

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BİR ENDÜSTRİYEL ROBOTUN İNSAN KOLU HAREKETLERİNİN
DERİNLİK HARİTASI İLE ALGILANMASIYLA KONTROLÜ**

Makina Mühendisliği Anabilim Dalı

YÜKSEK LİSANS TEZİ

Burak MERT

Tez Danışmanı: Yrd. Doç. Dr. Yiğit TAŞCIOĞLU

MART 2016

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans/Doktora derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Doç. Dr. Murat Kadri AKTAŞ
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 121511113 numaralı Yüksek Lisans Öğrencisi **Burak MERT** 'in ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "**BİR ENDÜSTRİYEL ROBOTUN İNSAN KOLU HAREKETLERİNİN DERİNLİK HARİTASI İLE ALGILANMASIYLA KONTROLÜ**" başlıklı tezi **15, 03, 2016** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Yrd. Doç. Dr. Yiğit TAŞCIOĞLU**

TOBB Ekonomi ve Teknoloji Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Esra KADIOĞLU ÜRTİŞ (Başkan)**

TOBB Ekonomi ve Teknoloji Üniversitesi

Yrd. Doç. Dr. Kutluk Bilge ARIKAN

Atılım Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Burak MERT

İMZA

ÖZET

Yüksek Lisans Tezi

BİR ENDÜSTRİYEL ROBOTUN İNSAN KOLU HAREKETLERİNİN DERİNLİK HARİTASI İLE ALGILANMASIYLA KONTROLÜ

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü

Makine Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Yiğit Taşcıoğlu

Tarih: Mart 2016

Bu çalışmada Microsoft Kinect ToF kamera kullanılarak operatör kolunun hareketleri eşzamanlı olarak teleoperasyon yöntemiyle altı serbestlik derecesine sahip olan Stäubli Rx160 endüstriyel robot koluna aktarılmıştır. Kinect derinlik kamerası 640x480 çözünürlükte derinlik görüntülerini 30 Hz frekansta sağlayabilmektedir. Microsoft firmasının sağladığı Kinect SDK'da tanımlı fonksiyonlar kullanılarak bu derinlik görüntüleri alınmış ve operatörün derinlik haritası çıkarılmıştır. Daha sonra bu harita üzerinde operatörü çubuk figürü olarak modelleyecek 20 ekleminin üç boyutlu uzay pozisyonları iskelet haritası üzerinden takip edilmiştir. Sağ kolda bulunan omuz, dirsek ve bilek eklemlerinin pozisyonları kullanılarak eklemler arası uzuvları temsil eden vektörler hesaplanmıştır. Bu vektörler daha sonra operatörün kol açılarının bulunması sırasında kullanılmıştır. Operatörün kol açıları ve Stäubli Rx160 robot kolunun parametreleri kullanılarak operatör çalışma uzayı robotun çalışma uzayına eşlenecek biçimde genişletilmiştir. Bulunan açılar arasında gürültü içeren derinlik görüntüsü kareleri yüzünden hatalı olabileceklerin elenmesi için insan kolunun kinematik modeli oluşturularak düz

kinematik eşitlikleri hesaplanmıştır. Bulunan açılar ve düz kinematik eşitlikler kullanılarak bilek pozisyonu hesaplanmıştır. Her görüntü karesi için hesaplanan ve kameradan alınan bilek pozisyonları karşılaştırılarak hesaplanan açıların doğruluğu kontrol edilmiş, kinematik modele uymayan açılar filtrelenmiştir. Operatör açılarının bulunmasından sonra robotun teleoperasyonunu sağlamak için üç farklı yöntem geliştirilmiştir. Bu yöntemler uç işlemcinin kontrolünü sağlayan üç serbestlik derecesinde oluşmaktadır. Robotun son üç eklemi bilek yönelimini vermektedir. Bu çalışmada bilek yönelimi uç işlemcinin açık pozisyonda kalacağı şekilde sabitlenmiş ve teleoperasyon yöntemlerinde robotun ilk üç ekleminin açıları hesaplanmıştır. Bu yöntemlerden ilkinde operatörün kolu ile robot kolunun benzer kinematik yapıları eşlenerek direkt açı besleme yöntemi kullanılmıştır. Diğer yöntemde gerçek zamanlı ters kinematik çözümü uygulanmış, operatörün robot çalışma uzayına eşlenmiş bilek ekleminin pozisyonu kullanılarak bu pozisyonu sağlayacak robot eklem açıları bulunmuştur. Aynı uç işlemci pozisyonuna robot farklı şekillerde ulaşabilmektedir. Bu nedenle operatörün hareketine en benzeyen çözüm kümesinin kullanılması için filtreleme eşitlikleri kullanılmıştır. Son yöntemde ise ters kinematik çözümü yöntemi ile oluşturulan çözüm örneklerinden oluşan veri setleri ve MATLAB Neural Network Toolbox kullanılarak yapay sinir ağı eğitilmiştir. Farklı sayıda çözüm örneği içeren veri setleri ile eğitilen yapay sinir ağlarının performansları karşılaştırılarak sonuçlar yorumlanmıştır. Farklı yöntemler kullanılarak bulunan robot açıları TCP/IP protokolü ile robotun kontrolcüsüne iletilmiştir. Kontrolcü üzerinde koşan VAL3 programlama dilinde yazılmış uygulama ile TCP/IP üzerinden gelen değerler eklem açılara çevrilerek eklemlere iletilmiştir. Son olarak geliştirilen teleoperasyon yöntemleri doğruluk ve hesaplama zamanı kriterlerine göre karşılaştırılmış, çalışmanın devamında yapılabilecek araştırmalar ile ilgili öneriler sunulmuştur.

Anahtar Kelimeler: Teleoperasyon, Endüstriyel robot kolu, Kinect ToF kamera, Derinlik haritası, Ters kinematik, Yapay sinir ağı

ABSTRACT

Master of Science

TELEOPERATION OF AN INDUSTRIAL ROBOT ARM BY ANALYZING HUMAN ARM DEPTH IMAGE SEQUENCES

Burak Mert

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Mechanical Engineering Science Programme

Supervisor: Asst. Prof. Dr. Yiğit Taşcıoğlu

Date: March 2016

In this study, human operator's arm movements synchronized to 6 DOF Stäubli Rx160 industrial robot arm by using Microsoft Kinect ToF camera. Kinect can provide depth images at 30 Hz rate in 640x480 resolution. By using Kinect SDK that is provided by Microsoft, depth images are analyzed and operator's depth map is obtained. Once depth map is obtained operator is modelled with a stick figure with 20 joints. Each joints' three dimensional position data can be obtained from the depth map using Kinect SDK. Vectors that represents the bones between joints are calculated using joint positions on operator's right arm shoulder, elbow and wrist joints. These vectors are then used for calculation of operator's arm angles. Operator's workspace extended to Stäubli Rx160's workspace by using arm angles and Stäubli Rx160' dimensions. Human arm kinematic model and its forward kinematic equations are calculated for checking the validity of the calculated arm angles. Calculated angles are filtered using human arm forward kinematic equations

to eliminate angles that does not match human arm's kinematic model. After obtaining angles that match human arm kinematic model, three different teleoperation methods applied and tested. These methods are used to control Stäubli Rx160's end position with 3 DOF. Stäubli Rx160's last three joints represents orientation of its wrist joint. Wrist orientation is neglected in this study, thus last three joints of Stäubli Rx160 is setted to a position where end effector is in line with the robot arm's orientation. For the first method, direct angle calculation is used. In order to send angles directly, human arm joint anlges are mapped to Stäubli Rx160's joint angles. Inverse kinematics calculation method is used for the second method. Wrist position of the operator is calculated from the extended workspace parameters and used as robot's end effector position. Robot's joint angles are calculated from end effector position. Inverse kinematics calculations resulted in different solution sets where all sets lead to the given end effector position. These solution sets are filtered to obtain elbow-down operation mode which resembles operator pose. For the third teleoperation method, an artificial neural network is trained using data sets obtained from inverse kinematics solution method. Two different neural networks are trained using Matlab's Neural Network Toolbox. Accuracy of the trained networks, which are trained with different number of data points, are compared and discussed. Calculated angles are sent to Stäubli Rx160's emulator using TCP/IP protocol. A VAL3 program on the emulator accepted the connection and read data from the incoming stream. Then this program converts incoming data stream back to robot joint angles and sends angles to corresponding joints motor driver. Developed teleoperation methods are compared against each other by means of accuracy and calculation time performance. Results are discussed and recommendations for future work are presented at the conclusion.

