur, modelin eğitim setindeki belirli giriş verilerine karşı duyarlı olduğunu ve bu noktalarda yüksek aktivasyonlar ürettiğini gösterir. Veri setindeki küçük varyasyonları aşırı öğrenmesine sebep olur ve bu şekilde başka bir sinir ağı ile düzelmesi beklenmektedir.



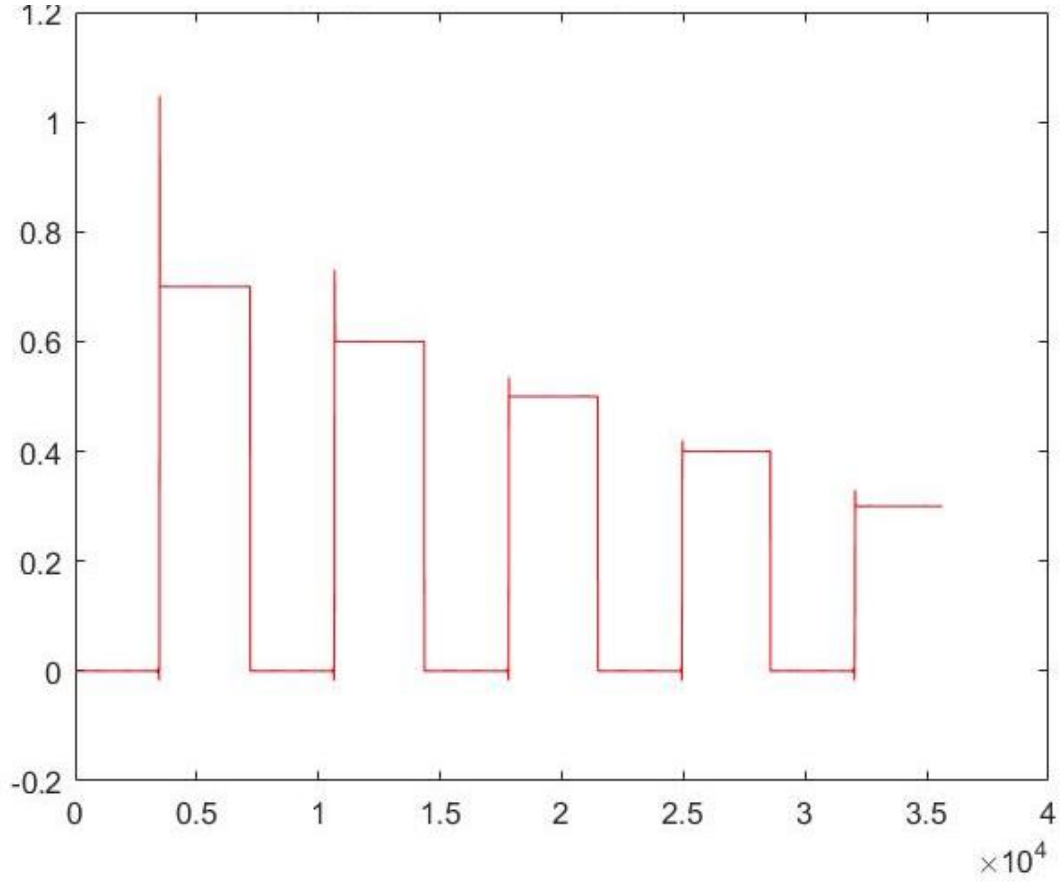
Şekil 3.14. NN'nin 4. Eklem için Yakınsaması

Şekil 3.15.'te 5. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Kırmızı eğrinin bazı noktalarda mavi eğriden uzaklaştığı görülmekte, bu da bazı durumlarda modelin hedef çıktılarından sapma gösterdiğini gösterir. Modelin hassaslaştırılması ve eğitim veri setinin genişletilmesi sapmaların engellenmesi için iyi bir yoldur. Çünkü, x ekseni 1478 adet veriye sahip olduğu için diğer eklemlere göre sapmanın fazla olması bu nedenden olduğu düşünülmektedir.



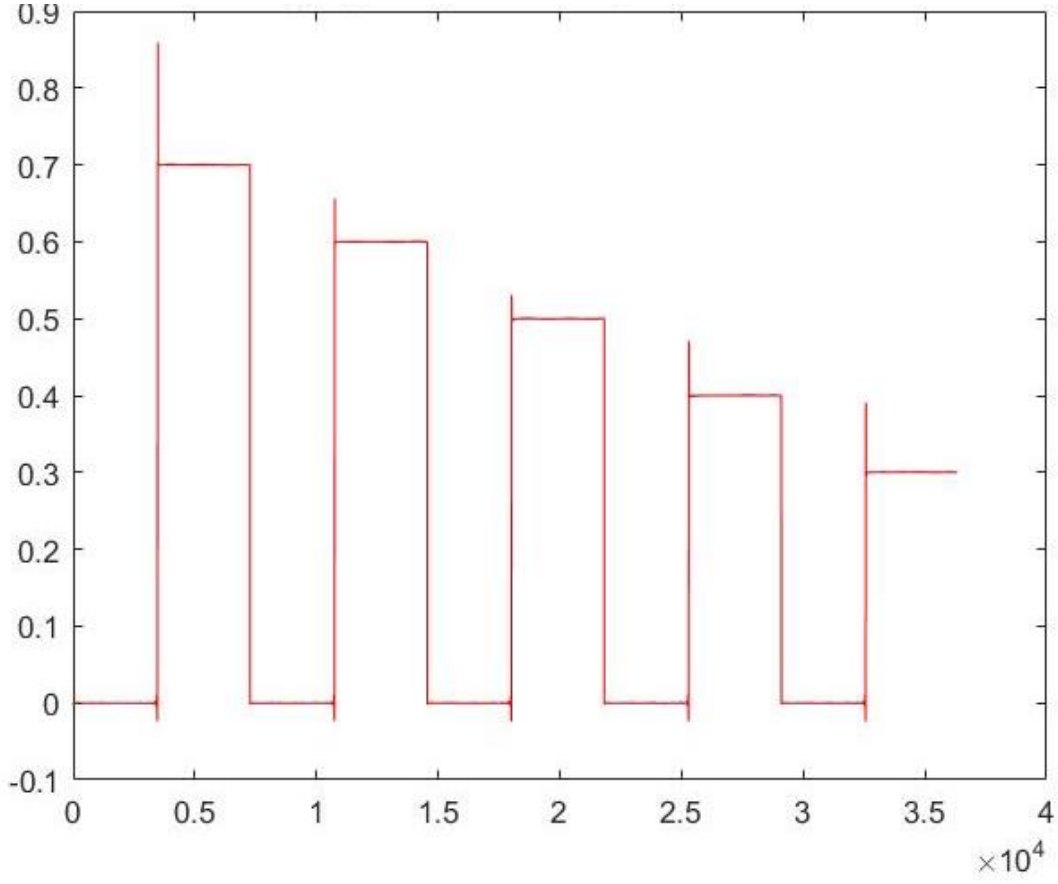
Şekil 3.15. NN'nin 5. Eklem için Yakınsaması

Şekil 3.16.'te 6. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. 6. Eklem için yakınsama istenilene uygun olarak gözlemlenmektedir.



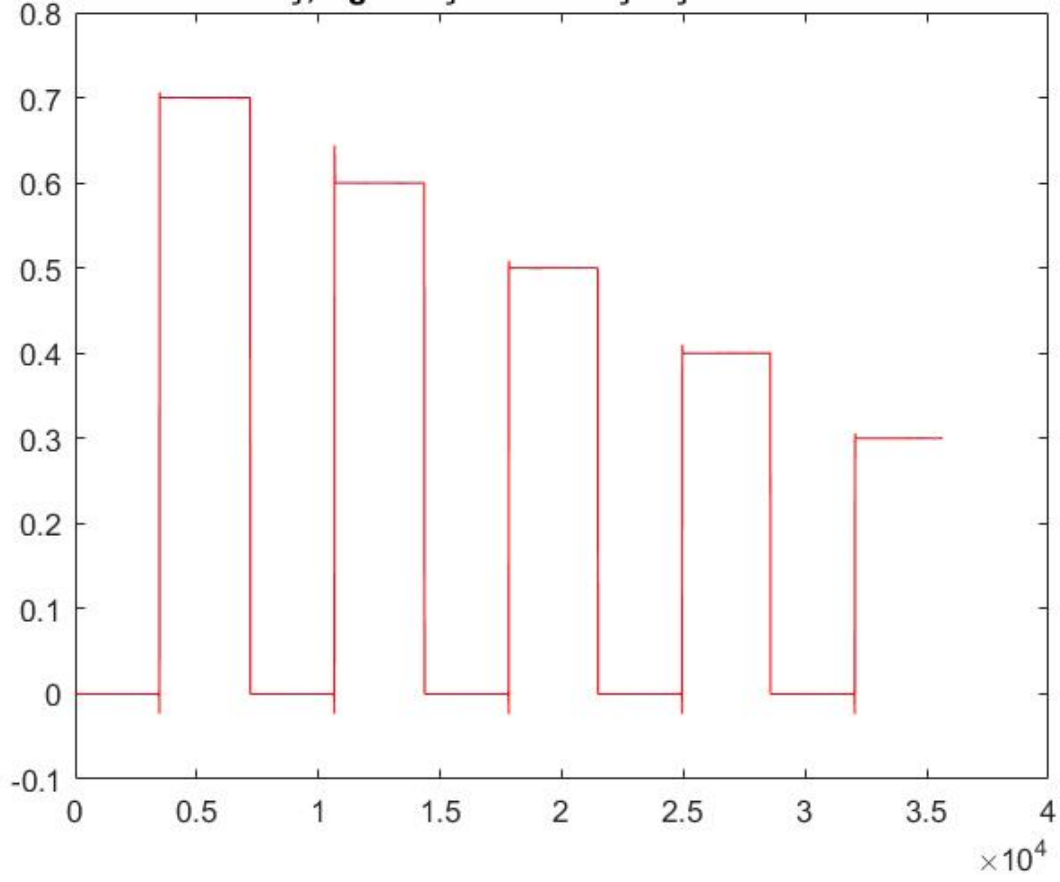
Şekil 3.16. NN'nin 6. Eklem için Yakınsaması

Şekil 3.17.'de 7. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. 7. Eklem için yakınsama istenilene uygun olarak gözlemlenmektedir.



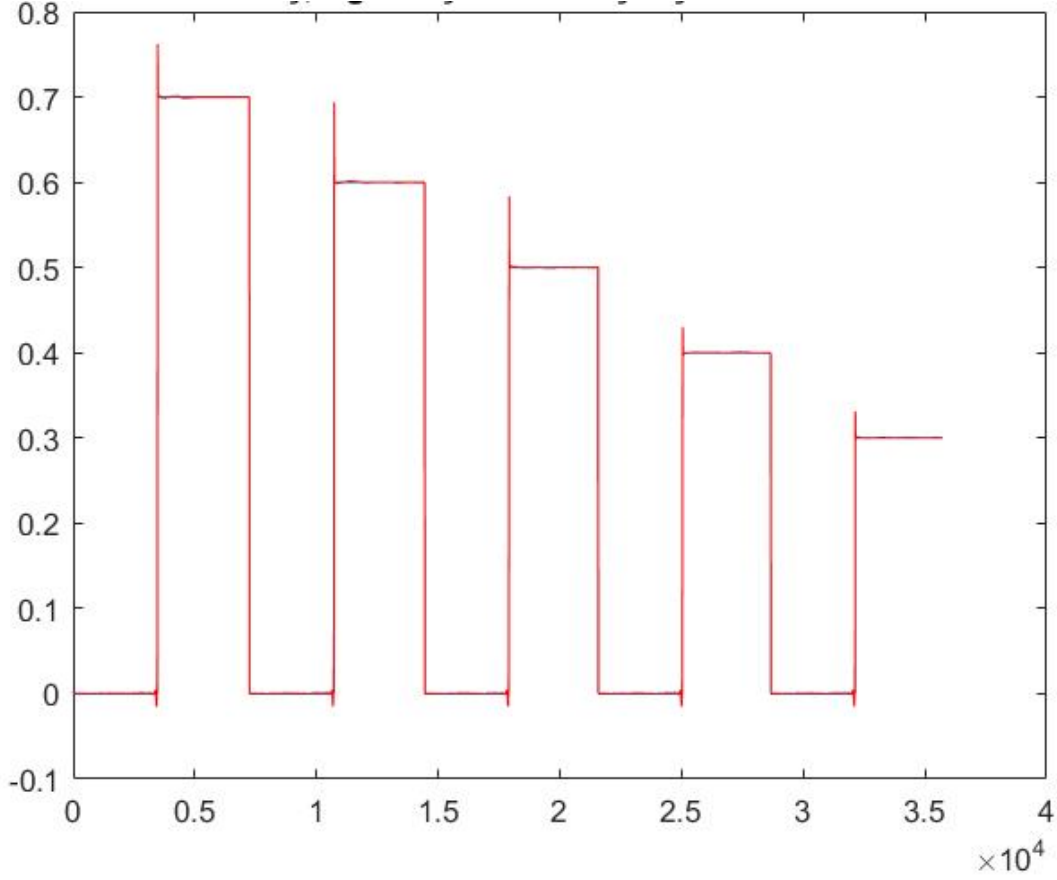
Şekil 3.17. NN'nin 7. Eklem için Yakınsaması

Şekil 3.18.'de 8. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. 8. Eklem için yakınsama istenilene uygun olarak gözlemlenmektedir.