Keywords: Teleoperation, Industrial robot arm, Kinect, Depth map, Inverse kinematics, Artificial neural network

TEŞEKKÜR

Çalışmalarım boyunca bana sağladığı destek ve verdiği yardım ve katkılarıyla beni yönlendiren ve bana destek olan değerli danışmanım Yrd. Doç. Dr. Yiğit Taşçıoğlu'na, bana verdikleri destek ve kattıkları vizyon için hocalarım Doç. Dr. Selin Aradağ Çelebioğlu ve Dr. Kutay Çelebiğolu'na, bütün Etü Hidro ekibine ve özellikle ofis arkadaşım Umut Aradağ'a teşekkürü borç bilirim. Sabah akşam demeden uygun olduğu bütün saatlerde labovatuarda bulunarak robot ile çalışmama yardımcı olduğu Ahmet Aksel'e çok teşekkür ederim Aynı zamanda bana sağladığı Ar-Ge bursu için TOBB ETÜ'ye teşekkür ederim.

Ayrıca ODTÜ Teknokent Arçelik Ar-Ge ofisindeki bütün çalışma arkadaşlarıma tek tek dostlukları, hayatıma kattıkları eğlenceli ofis ortamı ve destekleri için teşekkür ederim. Takım liderim Güven Orkun Tanık'a da bu tezin hazırlanmasında geçen zor zamanlarımda bana gösterdiği için hoşgörü ve verdiği destek için ayrıca teşekkür ederim.

Varlıkları ile beni mutlu eden, iyi kötü bütün günlerimde yanımda olan değerli dostlarım Özgür Deniz Canat, Merve Canat, Bahadır Artmaz, Çağla Şanver (yakında Artmaz olacak), Alişan Kürekçi, Onur Etöz ve Dilege Gülmez'e hissettiklerimi kelimeler ile anlatamam da usülen teşekkürü borç bilirim. Aynı zamanda son 8 aydır bizimle olan, küçük grubumuzun en küçük üyesi Mavi Deniz Canat'a da hayatımıza renk kattığı ve bizlere tarif edilemez bir mutluluk yaşattığı için çok teşekkür ederim.

Tezimde bulunan görsellerin hazırlanmasında bana destek olan, hayatımı güzelleştiren ve mutluluk katan, benimle beraber yürüyen ve sonsuz sevgisini, desteğini benden hiçbir zaman esirgemeyen en değerli varlığım Gizem Memiç'e çok teşekkür ederim.

Son olarak da en önemli teşekkürümü bana verdikleri sonsuz destekleri ve sevgileri için, bana hissettirdikleri güven için, her düştüğümde yanımda oldukları için, bazen günlük hayatın koşuşturmasından onlarla yeterince vakit geçirip konuşmasam da her konuşmak istediğimde beni can kulağıyla dinledikleri ve her anımda yanımda oldukları için başta canım anneme ve babama olmak üzere bütün aileme armağan etmeyi borç bilirim. Sizleri çok seviyorum.



İÇİNDEKİLER

Sayfa	
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	x
ŞEKİL LİSTESİ	xii
ÇİZELGE LİSTESİ	xiv
KISALTMALAR	xvi
SEMBOL LİSTESİ	xviii
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	3
2.1 Görüntü İşleme Temelli Hareket Analizi Sistemleri	3
2.2 Kızılötesi Sensör ve Uçuş Zamanı Kameralar Kullanılarak Yapılan Görüntü İşleme Çalışmaları	10
2.3 Kinect ile Robotik Alanında Yapılan Görüntü İşleme Çalışmaları	18
2.3.1 Kinect Kullanılarak Humanoid Robotlar Üzerinde Yapılan Çalışmalar ...	19
2.3.2 Kinect Kullanılarak Endüstriyel Robotlar Üzerinde Yapılan Çalışmalar..	28
3. DERİNLİK HARİTASI KULLANARAK İSKELET TAKİBİ	37
3.1 Eklem Açılarının Hesaplanması	39
3.2 İnsan Kolu Düz Kinematığı.....	45
3.3 İnsan Kolu Çalışma Alanının Robot Çalışma Uzayına Eşlenmesi.....	48
4. TELEOPERASYON YÖNTEMLERİ	51
4.1 Direkt Eşleme Yöntemi	52
4.2 Eş zamanlı Ters Kinematik Çözümü Yöntemi	53
4.3 Yapay Sinir Ağı Yöntemi	62
5. SONUÇLAR VE KARŞILAŞTIRMA	69
5.1 Pozların Görsel Karşılaştırılması.....	70
5.2 Ters Kinematik Yöntemi ve Yapay Sinir Ağı Yöntemlerinin Performans Karşılaştırmaları	75
5.2.2 Doğruluk Performansı Ölçümü	76
6. SONUÇLAR	89
KAYNAKLAR	93
EKLER	100
ÖZGEÇMİŞ	101



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1- Psi(ψ) pozı ile ilklendirme yapılmıř ve derinlik haritası kullanılarak arka plandan ayıklanmıř model [6]	4
Şekil 2.2- İnsan modeli gösteriliři ve oluřturulan 3 boyutlu hareket modeli [8].	5
Şekil 2.3- Görüntünün köřelerinin tespit edilmesi, t+1 anı için yapılan tahmin sonucunda filtrelenmiř görüntü ve oluk ¹ görüntüsü [9].	6
Şekil 2.4- Pointgrey triclops stereo kamera derinlik histogramı [20].	9
Şekil 2.5- RGB görüntü, RGB görüntüsünden elde edilen ön plan maskesi ve derinlik görüntüsünden elde edilen ön plan maskesi [27].	11
Şekil 2.6- Poz düzenleme algoritması kullanılmadan yapılan poz kestirimi [31] ve poz düzenleme algoritması kullanılarak iyileřtirilen poz kestirimi [32]. ..	13
Şekil 2.8- El pozisyonlarının derinlik görüntüsünden algılanması [35].	15
Şekil 2.9- P noktası parmak ucu olarak seçilmiř ve p noktası parmak ucu olarak iřaretlenmemiř el haritaları [36].	16
Şekil 2.10- Parmak ucu hareketleri takip edilerek geliřtirilen uygulamalar [36].	16
Şekil 2.11- Derinlik görüntüsünden algılanan el (üst), tahmin edilen eklem pozisyonları (orta), oluřturulan el iskeleti (alt) [39].	18
Şekil 2.12- İnsan kolu üzerinden hesaplanan açılar [41].	19
Şekil 2.13- Direk besleme ile gönderilen açılar sonrası operatör ve robotun pozisyonları [41].	20
Şekil 2.14- Omuz açılarının zyz euler açıları notasyonu ile gösterilmesi [52]	25
Şekil 2.15- Hesaplanan açıların robot üzerinde gösterilmesi [52].	25
Şekil 2.16- Dirsek eklemi açısı ve alt kol oryantasyonunun robot üzerinde gösterimi [52].	26
Şekil 2.17- Kullanıcının kolunun derinlik görüntüsünden segmente edilmesi [56]..	29
Şekil 2.18- YZ düzleminde hesaplanan maksimum ve minimum noktaları [56].	30
Şekil 2.19- Çalışmada kullanılan statik pozlar [57].	31
Şekil 2.20- (Kırmızı) filtre uygulanmadan önce hareket yörüngesi, (mavi) filtre uygulandıktan sonra düzleřtirilmiř hareket yörüngesi [58].	33
Şekil 2.21- Operatörün sanal ortamda bulunan bir objeyi seçme hareketinin algılanması [58].	33
Şekil 2.22- Derinlik görüntüsünden imu koordinatlarının elde edilmesi (sol), camshift algoritması kullanılarak elin takip edilmesi (sağ) [59].	34
Şekil 3.1- Microsoft kinect'ten alınan derinlik görüntüsü ve iskelet bilgisi.	38
Şekil 3.2- İskelet eklemlerinin hiyerarřik yapısı.	39
Şekil 3.3- Kinect koordinat sistemi [68].	41
Şekil 3.4- İskelet modeli kol açıları.	41
Şekil 3.6- θ_1 açısının negatif ve pozitif deđerleri.	44
Şekil 3.7- θ_3 açısını gösteren hareket yörüngesi.	44

Şekil 3.8- θ_4 açısını gösteren kol hareketi.....	45
Şekil 3.9- Stäubli rx160 bağ uzunluğu ve bağ ofseti parametreleri	48
Şekil 4.1- Genel sistem mimarisi	51
Şekil 4.2- İki katmanlı ileri beslemeli sinir ağı mimarisi.....	65
Şekil 4.3- Aynı uç işlemci pozisyonu için oluşan farklı pozlar [70].....	66
Şekil 4.4- Aynı uç işlemci noktasını veren farklı pozlar.....	67
Şekil 5.1- Açılı değerlerin iletmek için oluşturulan istemci/sunucu mimarisi.....	70
Şekil 5.2- Poz 1: (a) operatör, (b) direkt besleme yöntemi, (c) ters kinematik çözümü, (d) yapay sinir ağı.....	71
Şekil 5.3- Poz 2: (a) operatör, (b) direkt besleme yöntemi, (c) ters kinematik çözümü, (d) yapay sinir ağı.....	72
Şekil 5.4-Poz 3: (a) operatör, (b) direkt besleme yöntemi, (c) ters kinematik çözümü, (d) yapay sinir ağı.....	73
Şekil 5.5- Poz 4: (a) operatör, (b) direkt besleme yöntemi, (c) ters kinematik çözümü, (d) yapay sinir ağı.....	74
Şekil 5.6- Doğruluk karşılaştırması figür 1.....	77
Şekil 5.7- Doğruluk karşılaştırması figür 2.....	78
Şekil 5.8- Doğruluk karşılaştırması figür 3.....	78
Şekil 5.9- Doğruluk karşılaştırması figür 4.....	79
Şekil 5.10- Figür 1 hata histogramı.....	80
Şekil 5.11- Figür 2 hata histogramı.....	80
Şekil 5.12- Figür 3 hata histogramı.....	81
Şekil 5.13- Figür 4 hata histogramı.....	81
Şekil 5.14- Doğruluk karşılaştırması figür 1- yapay sinir ağı 2.....	82
Şekil 5.15- Doğruluk karşılaştırması figür 2- yapay sinir ağı 2.....	83
Şekil 5.16- Doğruluk karşılaştırması figür 3- yapay sinir ağı 2.....	83
Şekil 5.17- Doğruluk karşılaştırması figür 4- yapay sinir ağı 2.....	84
Şekil 5.18- Figür 1 hata grafiği- yapay sinir ağı 2.....	84
Şekil 5.19- Figür 2 hata grafiği- yapay sinir ağı 2.....	85
Şekil 5.20- Figür 3 hata grafiği- yapay sinir ağı 2.....	85
Şekil 5.21- Figür 4 hata grafiği- yapay sinir ağı 2.....	86

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1- Taranan çalışmalarda kullanılan yöntemler.....	28
Çizelge 3.1- İnsan kolu dh parametreleri.....	46
Çizelge 3.2- Doğrulama fonksiyonu sahte kodu.....	47
Çizelge 5.1- Ters kinematik ve yapay sinir ağı çözümlerinin hesaplama performansı testleri.....	76





KISALTMALAR

ToF	: Time of Flight kamera
RGB	: Red-Green-Blue(Renkli) kamera
AÖB	: Ataletsel ölçüm birimi
SDK	: Software Development Kit
DH	: Denavit Hartenberg
NUI	: Natural User Interaction
NNF	: Neural Network Fitting





SEMBOL LİSTESİ

Simgeler

Açıklama

$r_{i,j}$

Rotasyon matrisi i, j elemanı

p_i

i eksenindeki pozisyon

$V_{i,j}$

i, j eklemleri arasındaki vektör

m

eğim

R_i

i ekseninde rotasyon matrisi

$R_{i,\theta}$

i ekseninde θ açısı kadar dönme

a

Uzuv uzunluğu

d

Uzuv ofseti

α

Uzuv bükümü

θ

Ekleme açısı

T

Dönüşüm matrisi(Robot)

A

Dönüşüm matrisi(İnsan Kolu)



1. GİRİŞ

Günümüzde hizmet, sağlık ve güvenlik alanında yaşanan dijitalleşme hareketi ile beraber teknolojik gelişmeler farklı bir boyut kazanmıştır. Bu alanların hemen hemen hepsinde yer edinen robotik teknolojisi de yeni çağın en çok araştırılan ve çalışma yapılan konularından birisi olmuştur. Robotik kendi içerisinde birçok farklı alanda araştırma konusu içermesi ile beraber etrafında gelişen diğer teknolojilerle entegre bir şekilde de hızla gelişmektedir.

Son yıllarda hızla artan bu teknolojik gelişmelerden gerek akademik gerekse ticari anlamda en çok araştırma yapılan konulardan biri kızılötesi ışınların uçuş süresini hesaplayabilen uçuş zamanı kameralarıdır. Bu kameraların çalışma prensibi üzerlerinde bulunan kızılötesi yayıcı ve sensörüne dayanır. Yayıcıdan saçılan ışınlar karşısındaki hedefe çarpıp sensör tarafından algılandığında ışının havada geçirdiği süre ölçülür. Işığın hızı ve havada geçirdiği süreden karşısındaki hedefin kameraya olan uzaklığı hesaplanır. Bu sayede kamera önünde bulunan objelerin üç boyutlu uzayda kameraya göre pozisyonları hesaplanmış olur. Bu kameralar ilk yıllarda özellikle eğlence ve oyun alanlarında kullanılmış olsalar da gün geçtikçe özellikle robotik ve görüntü işleme alanında kendilerine yer edinmeyi başarmışlardır [1]. Microsoft firmasının geliştirdiği Kinect [2] sensörü bu ToF kameraların en gelişmişlerinden biridir.

ToF sensörlerin son kullanıcıların alıp kullanabileceği seviyeye gelmesi ile beraber özellikle robotik alanında önceden pahalı hareket algılama sistemleri ile özel laboratuvarlarda yapılan çalışmalar günümüzde ofis ortamında kullanılan donanımın desteklediği hassasiyette yapılabilir hale gelmiştir. Özellikle teleoperasyon yöntemlerinde operatörün üzerine herhangi bir ek donanım yerleştirilmeden sadece ToF kamera karşısında yaptığı hareketlerin algılanması bu alandaki çalışmalar içerisinde Kinect'in kullanılmasını cazip kılmıştır.

Teleoperasyon çalışmalarının büyük çoğunluğunda merkez-uydu yöntemi kullanılarak merkez olarak kabul edilen birimin hareketlerinin algılanarak uydu birimine aktarılması vardır. Uzaktan kontrol ve merkez-uydu yöntemleri özellikle humanoid robotlar ve endüstriyel sağlık robotlarında kullanılmaktadır [3]. Bu alanda yapılan çalışmalar Bölüm 2’de taranmıştır.

Bu çalışmada da benzer bir yöntem izlenmiştir. Kinect kamera ile robotu kontrol eden operatör merkez birimi olarak kabul edilmiş, kontrol edilen Stäubli Rx160 endüstriyel robot kolu ise uydu birimi olarak kabul edilmiştir. Bölüm 3’te Kinect kullanılarak yapılan derinlik haritası işleme çalışmaları anlatılmıştır. Bu çalışmalarda merkez birim olan operatörün iskelet takibi yapılarak eklem açıları ve uzuv parametreleri hesaplanmıştır. Daha sonra hesaplanan veriler uydu birimin çalışma alanına eşlenmiş ve bu çalışma alanı içerisinde Stäubli Rx160 robot kolunun eklem parametreleri anlık olarak hesaplanmıştır. Çalışma uzayı eşlenmesi yapılırken insan uzuvlarının boyutlarının genişletilerek robot boyutlarına getirilmesi senaryosu göz önüne alınmıştır. Bu senaryo seçilirken robot kullanılarak insanların uzanamayacağı yerlere uzanma ve taşıyamayacağı ağırlıkları taşıma gibi işlevler göz önünde bulundurulmuştur. Bölüm 4’te operatörün çalışma uzayında hesaplanan eklem açıları ve üç boyutlu pozisyon parametreleri robotun çalışma uzayın aktarılarak 3 farklı yöntem ile robot eklem açıları hesaplanmıştır. Bunlardan ilkinde operatör ve robotun benzer eklemleri kinematik olarak eşlenerek direkt açı besleme yapılmıştır. Diğer yöntemde operatörün sağ elinin bilek koordinatı robot çalışma uzayına eşlendikten sonra bu pozisyondan ters kinematik çözümü yapılarak robotun eklem açıları hesaplanmıştır. Son yöntemde ise MATLAB yapay sinir ağı aracı kullanılarak bir sinir ağı eğitilmiş ve operatör eklem açıları ve bilek koordinatları kullanılarak robot eklem açılarını verecek bir eşleme oluşturulmuştur. Bulunan robot eklem açıları TCP/IP protokolü kullanılarak robotun kontrolcüsüne aktarılmıştır. Kontrolcünün eklem açılarını okuyarak robotu harekete geçirebilmesi için VAL3 [4] programlama dili kullanılarak kontrolcü üzerinde çalışan bir uygulama geliştirilmiştir. Bölüm 5’te bu yöntemlerin sonuçları karşılaştırılmıştır. Bölüm 6’da ise elde edilen sonuçlar tartışılarak bundan sonra yapılabilecek çalışmalar ile ilgili fikir yürütülmüştür.