Şekil 3.18. NN'nin 8. Eklem için Yakınsaması

Şekil 3.19.'de 9. Eklem için önerilen ATS duruş kestirimi için oluşturulan yapay sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. 9. Eklem için yakınsama istenilene uygun olarak gözlemlenmektedir.



Şekil 3.19. NN'nin 9. Eklem için Yakınsaması

3.6. Uzun-Kısa Süreli Bellek (LSTM)

NNT için alternatif olması adına Matlab'da LSTM algoritması ile her ekleme bir eğitim yapılmıştır. Model, dört giriş özelliğine ve tek bir çıkışa sahip olacak şekilde yapılandırılmıştır. Gizli katman olarak adlandırılan katmanda iki yüz nöron bulunmaktadır. Modelin yapısı, giriş verilerini kabul eden bir dizi katmandan oluşuyor: bir dizilim giriş katmanı, uzun kısa vadeli hafıza (LSTM) katmanı, tam bağlantılı bir katman ve son olarak bir regresyon katmanı. LSTM katmanının modeldeki rolü, veri noktaları arasındaki uzun süreli bağımlılıkları öğrenmek ve zaman serisi tahmininde bulunmaktadır. Veriler, normalleştirme işleminden geçirilmiş, yani her bir giriş ve çıkış verisi, veri setinin genel ortalamasından çıkarılarak standart sapmaya bölünmüştür. Bu işlem, modelin eğitimi sırasında ölçeklenmiş veriler üzerinde çalışmasını sağlamıştır.

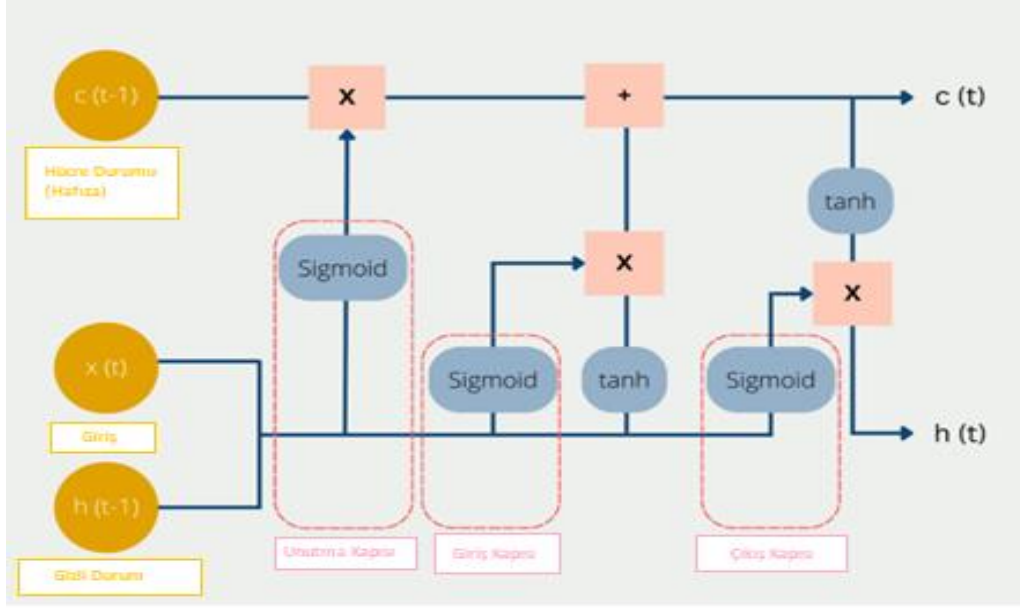
Önerilen sistemde, yer alan kodda LSTM katmanı, modelin mimarisinde belirli bir gizli nöron sayısı (numHiddenUnits) ile tanımlanmıştır. Bu katman, eğitim sırasında, her bir giriş

özelliğine (numFeatures) karşılık gelen zaman serisi verilerini işler ve bu verilerin içerdiği desenleri ve eğilimleri öğrenir. Eğitim sürecinde, LSTM katmanı giriş veri dizisini alır ve her bir zaman adımında bilgiyi işleyerek bir sonraki adıma aktarır. LSTM'in hücre yapısı, bilgi akışını düzenleyen ve gereksiz bilgileri filtreleyen kapılar (gate) içerir. Bu kapılar sayesinde, LSTM, öğrenilmesi gereken bilgiyi korurken, önemsiz bilgileri unuttur.

Kodda tanımlanan LSTM katmanı, eğitim veri setindeki her bir örneği sırayla işleyerek, her bir örneğin hedef çıktıyla olan ilişkisini modellemeyi amaçlar. Eğitim tamamlandığında, LSTM katmanı, öğrenilen desenleri kullanarak yeni veri örnekleri üzerinde tahminlerde bulunur.

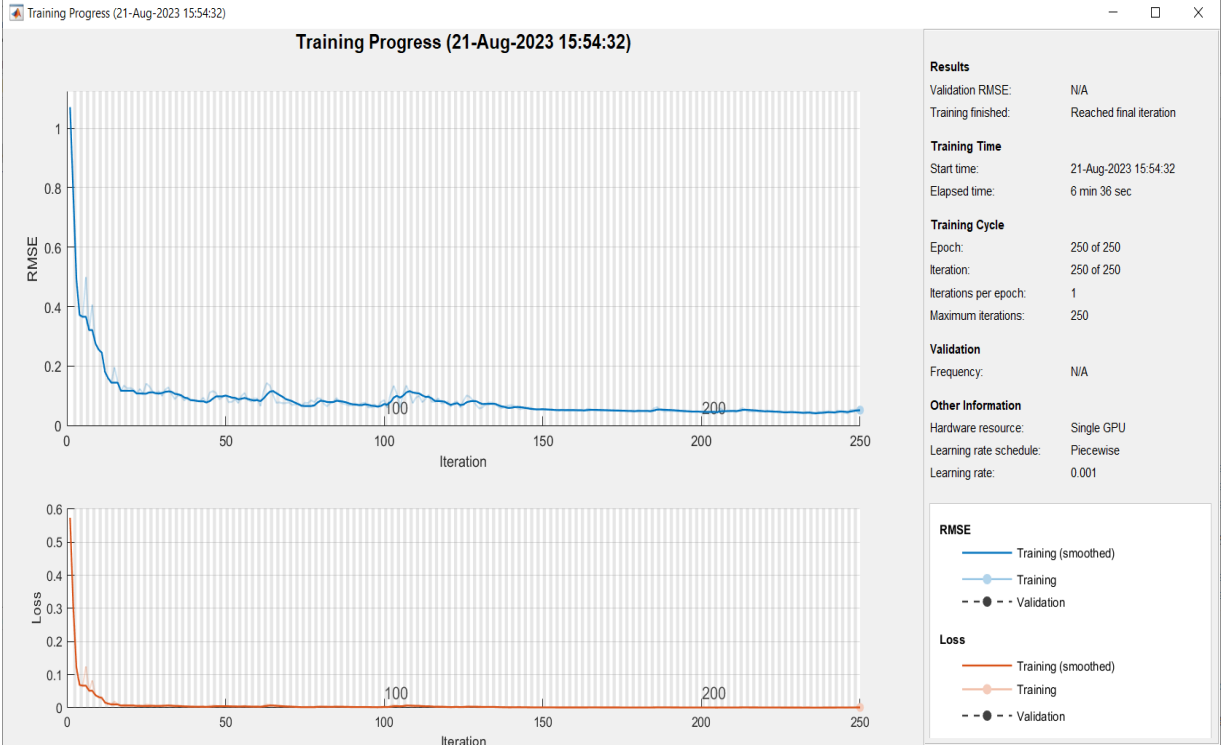
Özetle, LSTM katmanının kullanımı, önerilen sistemde yapay sinir ağı modeline, zamanla değişen veri akışlarını ve uzun süreli bağımlılıkları anlama yeteneği kazandırmıştır. Bu özellik, özellikle zaman serisi tahminleri, dizi sınıflandırma ve diğer benzer görevlerde önemlidir. Modelin LSTM katmanı sayesinde, karmaşık zaman serisi verileri üzerinde etkili bir şekilde eğitilmiş ve test edilmiş, böylelikle daha doğru ve güvenilir tahminler elde edilmesi sağlanmıştır. Şekil 3.20'de LSTM hücresinin yapısal şeması gösterilmiştir ve dört ana bölümden oluşmuştur. Unutma kapısı (forget gate), giriş kapısı (input gate), hücre durumu (cell state) ve çıkış kapısı (output gate). Her bir kapı ve hücre durumu, uygun ağırlıklar ve eğilimler (biases) ile parametreleştirilmiş fonksiyonlara sahiptir. Bu kapılar, hücre durumunun ve hücre çıktısının nasıl güncelleneceğini belirler. LSTM hücresinin bu dinamik yapısı, zamana bağlı verilerin karmaşık örüntülerinin etkin bir şekilde öğrenilmesine imkân tanır ve doğal dil işleme, konuşma tanıma ve zaman serisi tahmini gibi alanlarda uygulama bulur.

Bu kapılar, LSTM'nin hem kısa hem de uzun vadeli bilgileri etkili bir şekilde işlemesini sağlar, böylece karmaşık veri kümelerindeki bağımlılıkları öğrenir ve bu bilgileri gerektiğinde kullanır veya unuttur.



Şekil 3.20. LSTM Hücresinin Yapısal Şeması

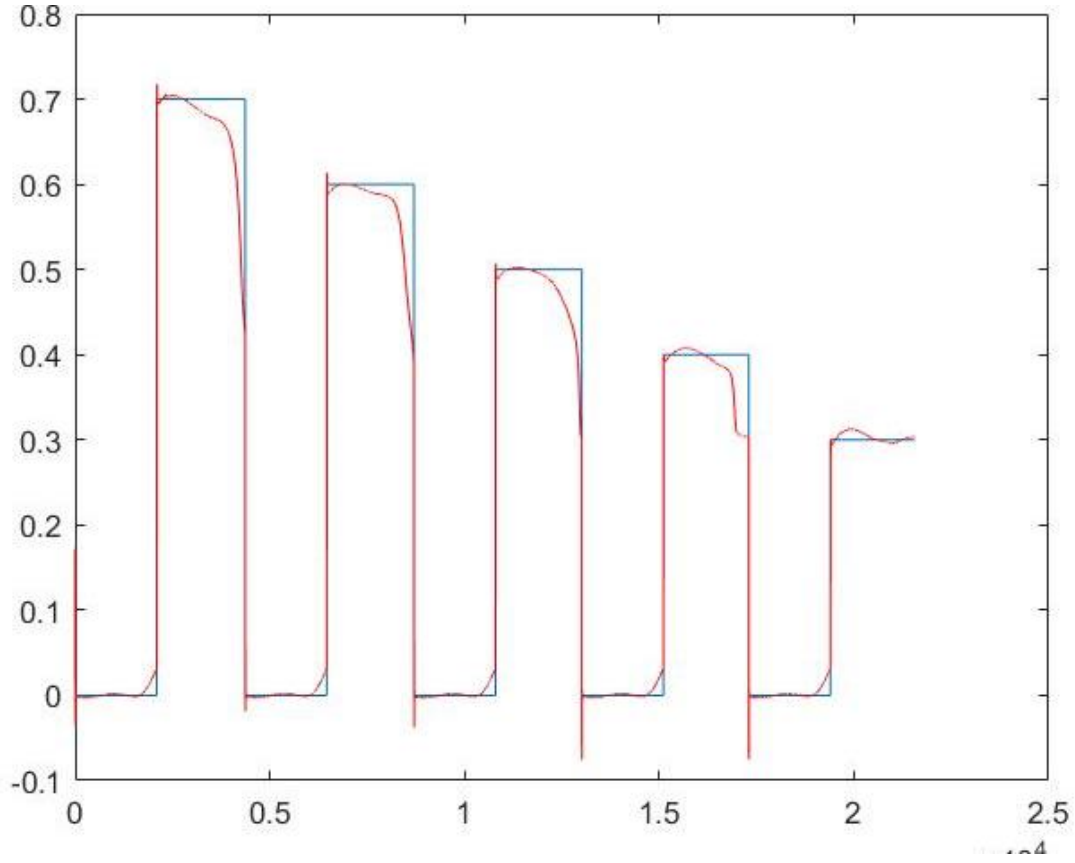
Şekil 3.21.'de Bölüm 3.4'te oluşturulan veri setlerinden yararlanarak duruş kestiriminde LSTM olarak kullanılan yapay sinir ağı yapısını 1. eklem için LSTM performansını görsel olarak sergilemektedir. Bu LSTM performansında; Grafikler, eğitim sırasındaki hata oranı (Root Mean Square Error - RMSE) ve kayıp (loss) değerlerinin iterasyonlar boyunca nasıl değiştiğini sergilemektedir. Mavi çizgi, eğitim seti üzerindeki T (RMSE) her iterasyonda nasıl düştüğünü göstermektedir. Düzgünleştirilmiş mavi çizgi, eğitim sürecindeki genel trendi temsil eder. Grafiğin başlangıcında keskin bir düşüş göstermesi, ağı hızlı bir öğrenme ilerlemesi kaydettiğini ve ilk iterasyonlarda hızlı bir performans iyileşmesi yaşandığını gösterir. Zamanla RMSE değerindeki düşüş hızı azalır ve bir düzleşme eğilimi gösterir, bu da modelin öğrenme sürecinde belirli bir olgunluğa ulaştığını ve ek iterasyonların hata oranını önemli ölçüde azaltmadığını gösterir. Kırmızı çizgiyle gösterilen eğitim kaybı (loss) değerinin iterasyonlar boyunca nasıl azaldığı görünür durumdadır. Düzgünleştirilmiş çizgi yine genel trendi gösterir. Eğitim kaybının düşük bir değere ulaşması, modelin eğitim verilerine iyi uyduğunu gösterir.