2. LİTERATÜR TARAMASI

Bu bölümde yapılan çalışmaya temel oluşturacak olarak fayda sağlayabilecek geçmişte yapılan çalışmalar taranmıştır. Bölüm 2.1’de görüntü işleme alanındaki gelişmelerin geçmişten günümüze nasıl geliştiği ile ilgili çalışmalar özetlenmiştir. Daha sonra Bölüm 2.2’de ToF kameralar ve kızılötesi sensörler kullanılarak yapılan hareket analizi sistemleri özetlenmiştir. ToF kamera olan Kinect ile yapılan görüntü işleme çalışmaları, bizim çalışmamızla doğrudan ilgili olduğu için ayrı bir başlık altında Bölüm 2.2.1’de incelenmiştir.

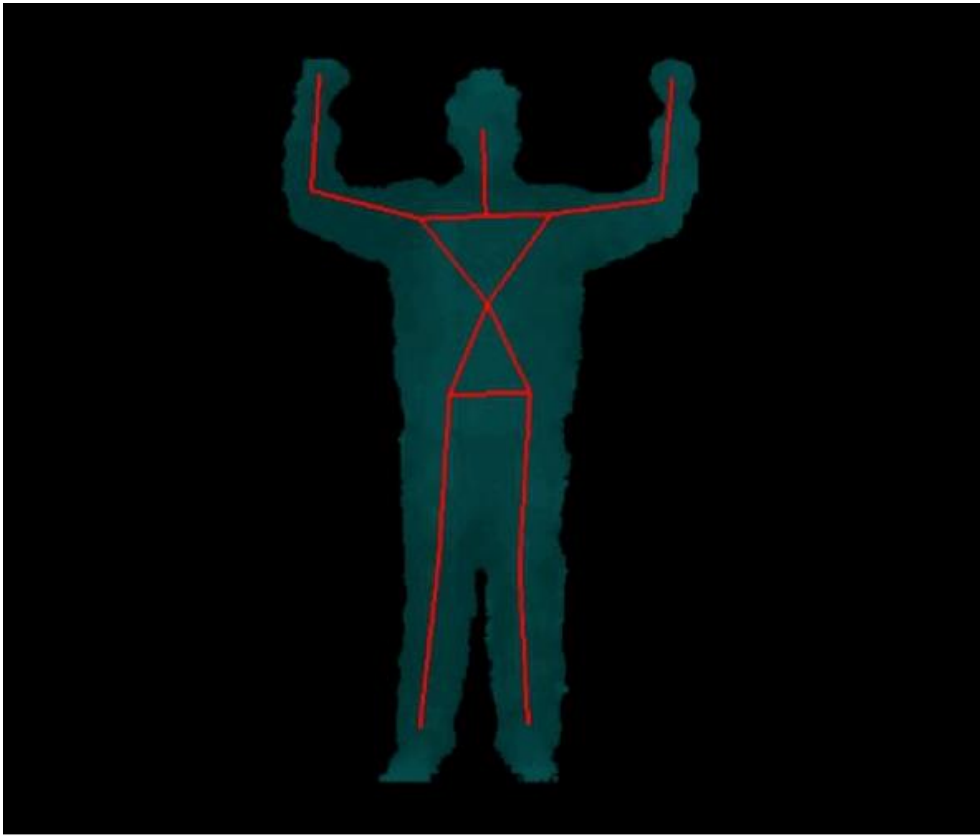
Bölüm 2.3’te Kinect kullanılarak robotik alanında yapılan uzaktan kontrol çalışmaları özetlenmiştir. Bu çalışmalar incelenirken humanoid robotlar ve endüstriyel robotlar ayrı başlıklar altında anlatılmıştır.

2.1 Görüntü İşleme Temelli Hareket Analizi Sistemleri

Görüntü işleme ile hareket analizi sistemleri özellikle robotik alanında çok kullanılan sistemlerdendir. Bu sistemlerin amaçlarından biri insan uzuvlarının algılanarak hareketlerinin takip etmektir. Görüntü işleme ile hareket analizi sistemleri, en üst seviyede, aktif algılama ve pasif algılama olarak ikiye ayrılır [5]. Aktif sistemlerde hareket objenin üzerine sinyal alışverişi yapabilen cihazlar yerleştirilerek algılanır. Pasif sistemlerde ise hareket algılama obje üzerine herhangi bir cihaz yerleştirilmeden, doğal sinyal kaynakları (ışık, elektromanyetik dalgalar v.b.) kullanılarak yapılır. Bununla beraber obje üzerine sinyal alışverişi yapmayan işaretçilerin yerleştirilmesi ile çalışan hareket algılama sistemleri de pasif sistemler olarak görülebilirler.

Literatürde bulunan görüntü işleme ile hareket analizi yapan sistemler incelendiğinde temel olarak aynı aşamaları içerdikleri görülür. Bu aşamalardan ilki iklendirme aşamasıdır. Bu aşamada yapılan işlemler kamera kalibrasyonu, model iklendirme, takip edilen objenin bulunduğu arka planın karakteristik özelliklerinin belirlenmesi

gibi işlemlerdir. Ayrıca çoğu pasif görüntü algılama sistemi model takibine başlamak için model tabanlı poz ilklendirme işlemini bu aşamada gerçekleştirir. Poz ilklendirme sisteminde takip edilecek olan modelin daha önceden belirlenmiş bir poza gelerek sistemi ilklendirmesi gerekmektedir. Daha sonraki aşama takip aşamasıdır. İlk aşamada saptanan obje çeşitli yöntemler kullanılarak arka plandan ayıklanarak takibe hazır hale getirilir. Şekil 2.1’de poz ilklendirme ve arka plan ayıklama işlemleri yapılmış ve sistem tarafından takip edilen bir model gösterilmiştir. Daha sonra takip edilen objenin vücut pozunu tahmin edilir. Son aşamada ise objenin pozunu analiz edilerek algılama yapılır.

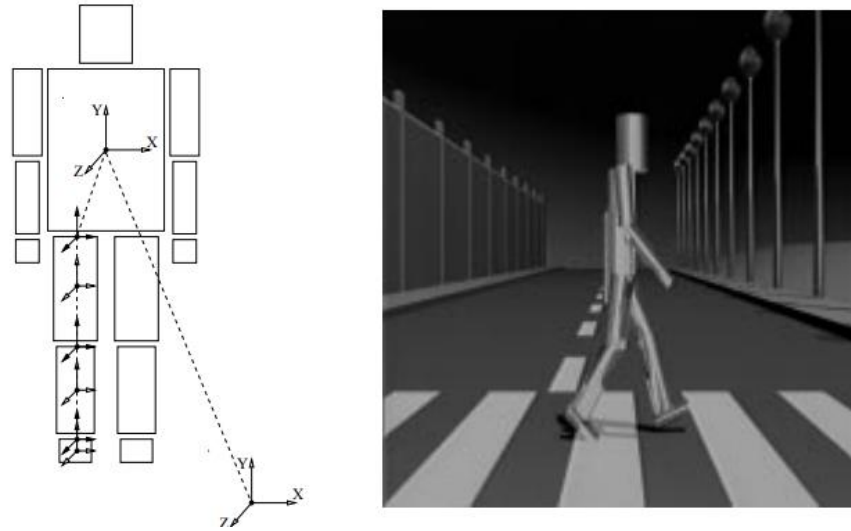


Şekil 2.1- Psi(ψ) pozunu ile ilklendirme yapılmış ve derinlik haritası kullanılarak arka plandan ayıklanmış model [6]

Görüntü verilerini incelemek için birçok seviyede karşılaştırma ve analiz yapılmaktadır. Görüntü analizini kolaylaştırmak için köşeler, silüetler, konturlar, eklemler, derinlik ve tekstür gibi soyutlama seviyeleri kullanılarak yapılan çalışmalar bulunmaktadır.

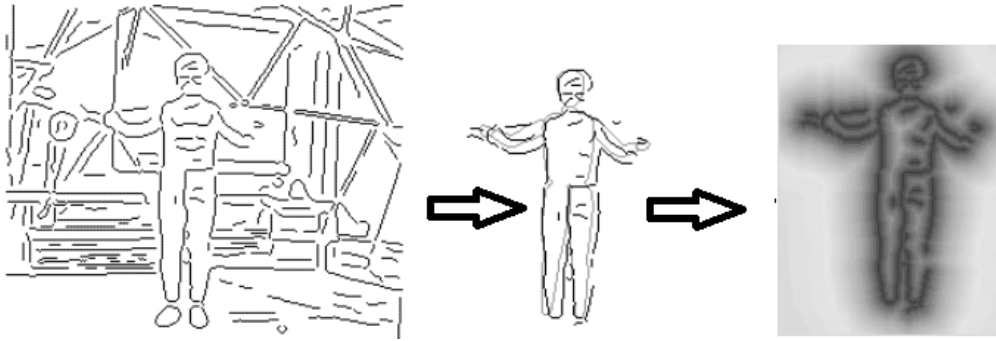
Bu soyutlama seviyelerinden en yaygın kullanılanlar köşe ve kontur tespiti aracılığıyla sınır bölgeler bulunarak analiz yapan yöntemlerdir. 1983 yılında Hogg tarafından yapılan araştırmada görüntü çıkarma yöntemi kullanılarak insan vücudunu çevreleyen bir kutu oluşturulmuştur [7]. Daha sonra oluşturulan bu sınır içerisinde köşeler bulunmuş ve bu köşeler takip edilen insan modelindeki köşeler ile karşılaştırılmıştır.

Daha sonra Rohr [8] benzer bir çalışmada, yürüyen bir insan modelinin üst gövde merkez noktasının üç boyutlu koordinatlarını kullanarak ve hareketin görüntü düzlemine paralel olduğu varsayımıyla üst gövde merkez noktasının hareket modelini oluşturmuştur. İkleme aşamasında arka arkaya gelen birbirinden bağımsız 10-15 görüntü incelenerek hareket eden insana bağlı olarak değişen görüntülerde sınır bölge tespiti yapılmış, üç boyutlu pozisyonlar ve poz aracılığıyla hareket durumlarının çıkarımı yapılmış ve Kalman filtresinin başlangıç değerleri belirlenmiştir. Kalman filtresi iklemesi yapıldıktan sonra da filtre gelen her görüntüye uygulanarak hareket durumlarının tahmini yapılmış, görüntülerden ve oluşturulan üç boyutlu modelden alınan sonuçlar karşılaştırılmış ve son olarak son hareket durumunun çıkarımı yapılarak üç boyutlu model hareket ettirilmiştir. Kullanılan yaklaşım ve oluşturulan hareket modeli Şekil 2.2’de gösterilmiştir.



Şekil 2.2- İnsan modeli gösterilişi ve oluşturulan 3 boyutlu hareket modeli [8].

Aynı problem Garvila ve Davis [9] tarafından çalışılmış ancak bu çalışma daha önceden bilinen bir hareket modeli olmadan yapılmıştır. Daha önceden bilinen bir hareket modeli yerine dört kameradan alınan görüntüler farklı görüş açılarından karşılaştırılmıştır. Burada birden fazla kamera kullanılarak çok açıdan alınan görüntülerden üç boyutlu vücut pozunu tekrar oluşturulmuştur. Oluşturulan 3 boyutlu vücut pozunu hareket algılama sistemine girdi olarak verilmiştir. Hareket takibi çalışması dört ana bölüme ayrılmıştır. İlk bölüm tahmin bölümüdür. Bu bölümde t anına kadar olan durumlar gözlemlenerek $t+1$ anı için bir tahmin çıkarımı yapılır. Daha sonra gelen sentez ve görüntü analizi bölümlerinde görüntü ve model arasında benzerlik karşılaştırılması yapılır. Bu karşılaştırmada görüntü ve modelin köşeleri ve çevritleri kontur tabanlı şekil karşılaştırma yöntemi ile karşılaştırılır. Son olarak da durum kestirimi yapılarak yeni durum bulunur. Bu çalışmada yapılan çevrit tabanlı şekil karşılaştırma aşamaları Şekil 2.3 'te gösterilmiştir.



Şekil 2.3- Görüntünün köşelerinin tespit edilmesi, $t+1$ anı için yapılan tahmin sonucunda filtrelenmiş görüntü ve oluk görüntüsü [9].

Silüetler ve silüetlerin çevritlerinin modelden ve görüntüden çıkarılması yaygın olarak kullanılan bir yöntemdir. Silüet bölgesel tabanlı olduğu için gürültüye ayrıt tabanlı yöntemlere göre daha az hassastır. Bu yöntemi kullanan bir çalışmada görüntüden çıkarılan silüet modelden çıkarılan silüetle karşılaştırılmıştır [10]. Bu karşılaştırma yapılırken bölgesel eşleşme stratejisi kullanılarak alan karşılaştırılması yapılmış, negatif ve pozitif eşleşme hataları eşleşme ağırlığı katsayıları ile ifade edilmiştir.

Takip edilen insan modeli çoğu zaman eklem verileri kullanılarak bir çubuk figürü ile temsil edilmektedir. Ancak bu gösterimin yapılabilmesi için modelin eklemlerinin

belirlenmesi ve takip edilmesi gerekmektedir. Lee ve Chen [11] tarafından yapılan çalışmalarda görüntü üzerindeki eklem pozisyonları ve eklemleri birbirlerine bağlayan uzuvların boyutları önceden bilinerek ikili ağaç yapısı oluşturulmuştur. Boyundan başlayarak oluşturulan bu ikili ağaç yapısında bir sonraki bağlantı, bir önceki bağlantının başlangıç pozisyonu ve aralarındaki uzuv boyutu bilinerek dönüşüm ve rotasyon matrisleri oluşturularak hesaplanmıştır.

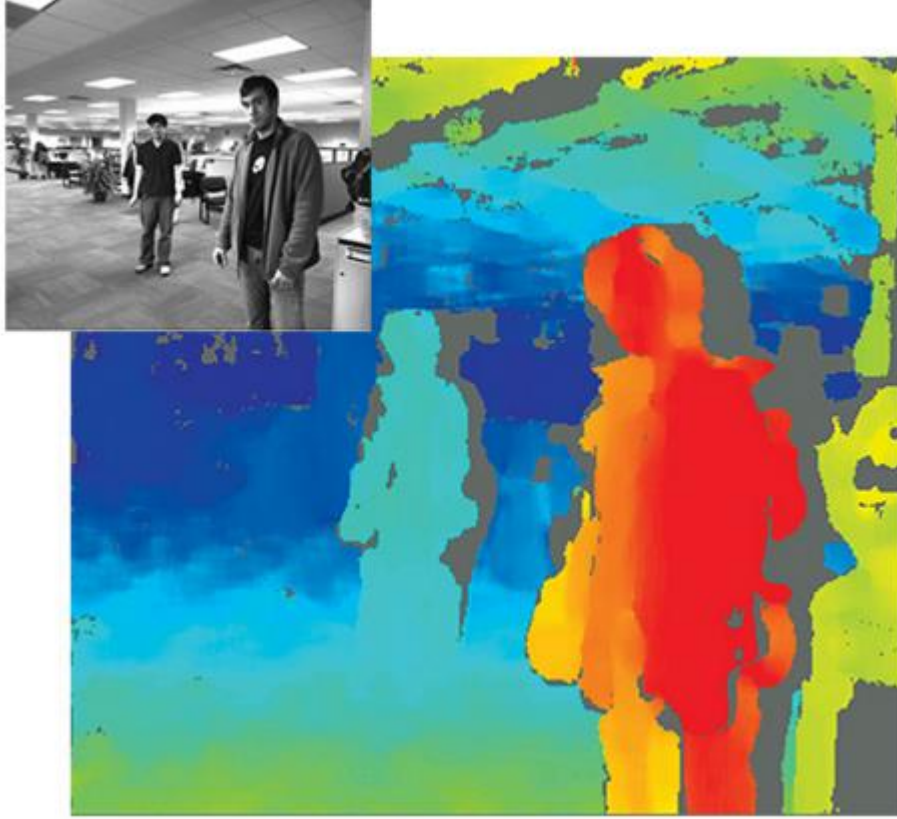
Bu hesaplamalar yapılırken gözlem yapılan modelin yürüdüğü varsayımı yapılmış ve model üzerinde kinematik kısıtlamalar yapılmıştır. İnsan eklemlerinin yapabileceği açılar tanımlanarak kinematik kısıtlar belirlenmiştir. Bu çalışma bize takip edilen modelin eklem pozisyonları takip edilerek ve insan iskeletini temsil etmesi için çubuk figürü modeli kullanılarak görüntü işleme analizi yapılabileceğini göstermiştir. Aynı ekip bir diğer çalışmalarında [12] bir önceki çalışmalarına ek olarak sisteme düzgün hareket kısıtı getirmişlerdir.