Şekil 3.21. 1. Eklem için LSTM Performansı

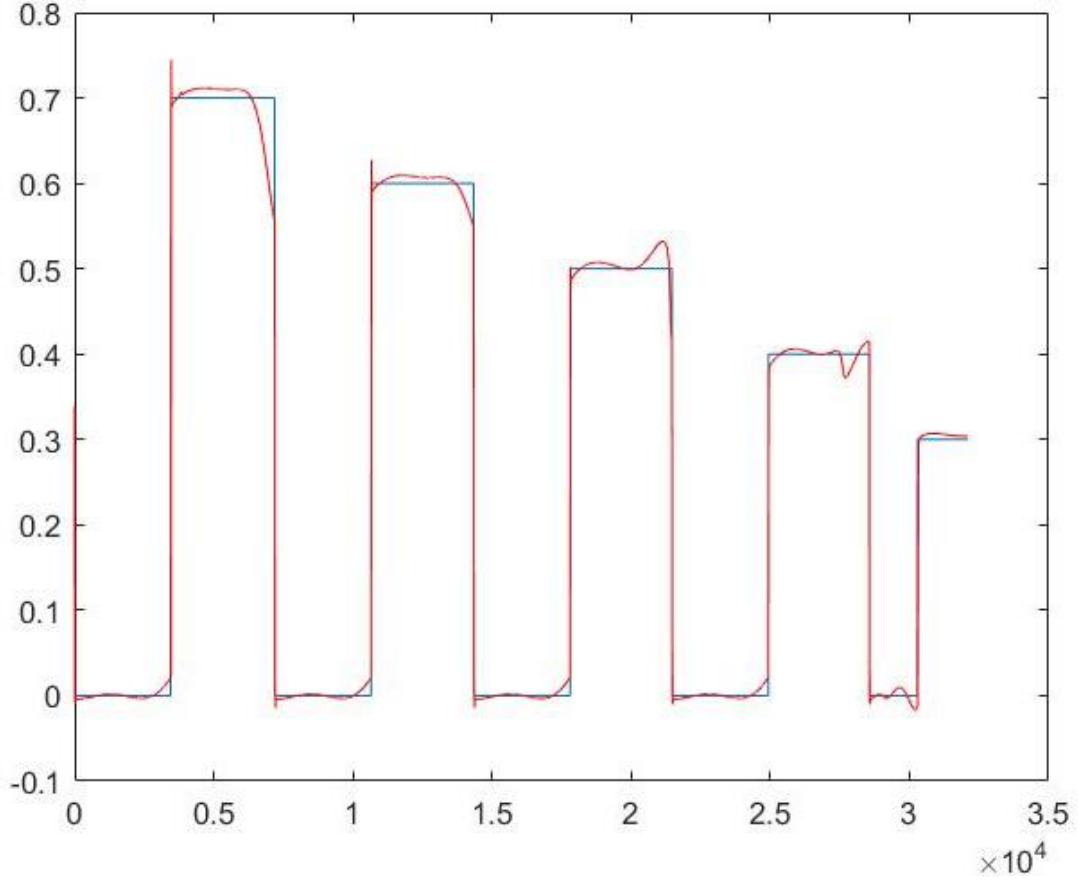
Genel olarak; LSTM modelinin eğitim süreci, RMSE ve kayıp değerlerinin iterasyonlar boyunca azalması ile karakterize edilmiştir. İlk başta hızlı bir iyileşme gösteren model, eğitim iterasyonlarının ilerlemesiyle daha stabil bir hata oranına ulaşmıştır. Elde edilen düşük RMSE ve kayıp değerleri, modelin eğitim veri setine iyi uyum sağladığını ve dolayısıyla eğitim sürecinin başarılı olduğunu göstermektedir. Eğitim detaylarının bu detaylı incelenmesi, LSTM modelinin beklentileri karşılayacak düzeyde olduğunu ve zaman serisi tahmin görevlerinde yüksek performans sergileyebileceğini işaret etmektedir.

Şekil 3.22.'de 1. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği Şekil 3.38'de verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Modelin eğitim süreci sırasında gösterdiği bu performans ve hedef değerlere olan yakınlığı, modelin gerçek dünya senaryolarına ve arıza tespiti gibi uygulamalara ne kadar uygun olduğunu göstermektedir.



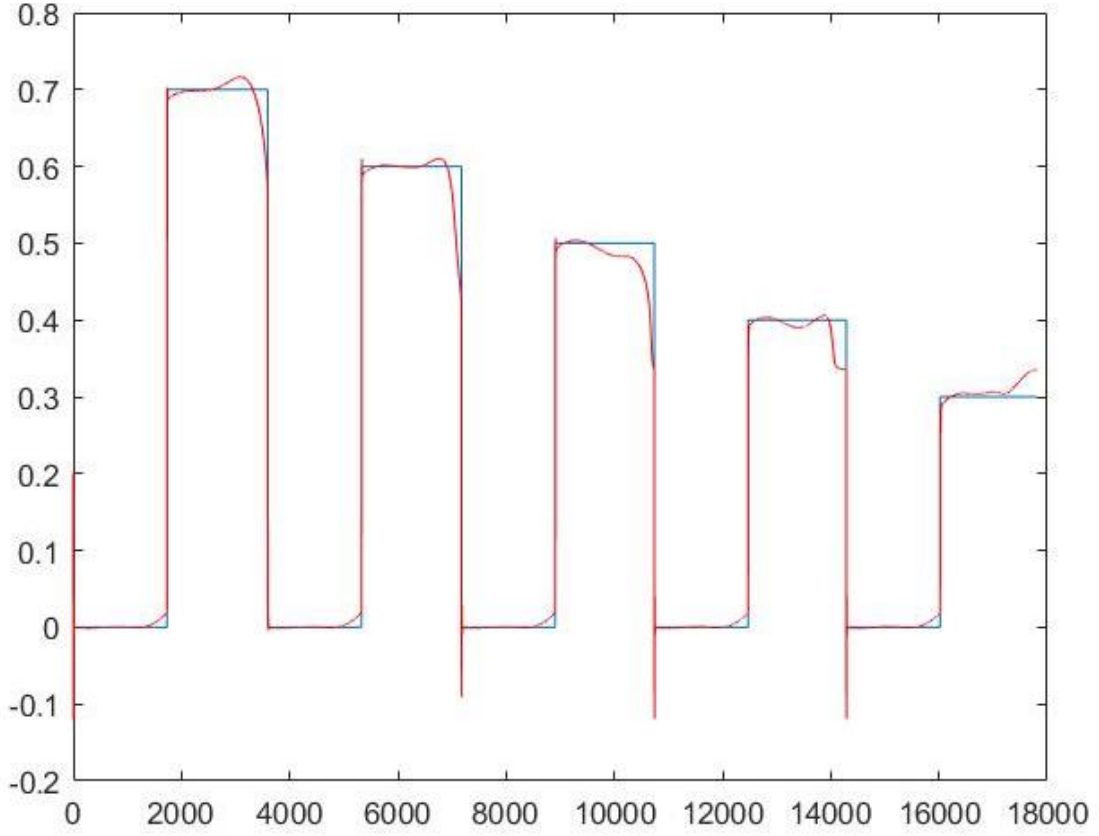
Şekil 3.22. LSTM'nin 1. Eklem için Yakınsaması

Şekil 3.23.'te 2. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Grafikteki mavi ve kırmızı çizgiler arasındaki uyum, genel olarak iyi bir yakınsama olduğunu göstermektedir. Çizgilerin neredeyse çakıştığı noktalar, modelin giriş verilerini doğru bir şekilde işlediğini ve tahminlerinin gerçek değerlere oldukça yakın olduğunu gösterir. Ancak, kırmızı çizginin bazı bölümlerde mavi çizgiden uzaklaştığı noktalar da mevcuttur, bu da modelin belirli durumlarda hedef değerlerden sapma gösterdiğini belirtir.



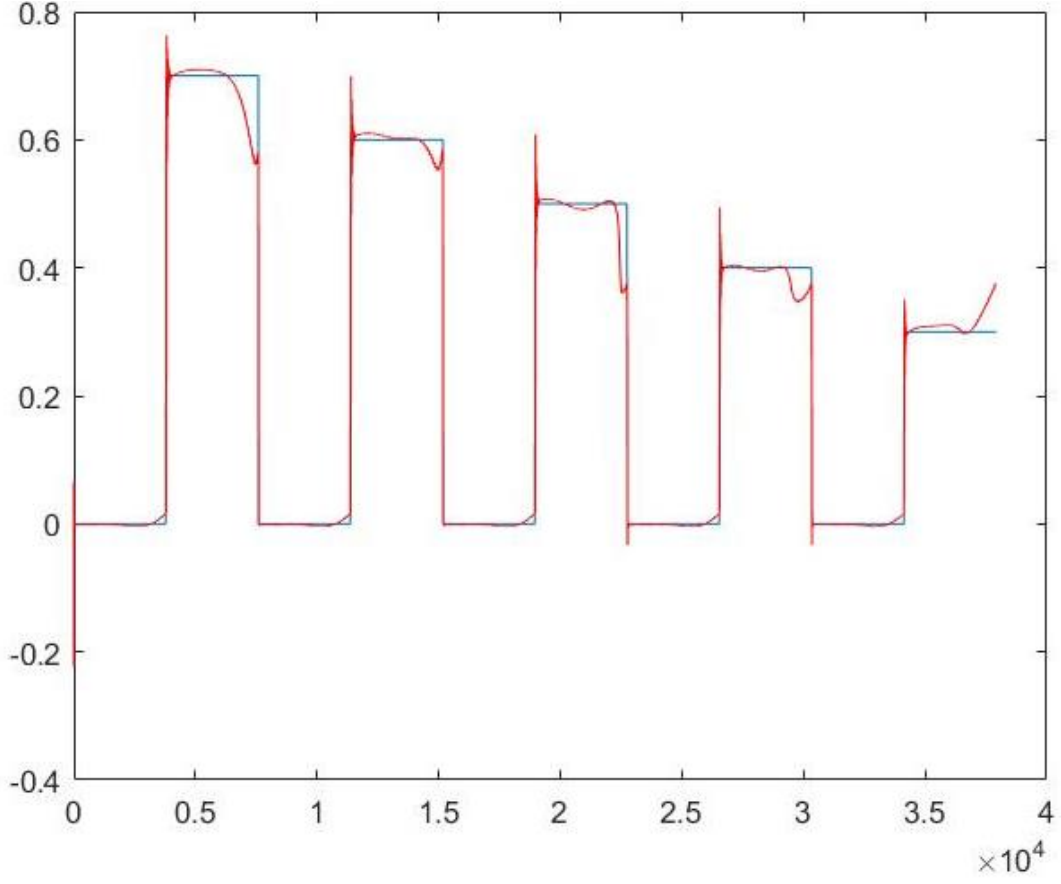
Şekil 3.23. LSTM'nin 2. Eklem için Yakınsaması

Şekil 3.24.'te 3. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği Şekil 3.39'da verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Modelin tahmin ettiği değerler, beklenen hedef değerlere genellikle yakın ancak modelin bazı durumlarında sapmalar göstermiştir.