Düzgün ve sürekli bir hareket tanımında vücut segmentlerinin sabit bir açısal hıza ve dolayısıyla sifıra yakın bir açısal ivmelenmeye sahip olacağı öngörülmüştür. Görüntü kareleri arasında düzgün hareket ölçümü yapabilmek için vücut konfigürasyonları ile bağlantılı açısal ivmelenme fonksiyonu oluşturulmuştur. Daha sonra görüntü karelerinden düzgün ve sürekli bir hareket formu bulma işlemi bir çizge arama problemi olarak formüle edilmiştir. Atwood [13] tarafından gerçekleştirilen çalışmada çözüm uzayı statik pozlar kullanılarak daraltılmıştır. Vücut duruşu, bir önceki çalışmaya benzer olarak, eklem açılarının yapabileceği açılar kinematik olarak kısıtlanması ile tanımlanmıştır. Eklem açısının belirlenen pozisyonlarda eksenler etrafındaki açısı yayılım olasılık yoğunluğu fonksiyonu ile belirtilmiştir. Her pdf eklem açısı ile eksen arasındaki ilişkiyi ve bu açının olası bütün açı yayılım aralığında oluşabilme olasılığını göstermektedir. Önceden belirlenen vücut pozlarının açılardan oluşan bir başvuru çizelgesine başvurularak bu pozda tüm eklemlerin olası açıları ile gözlemlenen modelin eklem açıları olasılık yoğunluk fonksiyonu kullanarak karşılaştırılmış ve poz algılama yapılmıştır. Bir diğer çalışmada ise [14] öznitelik ve stereoskopik kamera yöntemleri kullanılarak eklemlerin üç boyutlu pozisyonları bulunmuştur. Bu çalışmada hiyerarşik bir obje modeli bulma sistemi geliştirilmiştir. Takip edilen obje iç yapı, geometrik model ve öznitelik modeli olarak

incelenmiştir. İç yapı modeli eklemler ve uzuvlardan oluşan çubuk figürü olarak modellenmiştir. Geometrik model iç yapı üzerinde oluşturulan çubuk figürü olarak modellenmiştir. Geometrik model iç yapı üzerinde oluşturulan silindirler ve elipslerden oluşan modeli göstermektedir. Öznitelik modeli ise ayırt etmeye yardımcı olması için kullanılan özelliklerden oluşmaktadır. Bu özelliklerden en çok kullanılanı renk olmuştur. Modifiye edilmiş gözetimli öğrenme algoritması kullanılarak renk değişimlerinin varyasyonları kontrol edilmiştir. Bu algoritma kullanılarak öznitelik vektörü öncelikle sınıflandırılmış, daha sonra sınıflandırma ağırlıkları sürekli olarak güncellenerek denetimli öğrenme sağlanmıştır [15]. Daha sonra belirlenen öznitelikler ve stereo görüntü kullanılarak eklemlerin üç boyutlu pozisyonları bulunmuştur. Bir diğer çalışmada modelden oluşturulan çubuk figürü, siluet çıkarımı yapılarak elde edilen görüntüdeki iskelet modeli ile karşılaştırılmıştır [16]. Bu çalışmada potansiyel alan kavramı tanımlanmıştır. Bu tanıma göre problem potansiyel alan içerisinde minimum enerjiye sahip olan çubuk figürünü bulmaya dönüşmüştür. Diğer çalışmalarda olduğu gibi eklemler üzerinde kinematik açı kısıtlamaları uygulanmış ve tahmin yöntemi kullanılarak eşleştirme işlemi kolaylaştırılmıştır.

Takip edilen modelin eklem pozisyonlarını algılamak için ya da arka plan çıkarımı yapmak için üç boyutlu uzayda derinlik ölçümü yapmak literatürde kullanılan yöntemlerden bir diğeridir. Plänkners ve grubu yaptıkları bir çalışmada [17] üç kamera kullanarak çekilen video görüntülerinden takip edilen kişinin sağ kol derinlik verilerini elde etmişlerdir. Üç adet kalibre edilmiş kamera tarafından çekilen çoklu-görüntüler üzerinde en küçük kareler yöntemi kullanılarak eşleme işlemi yapılarak benzeşmeler bulunmuştur. Bu benzeşmenin aranacağı bölge video karesi üzerinden elle seçilerek işaretlenmektedir. Daha sonra seçilen noktaların üç farklı kareden takip edilmesi ile üç boyutlu yörünge hesaplanmış ve üç boyutlu nokta bulutu ve yüzeyler oluşturulmuştur. Yapılan bu çalışmalar dar kıyafetler kullanılarak yapılmıştır. Daha sonra elde edilen üç boyutlu nokta bulutundan eklemlerin pozisyonları belirlenmiş ve çubuk figürü ile iskelet modeli oluşturulmuştur. Bir diğer çalışmada [18] stereo kameralar kullanılarak derinlik-yoğunluk haritası oluşturulmuştur. Oluşturulan derinlik haritasından histogram veya morfolojik teknikler kullanılarak arka plan çıkarımı yapmanın görüntü üzerinde yoğunluk farkı kullanılarak arka plan çıkarımı

yapmaktan daha kolay olduđu belirtilmiřtir. Kendi alıřmalarında da arka plan ıkarımı yapmak iin uyumsuzluk histogramından eřik deđer kestirimi yapmıřlardır. Bunun iin PointGrey tarafından retilen Triclops stereo sistemini kullanmıřlardır [19]. Bu sistem zerinde bulunan stereo kamera sayesinde aldıđı farklı grnt karelerindeki pikseller arasındaki uyumsuzluđu histogram řeklinde vermektedir. Triclops stereo sisteminden alınan grnt rneđi řekil 2.4'te gsterilmiřtir.



řekil 2.4- PointGrey Triclops stereo kamera derinlik histogramı [20].

Eklem ve uzuv takibi yapabilmek iin kullanılan bir diđer yntem de dokulardan yararlanmaktır. Lerasle ve alıřma grubunun yaptıkları alıřmalarda [21] [22] doku gsterimi kullanılmıřtır. Takip edilen insanın bacađının modeli yaratılmıř, bu model alıřmada objenin giydiđi tayt ile doku kaplama yntemiyle kaplanmıřtır. Daha sonra farklı kameralardan alınarak birleřtirilen grntlerde objenin giydiđi tayt dokusu ile modelin zerindeki doku korelasyonu hesaplanarak bacađın  boyutlu hareketi ıkarılmıřtır. Bir diđer alıřmada da takip edilen insan silindirler ile modellenmiř ve hareket izleme sistemi yardımıyla takip edilmiřtir. Aynı zamanda grnt sekansları

üzerinden uzuvlar belirlenerek bu uzuvların üzerlerindeki doku modeli oluşturulmuştur [23]. Hareket izleme sisteminden ve doku modelinden alınan veriler kullanılarak piksellerinin pozisyonu ile silindir üzerinde olan pozisyonlar arasında bir eşleme yapılmış ve uzuvlar modellenmiştir.

Yukarıda bahsedilen yöntemlerin hepsi temelde belirli bir soyutlama seviyesi kullanarak objenin pozunu görüntü kareleri arasında karşılaştırmış ve hareketi hesaplamıştır. Aynı zamanda bu yöntemin tersi kullanılarak da üç boyutlu hareket modeli oluşturulabilmektedir. Yamamoto ve Koshikawa tarafından yapılan çalışmada [24] yürüyen bir insanın çeşitli vücut uzuvlarının optik akışı ölçülmüştür. Daha sonra bu ölçüm Jacobian matrisinde kullanılmış ve ters kinematik çözümü yapılarak model güncellenmiştir. Diğer bir çalışmada insan bacağını modellemek için 2 adet düzlemsel parça kullanılmıştır. Her bir düzlemsel parçanın hareketi 8 adet parametre ile modellenmiştir [25]. Düzlemsel parça üzerinde parlaklığın sabit olduğu varsayımı yapılarak bu parça üzerinde bulunan bütün piksellere optik akış kısıtı yapılmış ve parametreler kestirilmiştir. Daha sonra bulunan parametrelerden düzlemsel parçaların hareket eşitliği oluşturulmuştur ve hareket takibi yapılmıştır. Bu çalışma literatürdeki diğer yöntemlerden farklı olarak hareket algılama için herhangi bir üç boyutlu insan modeli kullanmamış, optik akışı direk olarak parametrize edilmiş düzlemsel parçalar üzerinden hesaplamıştır. Hareket algılama düzlemsel parçaların ayrı ayrı hareket eşitliklerinin çözülmesi ile yapılmıştır.

2.2 Kızılötesi Sensör ve Uçuş Zamanı Kameralar Kullanılarak Yapılan Görüntü İşleme Çalışmaları

ToF kameralar ve kızılötesi kameralar ile yapılan çalışmalarda renkli kameralar ile yapılan çalışmalardan farklı olarak eklemlerin üç boyutlu pozisyonları kolaylıkla algılanabilmektedir. Bu kameralar derinlik haritası üzerinden çalışan algoritmalar ile üç boyutlu pozisyon takibi yapılmasına olanak sağlamaktadırlar. Kameradan yayılan kızılötesi ışının bir engele çarpıp tekrar kamera tarafından algılanması ile havada geçirdiği süre hesaplanır. Bu sayede karşıdaki engelin kameraya olan uzaklığı bulunmuş olur. Bir nevi sonar sistemi gibi çalışan bu sistemde kameralar önlerindeki sahnedeki objelerin derinliklerini algılayabilmektedirler.