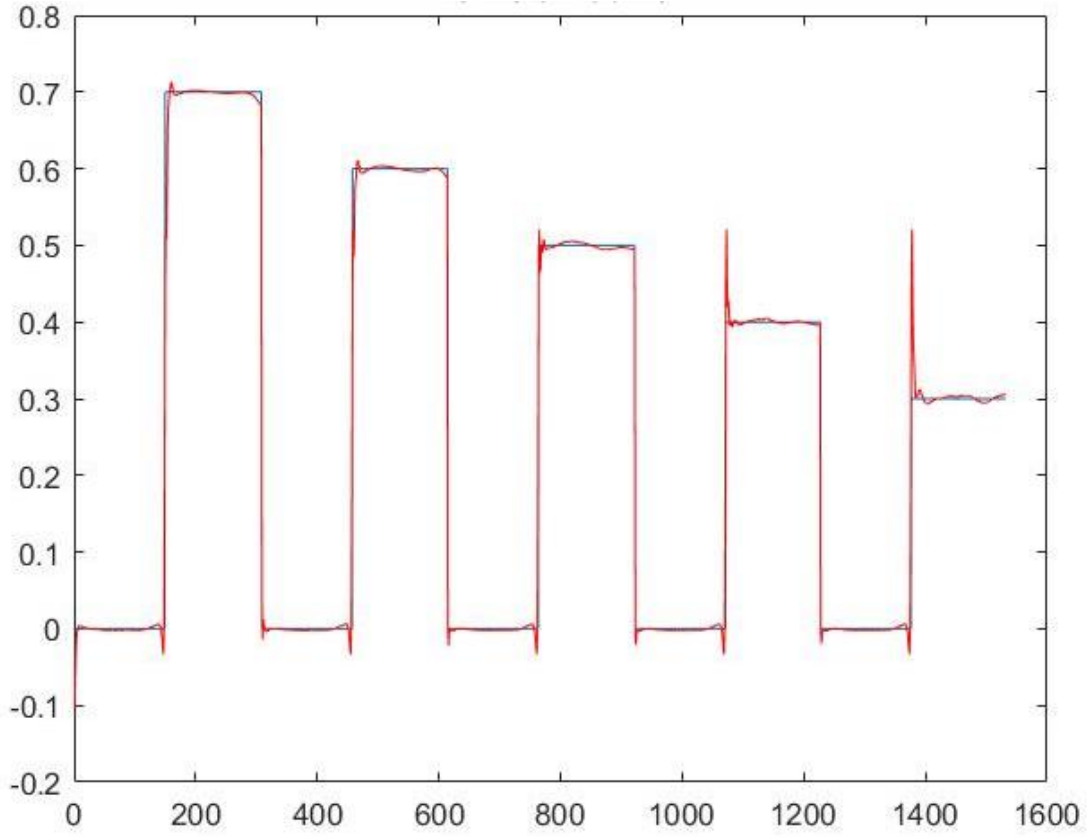
Şekil 3.24. LSTM'nin 3. Eklem için Yakınsaması

Şekil 3.25.'te 4. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Modelin eğitim süreci incelendiğinde, tahminlerin büyük bir kısmının hedef değerlere yakın olduğu görülse de, bazı noktalarda sapmalar mevcuttur. Bu sapmalar, modelin öğrenme kapasitesinin artırılması için parametre ayarlarının yeniden değerlendirilmesi veya eğitim veri setinin zenginleştirilmesi gibi iyileştirmelerin yapılması ihtiyacını ortaya koymaktadır. Ayrıca, modelin genelleştirme gücünü artırmak adına farklı mimari yapılarla denemeler yapılır.



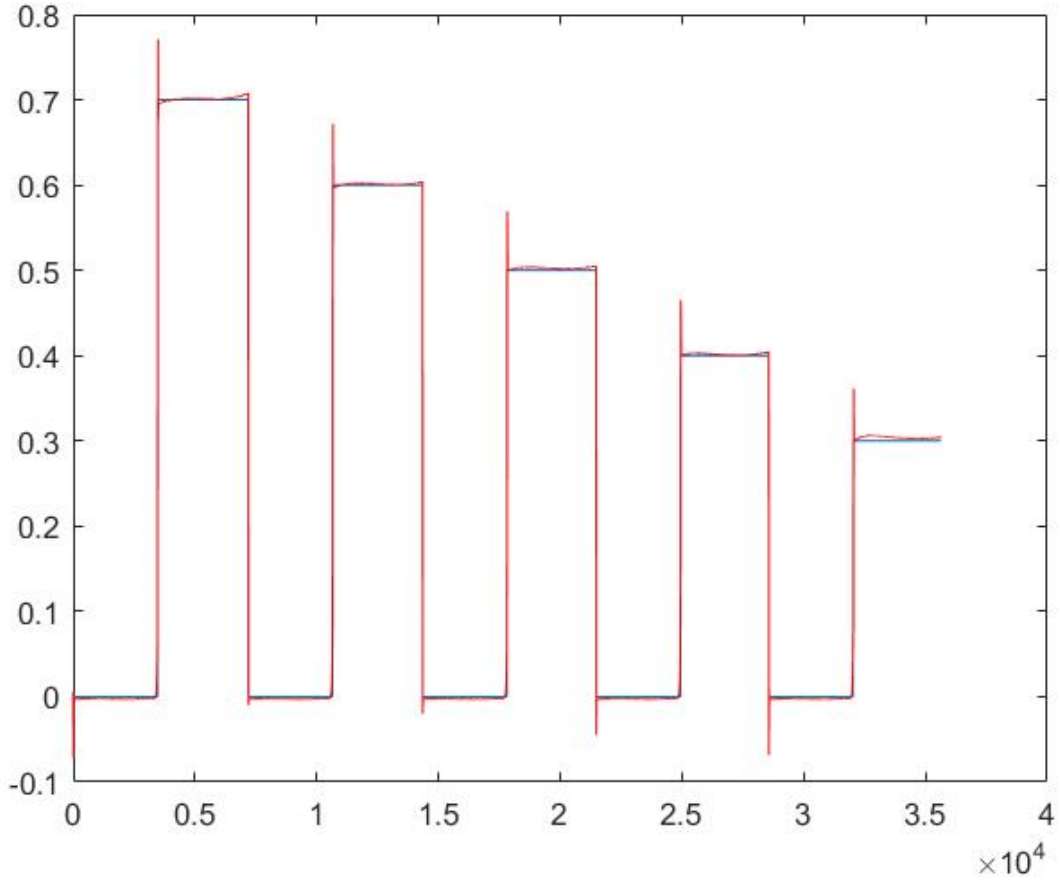
Şekil 3.25. LSTM'nin 4. Eklem için Yakınsaması

Şekil 3.26.'da 5. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Modelin tahminlerinin belirli noktalarda hedef değerlere oldukça yakın olduğu gözlemlenmiştir; ancak, hedef değerlerden hafif sapmaların olmuştur.



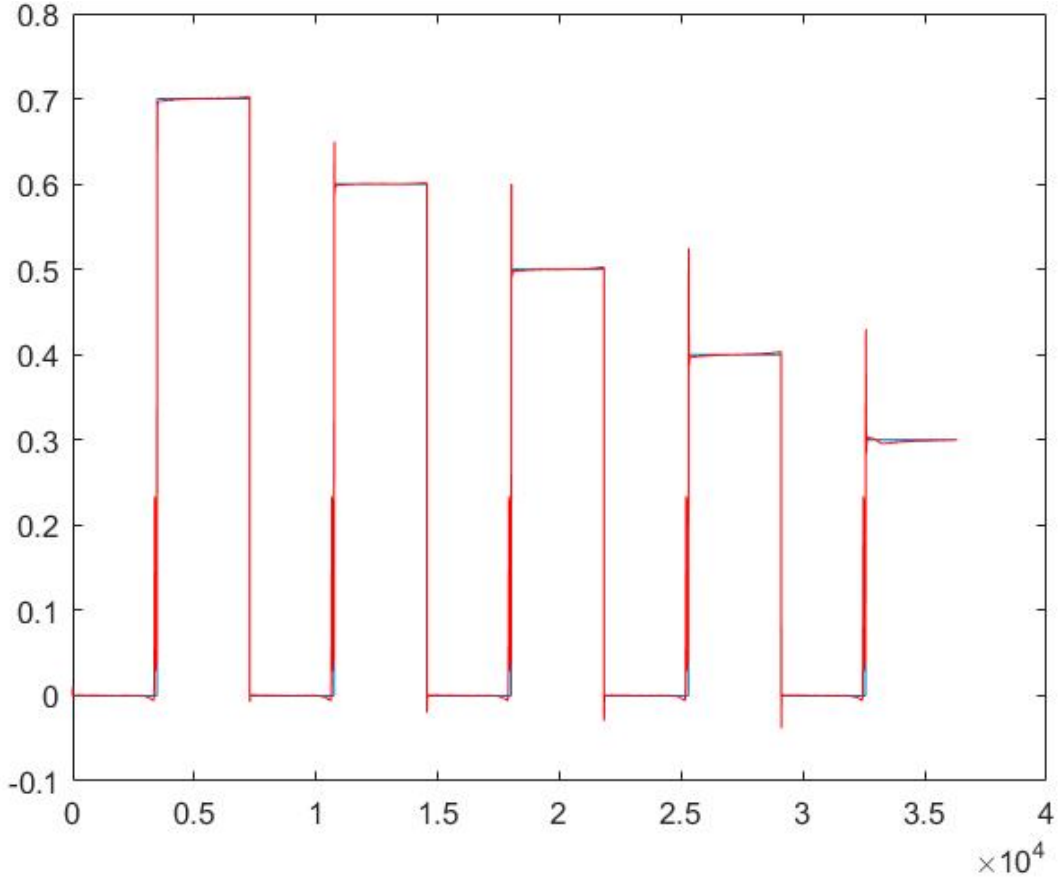
Şekil 3.26. LSTM'nin 5. Eklem için Yakınsaması

Şekil 3.27.'de 6. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Model güzel bir eğitim örneği sunmuştur.



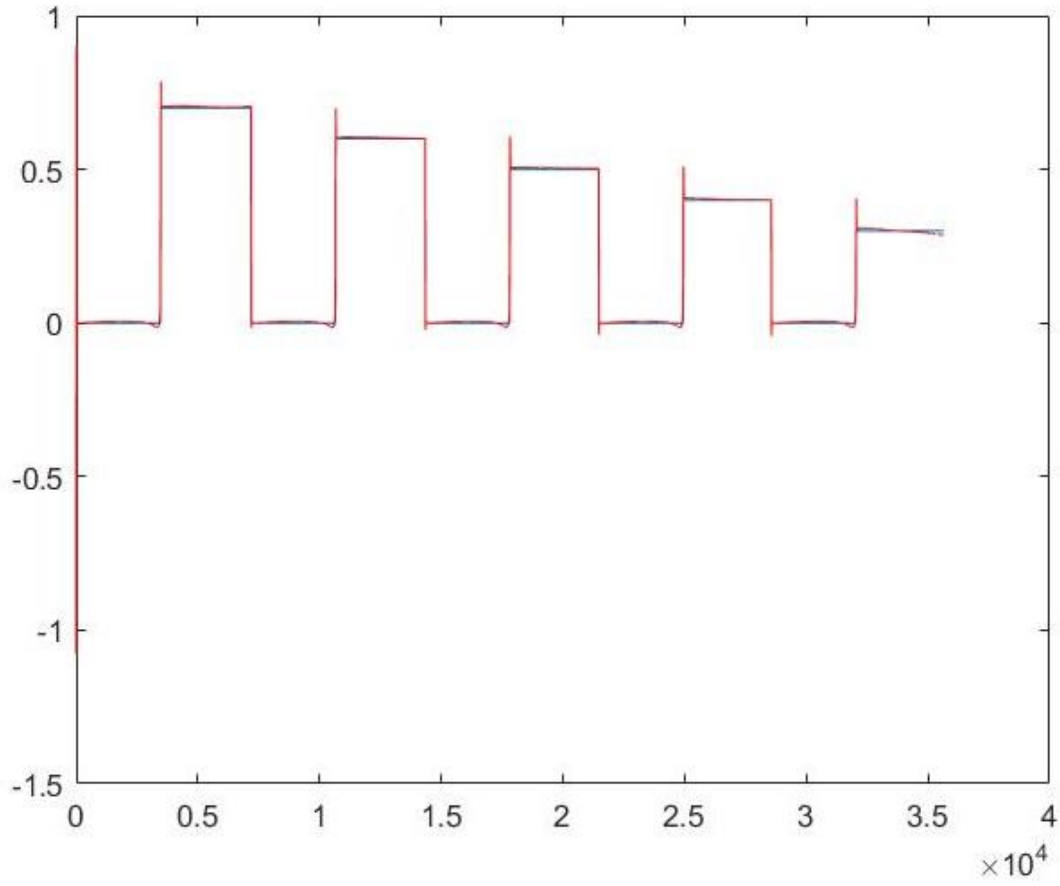
Şekil 3.27 LSTM'nin 6. Eklem için Yakınsaması

Şekil 3.28.'de 7. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Model güzel bir eğitim örneği sunmuştur.



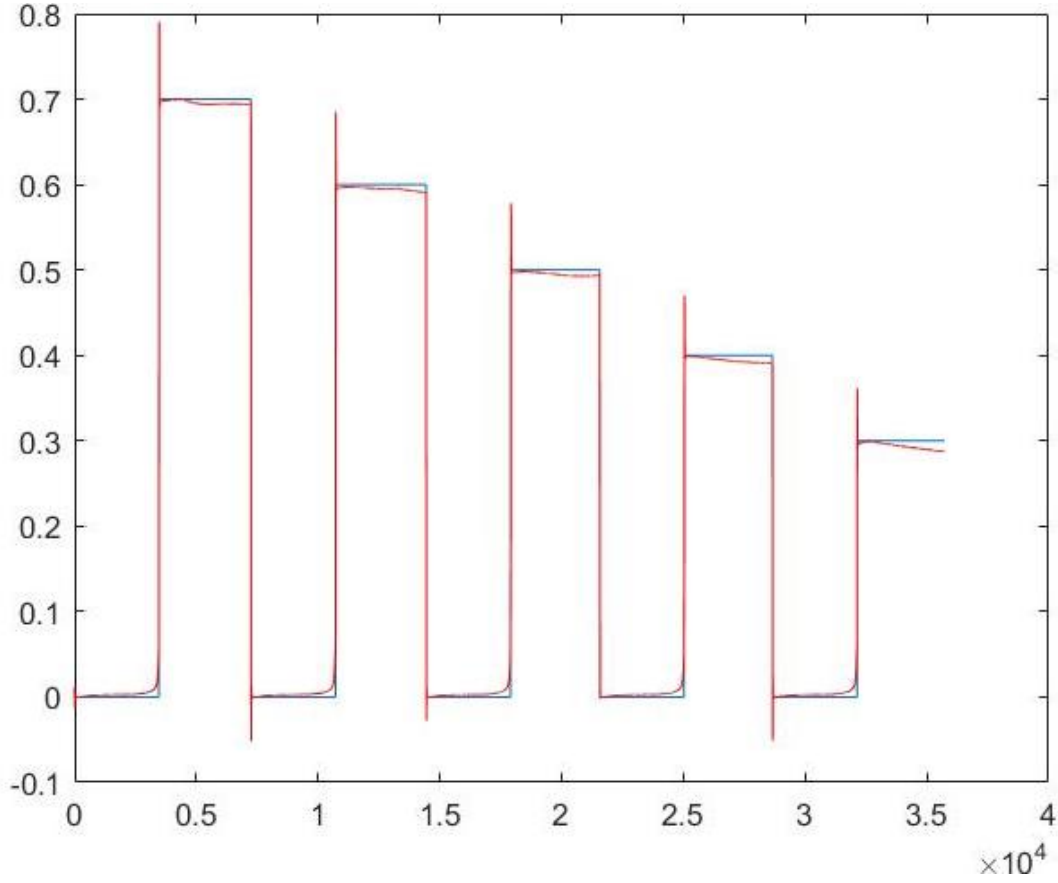
Şekil 3.28. LSTM'nin 7. Eklem için Yakınsaması

Şekil 3.29.'da 8. Eklem için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Model güzel bir eğitim örneği sunmuştur.



Şekil 3.29. LSTM'nin 8. Ekleme için Yakınsaması

Şekil 3.30.'da 9. Ekleme için önerilen ATS duruş kestirimi için oluşturulan LSTM sinir ağının, sahip olduğu giriş parametrelerine yakınsama grafiği verilmiştir. Grafikte mavi olarak gösterilen eğri olması gereken giriş değerlerini verir, kırmızı olarak gösterilen eğri ise yapay sinir ağının yakınsama eğrisini göstermektedir. Model çoğu noktalarda hedef değerlere yakındır ve bu da başarılı bir eğitim olduğunu göstermektedir.

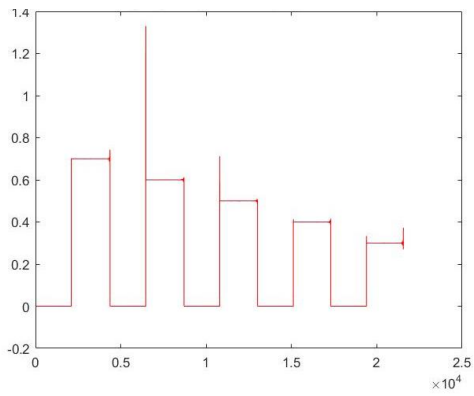


Şekil 3.30. LSTM'nin 9. Eklem için Yakınsaması

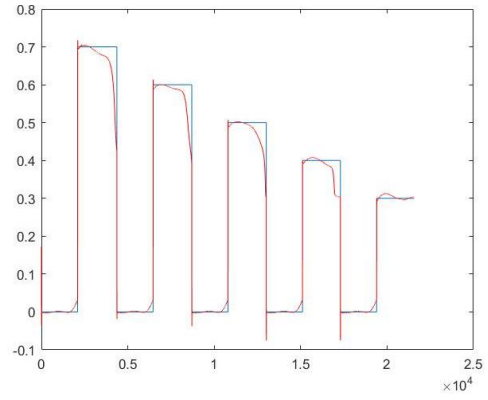
3.7. NN ve LSTM ile Yakınsama Karşılaştırılması

Bu çalışmada sunulan sistem, geleneksel arıza tespit yöntemlerinden farklı bir yaklaşım sergilemektedir. Sistem, başlangıç ve bitiş parametrelerine dayalı duruş tahminleri gerçekleştirdikten sonra, sistem bir seçim yapmakta. Bu seçimi de NN ve LSTM arasındaki eklem eğitimindeki başarıma göre yapmaktadır. Bu bölümde her konuma başlangıç noktasında başlanmasıyla 4 farklı konuma mobil manipülatör hareket etmiştir ve her eklem için bir NN ve LSTM karşılaştırılması yapılmıştır.

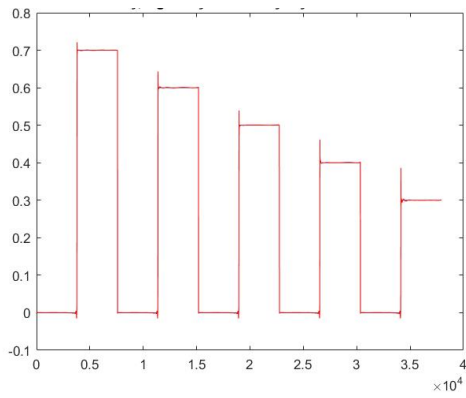
Şekil 3.31.'de 4 farklı durum için 1. Eklemin eğitilmiş verileri gösterilmiştir.



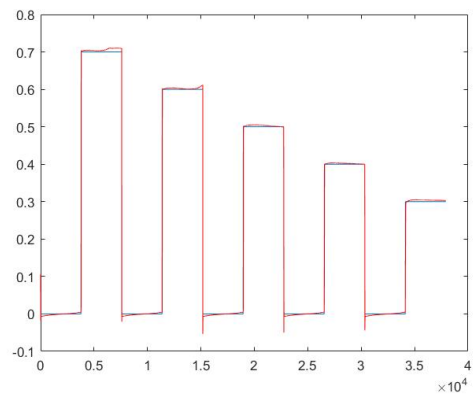
a)



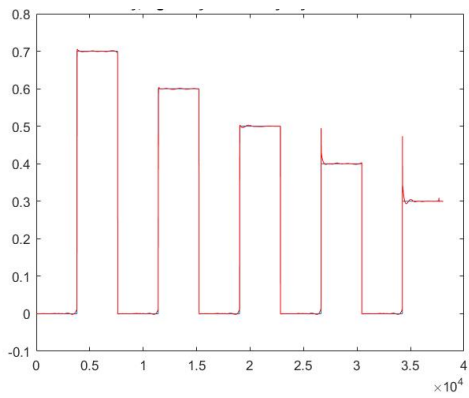
b)



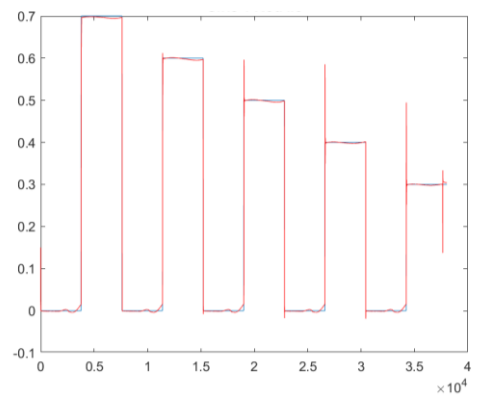
c)



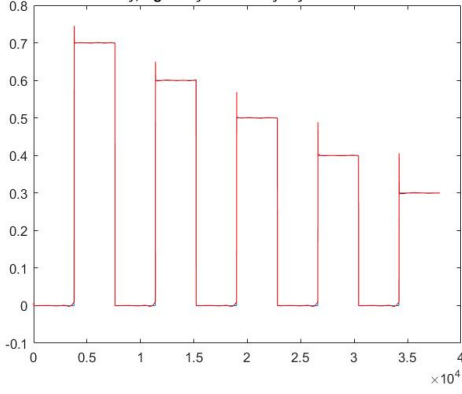
d)



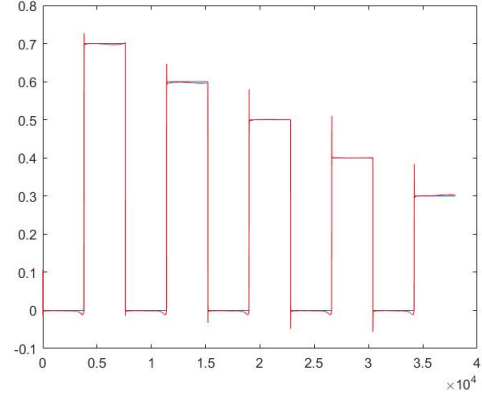
e)



f)



g)



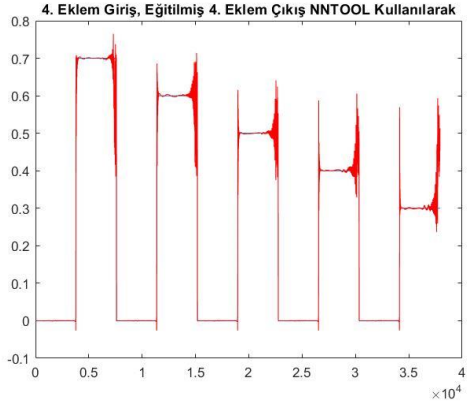
h)

Şekil 3.31. 1. Eklem için NN ve LSTM Karşılaştırması

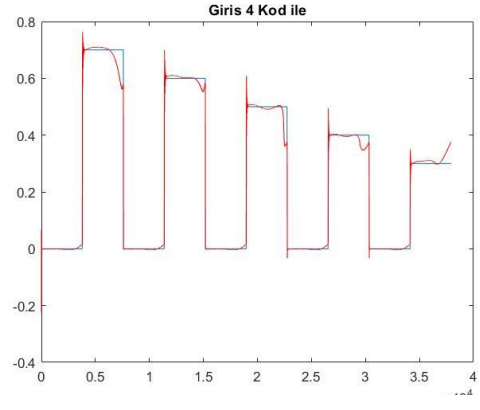
- a)** 1. durum 1. Eklem NN ile Eğitim **b)** 1. Durum 1. Eklem LSTM ile Eğitim
c) 2. Durum 1. Eklem NN ile Eğitim **d)** 2. Durum 1. Eklem LSTM ile Eğitim
e) 3. Durum 1. Eklem NN ile Eğitim **f)** 3. Durum 1. Eklem LSTM ile Eğitim
g) 4. Durum 1. Eklem NN ile Eğitim **h)** 4. Durum 1. Eklem LSTM ile Eğitim

Şekil 3.31'deki grafiklerdeki gözleme dayanarak 1. Eklem için, Karşılaştırmalar sonucunda, LSTM modelleri genel olarak daha düzgün ve tutarlı yakınsama eğrileri üretmiş ve 1. Eklem hareketinin tahmininde daha iyi performans sergilemişlerdir. Özellikle 3.31 a ve 3.31 f modelleri, ideal hareket profiline daha yakın sonuçlar vererek dikkat çekmiştir. Bu gözlemlere dayanarak, LSTM modellerinin 1. eklem hareketini öğrenme ve tahmin etme konusunda NN'den daha üstün olduğu sonucuna varılmıştır.