Bu kameralardan biri olan ve Microsoft tarafından geliştirilen Kinect ile birlikte bu alandaki çalışmalar. Kinect üzerinde 640x480 çözünürlüğe ve 30Hz frekansa sahip bir RGB kamera ve ToF kamera bulunmaktadır. Bu sebeple literatürde RGB-D kamera olarak da adlandırılmaktadır. Üzerinde bulunan değişik türdeki bu kameralar sayesinde senkronize bir şekilde renkli görüntü ve derinlik görüntüsü verebilmektedir. Aynı zamanda Microsoft tarafından sağlanan uygulama geliştirme kiti ile Kinect geliştirme kütüphanesi ve örnek uygulamalar da araştırmacılar ile paylaşılmıştır [26].

Literatürde Kinect kullanılarak yapılan çalışmalar bir derleme [27] makalesinde dört ana başlık altında incelenmiştir. Bu alanlardan bir obje tespiti ve takibidir. Yazarlar inceledikleri makaleler dışında çalışmalarının sonuçlarını da bu derleme makalesinde göstermişlerdir. Bu çalışmada Kinect tarafından verilen derinlik görüntüsü ve RGB görüntüsü Gaussian karışım modeli tabanlı bir arka plan çıkarımı algoritmasına girdi olarak verilmiştir. Bu iki farklı görüntü tipinin güvenilirliğini ölçmek için farklı durumlarda testler yapılmıştır. İlk durumda ön plan ve arka planın birbirine benzediği bir sahne kullanılmış, diğer durumda da sahnenin ışıklandırılması aniden değiştirilmiştir. Şekil 2.5'te durumların RGB ve derinlik görüntüleri gösterilmiştir. Yapılan çalışmada derinlik kamerasından segmente edilen ön plan objesinin RGB kamerasından segmente edilene göre çok daha kaliteli olduğu gözlemlenmiştir.

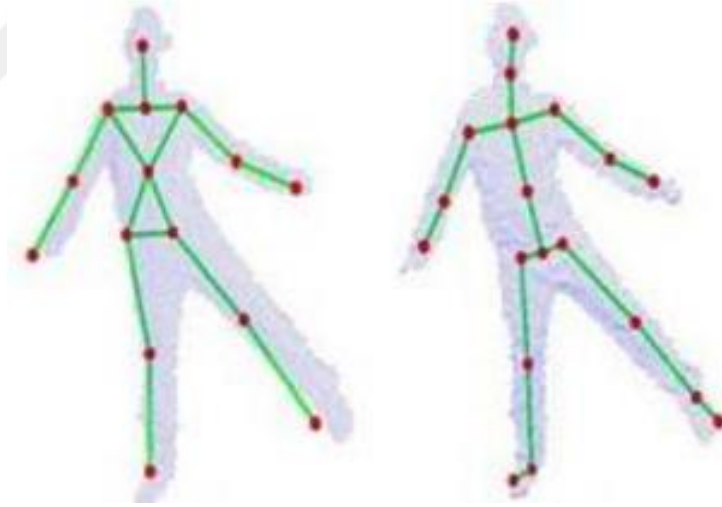


Şekil 2.5- RGB Görüntü, RGB görüntüsünden elde edilen ön plan maskesi ve derinlik görüntüsünden elde edilen ön plan maskesi [27].

Kinect kullanılarak yapılan çalışmaların ortak noktası derinlik görüntüsünden analiz yapılmasıdır. Bu çalışmalardan birinde iç ortamlarda insan algılama için derinlik görüntüsü kullanılmıştır [28]. Derinlik görüntüsünden alınan veri gürültü azaltımı yapıldıktan sonra, iki aşamalı kafa tespiti algoritması uygulanmıştır. Öncelikle derinlik görüntüsüne gömülü olarak gelen sınır bilgisinden içerisinde insan bulundurması olası bölgeler tespit edilmiştir. Buradaki algoritma iki boyutlu oluk tarama işlemi yapmaktadır. Daha sonra bu bölgelerde üç boyutlu kafa modeli kullanılarak inceleme yapılmış ve tahmin son haline getirilmiştir. Ardından belirlenen bölgelerde bütün vücudu bulmak için segmentasyon yöntemi kullanılmış ve bütün vücudun çevriti tespit edilmiştir. Başka bir çalışmada ise alınan derinlik görüntüsü karelerinden yaşlılar için düşme tespiti yapılmıştır [29]. Birden çok arka plan derinlik görüntüsü eğitim amaçlı kullanılarak arka plan maskesi oluşturulmuştur. Daha sonra bu maske kullanılarak arka plan çıkarımı yapılarak ön planda kalan görüntüde insan segmentasyonu yapılarak model çıkarılmıştır. Aynı zamanda derinlik görüntüsünden yer düzlemi de tespit edilerek daha önce elde edilen insan modelinin kütle merkezinin bu düzleme uzaklığı gözlenmiş ve bu şekilde düşme tespiti yapılmıştır.

Kinect kullanılarak geliştirilen diğer bir çalışma alanı da insan aktivitesi analizidir. Bu alanda yapılan çalışmalar da kendi içerisinde ikiye ayrılabilir. İlk alanda araştırmacıların yaptığı iskelet takibi ve poz kestirimi çalışmaları gösterilebilir. Özellikle bu alanda Kinect SDK'nın da sunduğu gelişmiş iskelet takibi analizi ile çalışmalar hızlanmıştır. Kinect SDK tarafından sunulan iskelet takibi algoritmasının temel kısımları araştırma ekibi tarafından yayınlanmıştır [30]. Bu çalışmada yayınlanan algoritmanın temelinde piksel bazında vücut bölgesi sınıflandırması bulunmaktadır. Daha sonra bölgesel kütle merkezleri bulunarak eklem pozisyonlarının bulunması için tahmin yürütülmesi yapılmaktadır. Bu algoritmanın geliştirilmesi sırasında değişik şekil ve büyüklüklerde gerçekçi sentetik derinlik görüntüleri kullanılarak rasgeleli karar ağacı sınıflandırıcısı eğitilmiştir. Bu algoritma herhangi bir zamansal veriye ihtiyaç duymadan her görüntü karesi üzerinde çalışarak anlık karar verisi üretmektedir. Bu çalışma literatüre poz kestirimi gibi konulara veri sağlayabilecek bir çıktı sağlayarak katkıda bulunmuştur. Bu çalışma sonucunda elde edilen sonuçlar, daha önce elde edilen sonuçlara göre en az bir basamaksal büyüklük

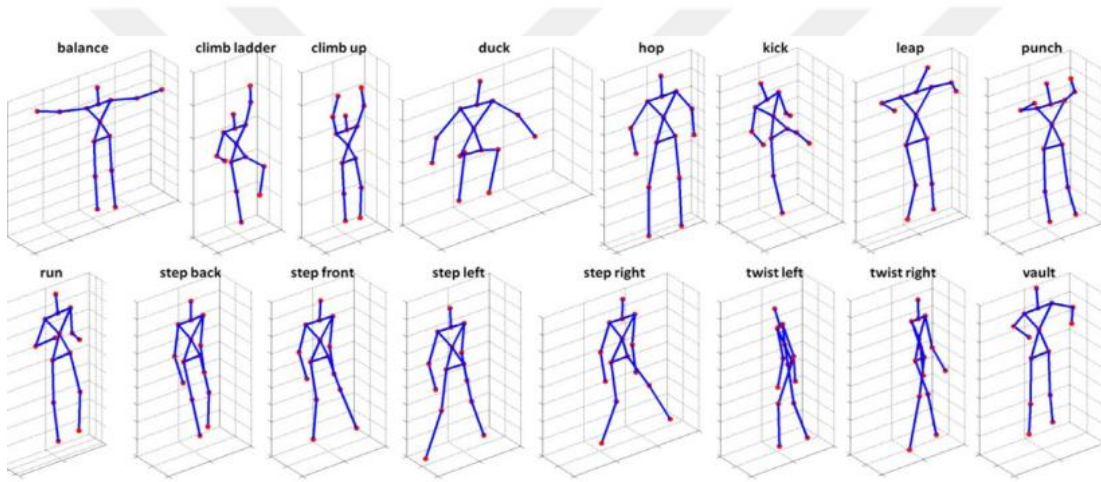
göstermektedir. Elde edilen sonuçlar ile yayınlanmayan, patentli bir algoritma kullanılarak varsayılan eklem pozisyonlarından iskelet modeli oluşturulmuştur. Oluşturulan bu model üzerinde zamansal ve kinematik sınırlamalar kullanılarak düzgün bir iskelet takibi çıktısı vermesi sağlanmıştır. Diğer çalışmalarda ise tek derinlik görüntüsünden elde edilen poz kestirimi ile doğruluğu arttırılmaya çalışılmıştır [31] [32]. Bu çalışmalarda geliştirilen poz kestirimi algoritmaları derinlik görüntüsünden çıkarılan nokta bulutu girdisini kullanarak daha önceden oluşturulmuş hareket veri setinden en iyi eşlemeyi bulmaya çalışmaktadır. Daha sonra en iyi eşleşen poz verisi, derinlik haritası üzerinden alınan vücut görüntüsü üzerine yerleştirilerek tekrar düzenleme yapılmış ve daha doğru bir iskelet modeli elde edilmiştir. [31]'da poz düzenleme Coherent Drift Point (CPD) algoritması kullanılarak elde edilmiştir. [32] 'de ise poz düzenleme yöntemi zamansal hareket tutarlılığı ile sistematik eğilimin beraber optimize edilmesi ile sağlanmıştır. Bu çalışmalar ile elde edilen poz kestirimi iyileştirmesi Şekil 2.6'da gösterilmiştir.