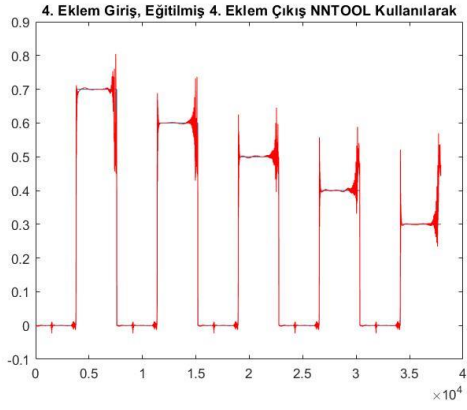
Dördüncü eklem önerilen sistemde yakınsamanın az olduğu eklemlerden biri, dördüncü eklem için de Şekil 3.32'de 4 farklı durum için NN ve LSTM karşılaştırılması eklenmiştir.



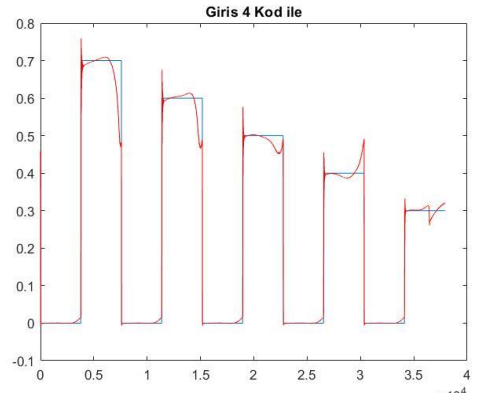
a)



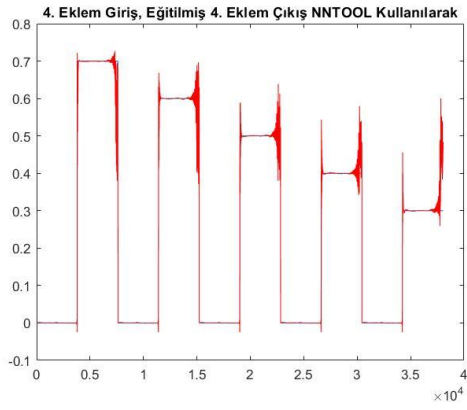
b)



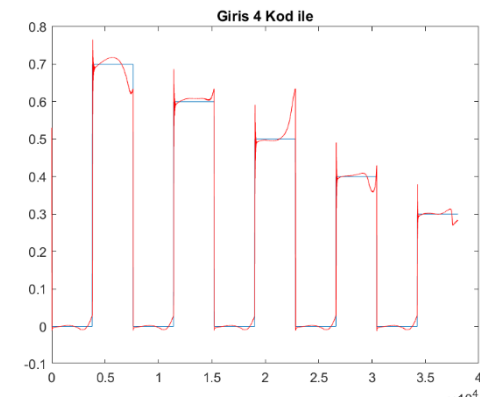
c)



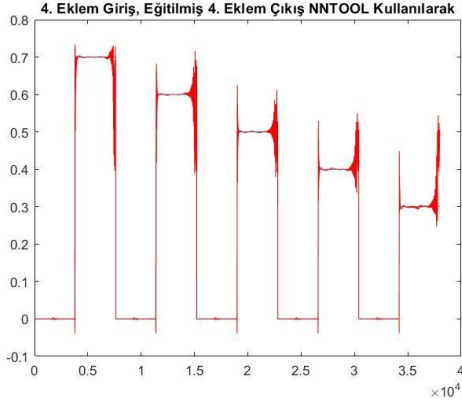
d)



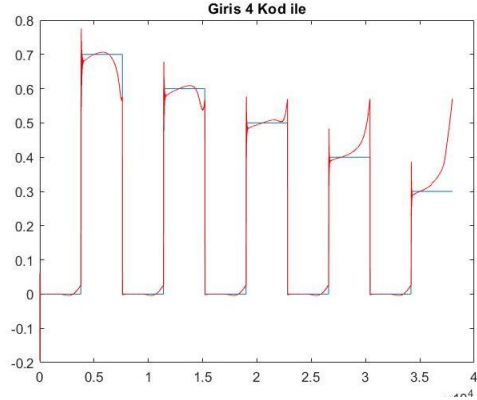
e)



f)



g)



h)

Şekil 3.32. 4. Eklem için NN ve LSTM Karşılaştırılması

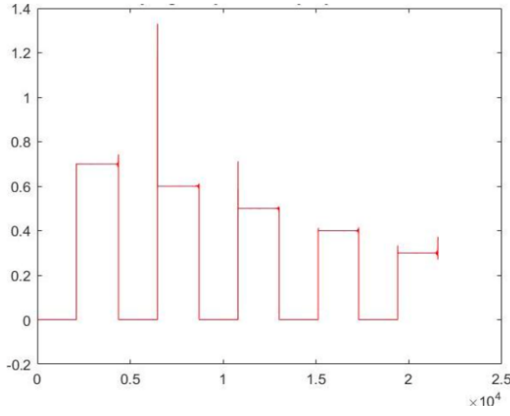
- a)** 1. durum 4. Eklem NN ile Eğitim **b)** 1. durum 4. Eklem LSTM ile Eğitim
c) 2. durum 4. Eklem NN ile Eğitim **d)** 2. durum 4. Eklem LSTM ile Eğitim
e) 3. durum 4. Eklem NN ile Eğitim **f)** 3. durum 4. Eklem LSTM ile Eğitim
g) 4. durum 4. Eklem NN ile Eğitim **h)** 4. durum 4. Eklem LSTM ile Eğitim

Şekil 3.32.'ye göre; dördüncü eklem hareketini öğrenmek için uygulanan iki farklı yapay zekâ modelinin, NN ve LSTM ağlarının performansları görsel olarak karşılaştırılmıştır. Görsel değerlendirmeye göre, LSTM modelleri genel olarak ideal giriş değerlerini daha tutarlı bir şekilde taklit etmiş ve daha düzgün bir yakınsama göstermiştir. Özellikle Şekil 3.32 b ve Şekil 3.32 f, mavi çizgiye yani beklenen eğriye daha yakın bir uyum sergileyerek, bu modellerin zaman serisi verilerini işlemede NN modellerine göre daha etkili olduğunu göstermiştir. Ancak, Şekil 3.32 d ve Şekil 3.32 h modellerinin NN karşılıklarıyla benzer performanslar sergilemesi, belirli durumlarda NN modellerinin de uygun olabileceğine işaret etmektedir. Bu görsel incelemeler sayesinde, 4. eklem için LSTM modellerinin zaman serisi tahminlerinde NN modellerine kıyasla potansiyel üstünlükleri ortaya konmuştur.

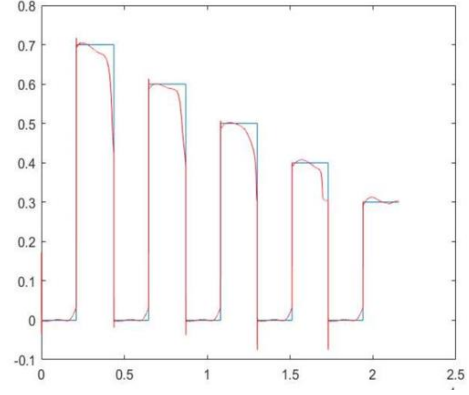
Tablo 3.2 NN MSE ve LSTM MSE değerlerini göstermektedir. Şekil 3.33 ise 1. Eklem için yakınsamaları göstermektedir:

Tablo 3.2. NNMSE ve LSTM MSE deęerleri

Eklem No:	NN MSE Deęeri	LSTM MSE Deęeri
1.Eklem	0.00060561	0.0000537
2.Eklem	0.00008044	0.0006476
3.Eklem	0.00091727	0.0000783
4.Eklem	0.000555	0.0002026
5.Eklem	0.000632	0.00034505
6.Eklem	0.00030063	0.00034505
7.Eklem	0.00006730	0.00012576
8.Eklem	0.000055	0.00011153
9.Eklem	0.0057	0.0023



a)



b)

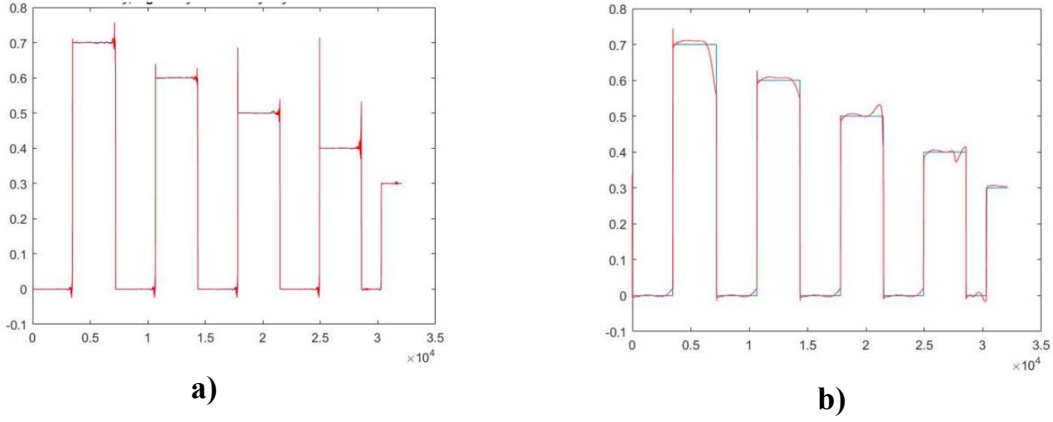
Şekil 3.33 1. Eklem için NN ve LSTM Yakınsaması

a) 1. Eklem NN ile Eğitim

b) 1. Eklem LSTM ile Eğitim

1. Eklem için Şekil 3.33 ve Tablo 3.2'deki MSE deęerleri ortak olarak deęerlendirildiğinde; Şekil 3.33 a için MSE deęeri 0.00060561 ve bu, modelin eğitim sırasında yüksek hata deęerlerine sahip olduğuna işaret ediyor. Kırmızı çizgilerin genlikleri büyük olduğundan ve bazı noktalarda beklenen mavi alandan oldukça uzakta olduğundan, bu modelin istenilen sonuçlardan sapmalar gösterdiğini ve optimizasyon için daha fazla işe ihtiyaç duyulduğunu gösteriyor. Şekil 3.33 b'de ise; kırmızı çizgilerin genliklerinin daha düşük olması ve daha dar bir aralıkta yer alması, modelin eğitim verilerine daha iyi uyduğunu ve daha tutarlı sonuçlar ürettiğini gösteriyor. İkinci grafik, modelin daha az hata ile tahminlerde bulunduğunu ve genel olarak daha istikrarlı bir öğrenme sürecine işaret ettiğini gösteriyor ve MSE deęeri de bunu gösteriyor.

Şekil 3.34’de 2. Eklem için NN ve LSTM yakınsamaları verilmiştir.



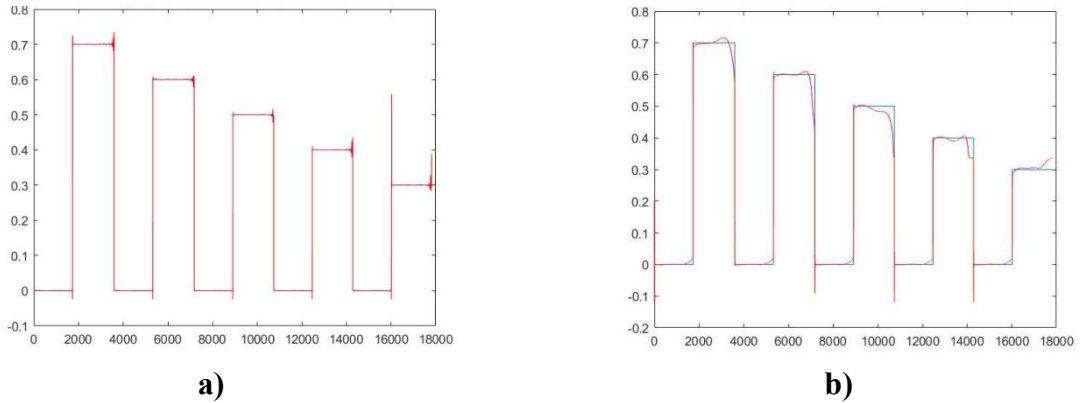
Şekil 3.34 2. Eklem için NN ve LSTM Yakınsaması

a)2. Eklem NN ile Eğitim

b) 2. Eklem LSTM ile Eğitim

Şekil 3.34 a’da MSE değeri 0.00008044 ve kırmızı çizgilerin genlikleri, Şekil 3.34 b’de MSE değeri 0.0006476’ya göre daha düşük ve dar bir aralıkta yer almaktadır. Bu, ilk modelin gerçek değerlere daha yakın tahminler yaptığını ve dolayısıyla daha düşük bir hata oranına sahip olduğunu gösterir. İkinci grafikte ise kırmızı çizgilerin genliklerindeki artış ve bazı noktalarda beklenen değerlerden büyük sapmalar göze çarpmaktadır. Bu durum, ikinci modelin verilerin altında yatan örüntüleri ilk modele kıyasla daha az başarıyla öğrendiğini ve daha yüksek hata değerleri ürettiğini gösterir.

Şekil 3.35’te 3. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



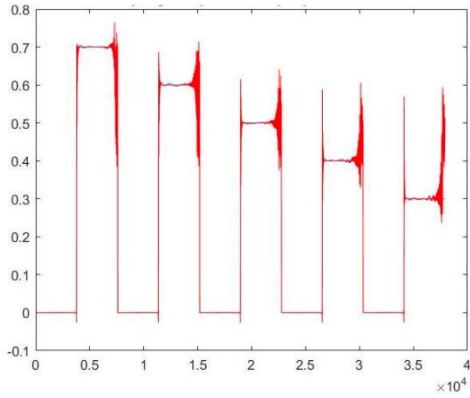
Şekil 3.35 3. Eklem için NN ve LSTM Yakınsaması

a) 3. Eklem NN ile Eğitim

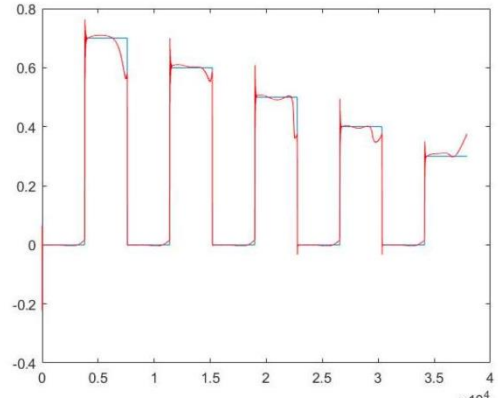
b) 3. Eklem LSTM ile Eğitim

Şekil 3.35'te a'da modelin MSE'si 0.00091727 iken, Şekil 3.35'te b'de modelin MSE'si daha düşük bir değer olan 0.0000783'tür. Yüksek MSE değeri, modelin eğitim veri setindeki varyasyonu yakalamakta zorlandığını gösterir ve NN ile eğitilen modelde oluşan sapmalar MSE değerini yükseltmiştir. LSTM ile eğitilen model ise, tahminlerinin gerçek değerlere daha yakın olduğunu ve dolayısıyla daha iyi bir performans sergilediğini gösterir. MSE değeri de bunu sağlamaktadır.

Şekil 3.36'te 4. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



a)



b)

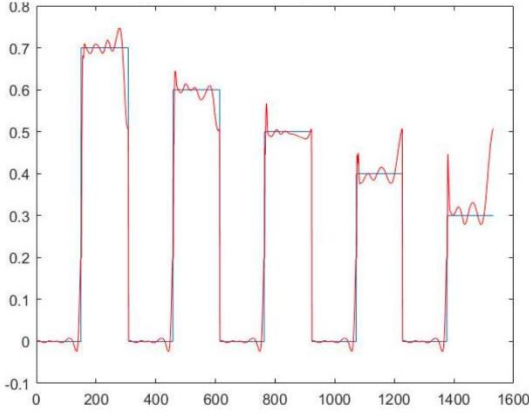
Şekil 3.36 4. Eklem için NN ve LSTM Yakınsaması

a) 4. Eklem NN ile Eğitim

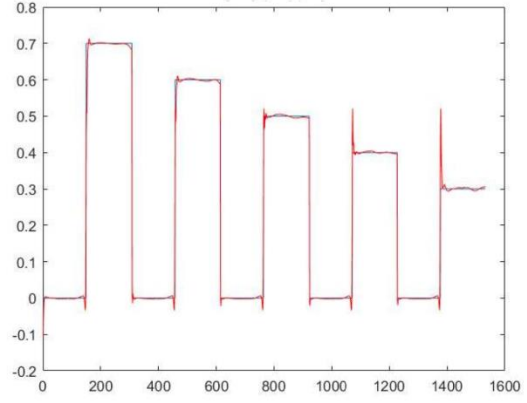
b) 4. Eklem LSTM ile Eğitim

Şekil 3.36'da a'da modelin MSE'si 0.000555 iken, Şekil 3.36'da b'de modelin MSE'si daha düşük bir değer olan 0.0002026'dır. Grafikte de görüldüğü gibi NN ile eğitimde daha fazla sapma vardır ve LSTM ile eğitim bu sapmaları azaltmıştır. LSTM daha iyi bir performans sergilemiştir.

Şekil 3.37'te 5. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



a)



b)

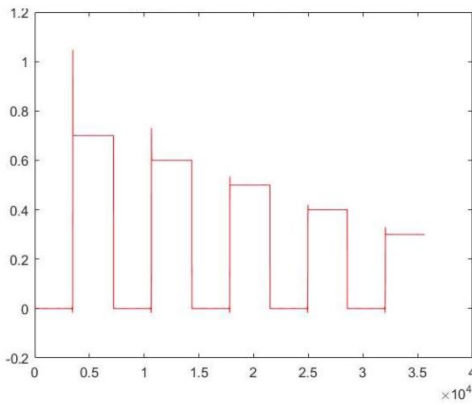
Şekil 3.37 5. Eklem için NN ve LSTM Yakınsaması

a) 5. Eklem NN ile Eğitim

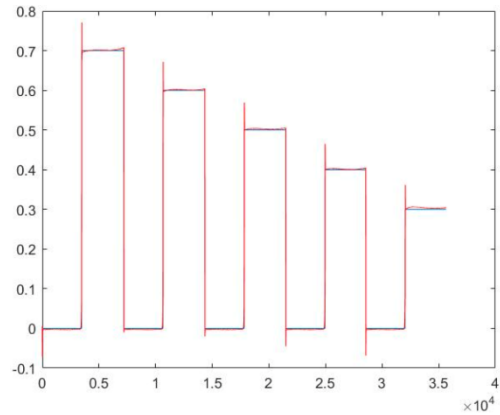
b) 5. Eklem LSTM ile Eğitim

LSTM ile eğitim sonrası elde edilen MSE değeri 0.00034505 iken, NN ile eğitim sonrası elde edilen MSE değeri 0.000632'dir. Karşılaştırma sonucunda, LSTM modelinin MSE değerinin NN modeline göre daha düşük olduğu görülmektedir. Bu, LSTM modelinin bu özel eğitim veri setinde daha iyi bir performans gösterdiği ve daha az hata ile tahminlerde bulunduğu anlamına gelir ve Şekil 3.37'de de LSTM ile eğitilen 5. Eklemdeki sapmaların NN ile eğitilene göre daha az olduğu belli olmaktadır.

Şekil 3.38'te 6. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



a)



b)

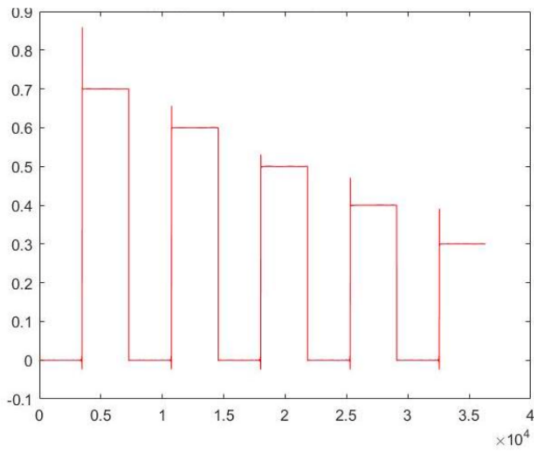
Şekil 3.38 6. Eklem için NN ve LSTM Yakınsaması

a) 6. Eklem NN ile Eğitim

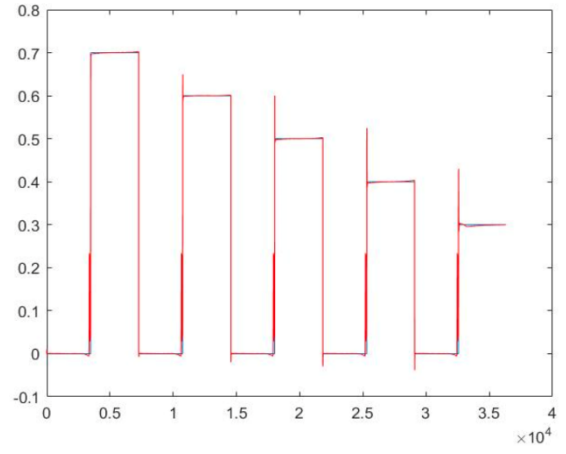
b) 6. Eklem LSTM ile Eğitim

NN ile eğitim sonrası elde edilen MSE değeri 0.00030063, LSTM ile eğitim sonrası elde edilen MSE değeri ise 0.00034505'tir. Yapılan karşılaştırmaya göre, NN modelinin MSE değeri LSTM modeline göre daha düşüktür. Bu, NN modelinin bu özel eğitim veri setinde daha iyi bir performans gösterdiği ve daha az hata ile tahminlerde bulunduğu anlamına gelir, ve grafiğe bakıldığında da sapmalar konusunda NN ile eğitilen veri setinin daha az olduğu gözlemlenmektedir.

Şekil 3.39'da 7. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



a)



b)

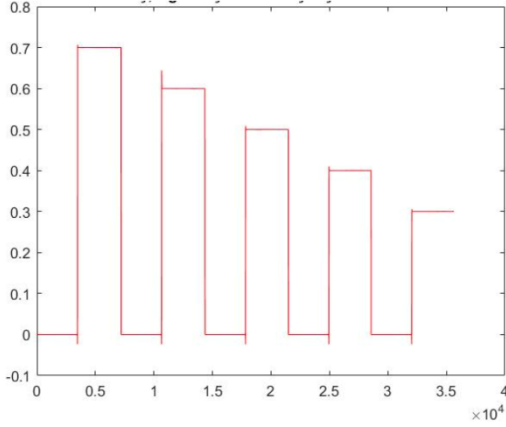
Şekil 3.39 7. Eklem için NN ve LSTM Yakınsaması

a) 7. Eklem NN ile Eğitim

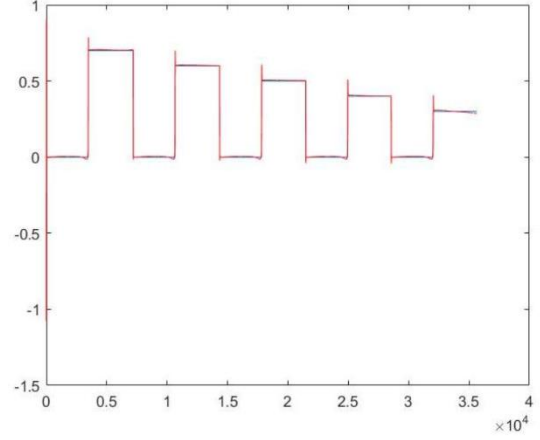
b) 7. Eklem LSTM ile Eğitim

NN ile eğitim sonrası elde edilen MSE değeri 0.0000673, LSTM ile eğitim sonrası elde edilen MSE değeri ise 0.00012576'dır. Karşılaştırmaya göre, NN modelinin MSE değeri LSTM modeline göre daha düşüktür. Bu, NN modelinin bu özel eğitim veri setinde daha az hata ile tahminlerde bulunduğunu ve dolayısıyla daha iyi bir performans gösterdiğini anlamına gelir, Şekil 3.39'da oluşan sapmalarda NN'de daha az olduğu görülmektedir.

Şekil 3.40'ta 8. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



a)



b)

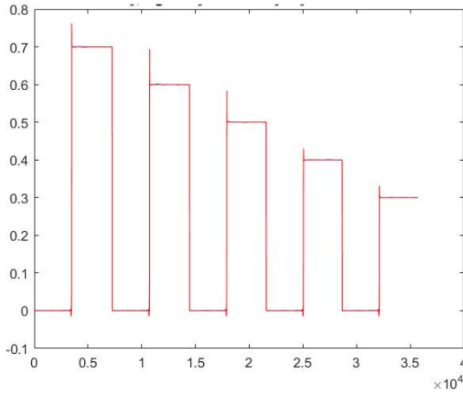
Şekil 3.40 8. Eklem için NN ve LSTM Yakınsaması

a) 8. Eklem NN ile Eğitim

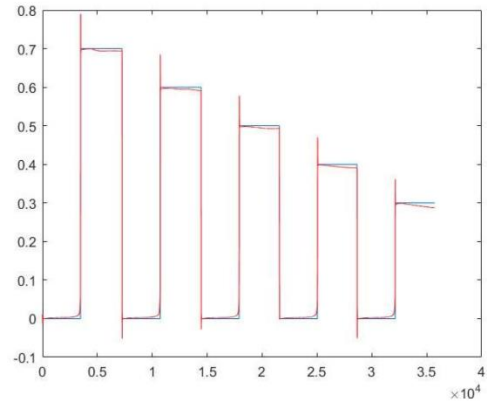
b) 8. Eklem LSTM ile Eğitim

Karşılaştırma sonucunda, NN ile eğitilen modelin MSE değerinin daha düşük olduğu görülmektedir. Bu, NN modelinin bu özel eğitim veri setinde daha iyi bir performans sergilediğini ve daha az hata ile tahminlerde bulunduğunu gösterir. Dolayısıyla, bu karşılaştırma, NN modelinin daha başarılı olduğunu göstermektedir. Grafiklerde de LSTM'de daha fazla sapma gözükürken NN'de bu sapmaların oranı daha azdır.

Şekil 3.41'de 9. Eklem için NN ve LSTM yakınsamaları gösterilmektedir;



a)



b)

Şekil 3.41 9. Eklem için NN ve LSTM Yakınsaması

a) 9. Eklem NN ile Eğitim

b) 9. Eklem LSTM ile Eğitim

Karşılaştırma sonucunda, NN (Yapay Sinir Ağı) ile eğitilen modelin MSE değerinin daha düşük olduğu gözlemlenmektedir. Bu, NN modelinin bu özel eğitim veri setinde daha iyi bir performans sergilediğini ve daha az hata ile tahminlerde bulunduğunu gösterir. Bu karşılaştırma, NN modelinin daha başarılı olduğunu göstermektedir. Grafikler arasında da LSTM’de sapmaların olduğu net şekilde görülmektedir.

4. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu çalışmada ele alınan mobil manipülatörlerde derin öğrenme tabanlı arıza tespiti, robotik sistemlerin giderek daha yaygın hale geldiği günümüz dünyasında büyük önem taşımaktadır. Mobil manipülatörler, otomasyon, üretim, sağlık hizmetleri ve uzay araştırmaları gibi çeşitli alanlarda kritik roller üstlenmektedir. Bu sistemlerin verimli ve güvenilir bir şekilde çalışması, operasyonel süreklilik ve güvenlik için hayati önem taşır. Arızaların hızlı ve doğru bir şekilde tespit edilmesi, bu kompleks makinelerin bakım ve onarım süreçlerini optimize ederken, aynı zamanda potansiyel tehlikeleri ve işletme maliyetlerini azaltır. Dolayısıyla, derin öğrenme tabanlı arıza tespit yöntemleri, robotik sistemlerin performansını artırma ve endüstriyel uygulamalarda daha güvenilir çözümler sunma potansiyeline sahiptir. Bu bağlamda, bu çalışmada sunulan bulgular, robotik teknolojilerin geleceğini şekillendirmede önemli bir adım teşkil etmektedir.

Bu araştırmada, mobil manipülatörlerdeki arızaların tespiti için derin öğrenme tabanlı bir yaklaşım geliştirilmiştir. Sunulan metodoloji, mobil manipülatörün hareket verilerini detaylı bir şekilde inceleyerek arızaları saptamaktadır. Önerilen sistemde klasik ATS'den farklı olarak duruş kestirimi için YSA mimarisi kullanılmıştır ve iki farklı sinir ağı kıyaslanmıştır. Arızaların tespit edilip düzeltilmesiyle ilgili bir çalışma yapılmasa da kontrol işlemi bu çalışmaya katkı sağlamaktadır.

İlgili çalışma, manipülatörün eklemlerinde oluşabilecek bozulmalar üzerinde yoğunlaşmış ve her 0.5 saniyelik zaman diliminde yüzde 30, 40, 50, 60 ve 70 olarak değişen oranlarda bozulma senaryolarını içermiştir. Dokuz farklı eklem üzerinden bu bozulma yüzdeleri ayrı ayrı incelenmiş, mobil robot manipülatörün bir bölgeden farklı bir bölgeye hareketini sağlayacak şekilde farklı eklem hareketleriyle de 5 deneme yapılarak her bir eklem için 5 farklı senaryo elde edilmiştir ve bu senaryolar için 5 farklı yüzde olarak değişen oranlar eklenmiştir toplamında 255 farklı senaryo incelenmiştir. Bu şekilde toplam elde edilen veriler NN ve LSTM algoritmaları ile işlenerek, bu yöntemlerin arıza tespitindeki etkinliği ölçülmüştür. Normal durumda çalışan veri setleri ile arızalı durumları kıyaslandığında, her iki derin öğrenme algoritmasının da eğitimden sonra belirlenen bir eşik değeri kullanarak arıza tespiti yaptığı tespit edilmiştir. Hem grafiklere hem de yakınsama hatalarına bakıldığında küçük hatalar dahilinde başarılı bir yakınsama olduğu görülmektedir.

Bu bulgular ışığında, mobil manipülatörlerde arıza tespiti için derin öğrenme yaklaşımlarının değerini göstermektedir. Gelecekteki çalışmalarda, daha geniş ve çeşitli veri

setlerinin, gelişmiş derin öğrenme modellerinin ve gerçek zamanlı uygulamaların kullanılması öngörülmektedir. Çoklu sensör entegrasyonu ve veri füzyonu ile daha kapsamlı ve doğru arıza tespitlerinin yapılması sağlanır. Ayrıca, enerji verimliliği ve kullanıcı dostu arayüzlerle bu teknolojilerin birleştirilmesi, geniş bir kullanıcı kitlesine hitap eden daha sürdürülebilir ve erişilebilir robotik çözümler sunulmasına katkıda bulunur.

KAYNAKÇA

- Akhtar, N.** (2012). *Improving reliability of mobile manipulators against unknown external faults*. Sankt Augustin,, Germany: Dean Prof. Dr. Wolfgang Heiden.
- Chiang, L., Russell, E., & Braatz, R.** (2000). *Fault Detection and Diagnosis in Industrial Systems*. USA: Springer Science & Business Media.
- Çınar, S., Fidan, M., Demirci, M., & Yabanova, İ.** (2020). LabVIEW ve Mikro-denetleyici Tabanlı Sıcaklık Kontrol Sistemi Tasarımı. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 3(10), 1628-1639.
- Ergezer, H., Dikmen, M., & Özdemir, E.** (2003). Yapay sinir ağları ve tanıma sistemleri. *PiVOLKA*, 6(2), 14-17.
- Hamner, B., Koterba, S., Shi, J., Simmons, R., & Singh, S.** (2010). An autonomous mobile manipulator for assembly tasks. *I(28)*, 131-149.
- Han, M.-W., & Kopacek, P.** (1997). Neural Networks and Fuzzy Robot Control. *IFAC Intelligent Manufacturing Systems*, 14(30), 291-296.
- Hernandez-Barragan, J., Lopez-Franco, C., Arana-Daniel, N., & Alanis, A.** (2021). Inverse kinematics for cooperative mobile manipulators based on self-adaptive differential evolution. *PeerJ Computer Science* 7(7), e419.
- Javid, K.** (2016). *Interfacing simulink/MATLAB with V-rep for analysis and control synthesis of a quadrotor*. Ankara: Middle East Technical University.
- Kang, Y., Li, Z., Dong, Y., & Xi, H.** (2012). Markovian-Based Fault-Tolerant Control for Wheeled Mobile Manipulators . *IEEE Transactions on Control Systems Technology*, 20(1), 266-376.
- Khan, H., Khatoon, S., Gaur, P., Abbas, M., Saleel, C., & Khan, S.** (2023). Speed Control of Wheeled Mobile Robot by Nature-Inspired Social Spider Algorithm-Based PID Controller. *Processes* 2023, 4(11), 1202.
- Lee, S.-D., & Song, J.-B.** (2016, JANUARY). Sensorless Collision Detection based on Friction Model for a Robot Manipulator. *International Journal Of Precision Engineering And Manufacturing*, 17(1), 11-17.

McIntyre, M., Dixon, W., Dawson, D., & Walked, I. (2004). Fault Detection and Identification for Robot Manipulators. *Proceedings of the 2004 IEEE International Conference on Robotics & Automation* (s. 4981-4986).

Pan, J., Qu, L., & Peng, K. (2022). Research on robotic manipulator fault detection and diagnosis technology based on machine vision in complex environments. *Journal of Field Robotics*, 2(40), 231-242.

Park, D., Kim, S., An, Y., & Jung, J.-Y. (2018). A Light-Weight Real-Time Fault Detection System for Edge Computing Using LSTM Recurrent Neural Networks. *Sensors*, 18(7), 2110.

Raj, R., & Kos, A. (2022). A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives. *Applied Sciences*, 12(14), 1.

Rodriguez-Tirado, A., Magallan-Ramirez, D., Martinez-Aguilar, J., Moreno-Garcia, C., Balderas, D., & Lopez-Caudana, E. (2020). A pipeline framework for robot maze navigation using computer vision, path planning and communication protocol. *2020 13th International Conference on Developments in eSystems Engineering (DeSE)* (s. 152-157).

Skoundrianos, E., & Tzafestas, S. (2004). Fault Diagnosis on the Wheels of a Mobile Robot Using Local Model Neural Networks. *IEEE Robotics & Automation Magazine*, 3(11), 83-90.

Song, T., Xi, F., Guo, S., & Lin, Y. (2016). Optimization of a Mobile Platform for a Wheeled Manipulator. *ASME. J. Mechanisms Robotics*, 6(8), 061007.

Tchon, K., & Jakubiak, J. (2005). A Hyperbolic, Extended Jacobian Inverse Kinematics Algorithm For Mobile Manipulators. *IFAC Proceedings Volumes*, 1(38), 43-48.

Thakar, S., Srinivasan, S., Al-Hussaini, S., M. Bhatt, P., & Rajendran, P. (2023). A Survey of Wheeled Mobile Manipulation: A Decision-Making Perspective. *Journal of Mechanisms and Robotics*, 2(15), 020801.

Uzuner, S., Akkuş, N., & Toz, M. (2020). 5-DOF serial robot manipulator design, application and inverse kinematic solution through analytical method and simple search technique. *Pamukkale University Journal of Engineering Sciences*, 2(26), 392-401.

Ünlü, A., & Yüksel, T. (2021). Dört Kanatlı İnsansız Hava Araçları İçin Yapay Sinir Ağları İle Konum Tabanlı Görsel Servolama Position Based Visual Servoing with Artificial Neural Networks

for Quadrotor-type Unmanned Aerial Vehicles. *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 1-4.

Vemuri, A., & Polycarpou, M. (2004). A methodology for fault diagnosis in robotic systems using neural networks. *Robotica*, 4(22), 419-438.

Wang, T., Zhang, L., & Wang, X. (2023). Fault Detection for Motor Drive Control System of Industrial Robots Using CNN-LSTM-based Observers. *Fault Detection for Motor Drive Control System of Industrial Robots Using CNN-LSTM-based Observers*, 7(2), 144-152.

Wilamowski, B., & Yu, H. (2011). Intelligent Systems. J. Irwin, & B. Wilamowski, *Intelligent Systems* 12-1-12-16.

Wu, Y.-c., & Feng, J.-w. (2018). Development and Application of Artificial Neural. *Wireless Personal Communications*, 1645-1656.

Xie, D., Zeng, L., Xu, Z., Guo, S., Cui, G. H., & Song, T. (2023). Base position planning of mobile manipulators for assembly tasks in construction environments. *Adv. Manuf. II*, 93-110.

Xu, D., Zhao, D., Yi, J., & Tan, X. (2009). Trajectory Tracking Control of Omnidirectional Wheeled Mobile Manipulators: Robust Neural Network-Based Sliding Mode Approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 3(39), 788-799.

Yamamoto, Y., & Yun, X. (1994). Coordinating locomotion and manipulation of a mobile manipulator. *IEEE Transactions on Automatic Control*, 39(6), 1326-1332.

Yu, Q., & Cheng, I.-M. (2002). A General Approach to the Dynamics of Nonholonomic Mobile Manipulator Systems. *J. Dyn. Sys., Meas., Control.*, 4(124), 512-521.

Yüksel, T., & Sezgin, A. (2009). Doğrusal Olmayan Dizgeler İçin Model Temelli. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 1(24), 79-94.

Yüksel, T., & Sezgin, A. (2010). Two fault detection and isolation schemes for robot manipulators using soft computing techniques. *Applied Soft Computing*, 125–134.