Şekil 2.6- Poz düzenleme algoritması kullanılmadan yapılan [31] ve poz düzenleme algoritması kullanılarak iyileştirilen poz kestirimi [32].

İnsan aktivite analizinin diğer bir alt başlığı da uygulamanın semantiğine uygun olarak aktivite algılamaktır. Bu alanda yapılan çalışmaların odak noktası çıkarılan iskelet modeli, derinlik haritası ve eklem pozisyonu gibi verilerden ayırt edilebilir özellikler çıkararak aktivite algılaması yapabilmektir. Örneğin [33] ve [34] 'da

yapılan çalışmalarda iskelet verisinden takip edilen eklemlerin uzaysal ve zamansal farkları incelenerek insan hareketleri saptanmaya çalışılmıştır. Hareketlerin algılanması, hareket esnasında her bir eklemin ayrı ayrı hareket yörüngesi ve eklemlerin birbirlerine göre konfigürasyonlarının karşılaştırılması ile elde edilmiştir. Eklemlerin uzaysal ve zamansal aralıkta hareketleri, belirli hareket düzenlerinde görüntü kareleri boyunca incelenerek oluşturulan özellik vektörleri ile eklem dinamikleri modellenmiştir. Daha sonra temel prensip analizi (Principal Component Analysis- PCA) uygulanarak EigenJoint'ler bulunmuştur. Bulunan EigenJoint'ler üzerinde Naive-Bayes-Nearest-Neighbor (NBNN) sınıflayıcısı uygulanarak hareketler kategorize edilmiştir. Yapılan bu kategorizasyon sonucunda oluşan hareket sınıflarının bir örneği Şekil 2.7 'de gösterilmiştir.

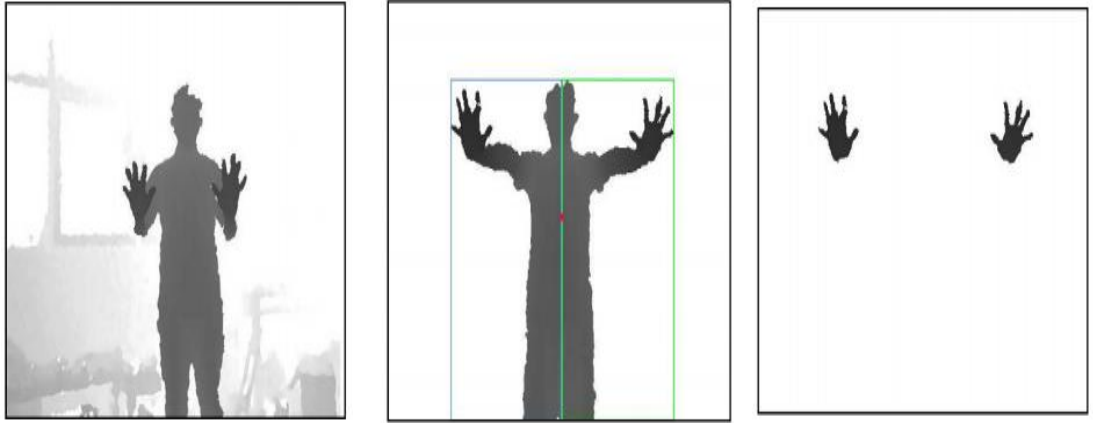


Şekil 2.7- Eigen Joint'lerin NBNN kullanılarak sınıflandırılması ile hareket analizi yapılması [34].

Kinect kullanılarak yapılan bir diğer görüntü işleme analizi de el hareketlerini algılama alanında yapılmaktadır. Özellikle bu alanda yapılan çalışmalar ile kullanıcıların makinelerle daha iyi etkileşime geçmeleri amaçlanmaktadır. Bu alanda yapılan çalışmalar el algılanması, el pozunun kestirimi ve hareketlerin sınıflandırılması olarak sınıflandırılabilir.

Vücut eklemlerinin algılanması çalışmalarında olduğu gibi el algılanması da derinlik görüntüleri analiz edilerek yapılabilir. Örneğin yapılan bir çalışmada, el algılanması derinlik kümelemesi problemi olarak ele alınmıştır [35]. Antropometrik analiz yapılarak insan vücudunun derinlik görüntüsünden segmente edilmesinden sonra, vücudun kütle merkezi hesaplanmıştır ve bu hesaba göre vücudun sağ ve sol

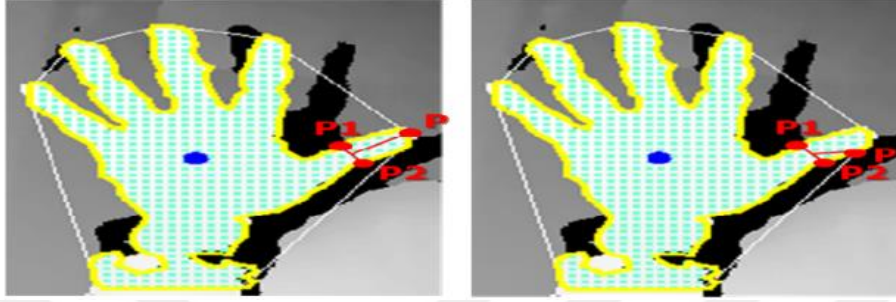
kısımları ayrılmıştır. Her bölgenin kendi içerisinde bir el görüntüsüne sahip olacağı varsayılarak, el lokasyonları poz analizi yapılarak bulunmuştur. Bulunan el pozisyonları orjinal derinlik görüntüsünden daha önceden hesaplanmış değerler ile eşik değer operasyonu yapılarak ayrılmıştır. Bu işlemler yapılırken takip edilen objenin derinlik olarak kameraya en yakın obje olduğu ve objenin ellerinin de vücudunun ön tarafında olduğu varsayımları yapılmıştır. Bu sayede eşik değer operasyonu kolayca uygulanarak el pozisyonları bulunmuştur. Bahsedilen varsayımlara sahip bir derinlik görüntüsü Şekil 2.8’de gösterilmiştir.



Şekil 2.8- El pozisyonlarının derinlik görüntüsünden algılanması [35].

Başka bir çalışmada da eller bulunduktan sonra parmak uçlarının pozisyonları bulunarak kullanıcı etkileşimli uygulamalarda kullanılmıştır [36]. Bu çalışmada da ilk aşama olarak derinlik görüntüsünden eller segmente edilmiş, görüntü temizlenerek sadece el bölgeleri bırakılmıştır. Daha sonra k-ortalama kümeleme algoritması kullanılarak el noktaları karelerden ayrılmıştır. Daha sonra kümelenen veri üzerinde modifiye edilmiş Moor Neighbor Tracing algoritması ile elin çevriti belirlenmiş, Graham Scan algoritması kullanılarak da dışbükey örtü (convex hull) hesaplanmıştır. Daha sonra bulunan dışbükey örtüdeki her bir nokta, parmak ucu için aday nokta olarak sayılmış, bu noktaya en yakın olan nokta el çevriti üzerinde bulunarak P noktası olarak işaretlenmiştir. Bulunan her P noktası için el çevriti analiz edilerek ters yönde iki adet destek noktası, P1 ve P2, bulunmuştur. Bulunan bu üç noktanın, P, P1 ve P2, aynı doğrultuda olup olmadığı hesaplanmış, aynı doğrultuda oldukları durumda P noktası parmak ucu aday noktası olmaktan çıkarılmıştır. Aynı doğrultuda olmadıkları durumda ise P noktasının P1 ve P2 noktalarından geçen

doğruya dik uzaklığı hesaplanarak daha önce deneysel olarak hesaplanan parmak uzunluğu değerine yakın olup olmadığına bakılmıştır. Bu noktadan sonra eğer hesaplanan değer deneysel olarak belirlenen değer ile örtüşüyorsa P noktası parmak ucu noktası olarak kabul edilmiştir. İzlenen bu yaklaşım ile işaretlenen baş parmağı gösteren derinlik görüntüsü Şekil 2.9 'da gösterilmiştir.



Şekil 2.9- P noktası parmak ucu olarak seçilmiş ve P noktası parmak ucu olarak işaretlenmemiş el haritaları [36].

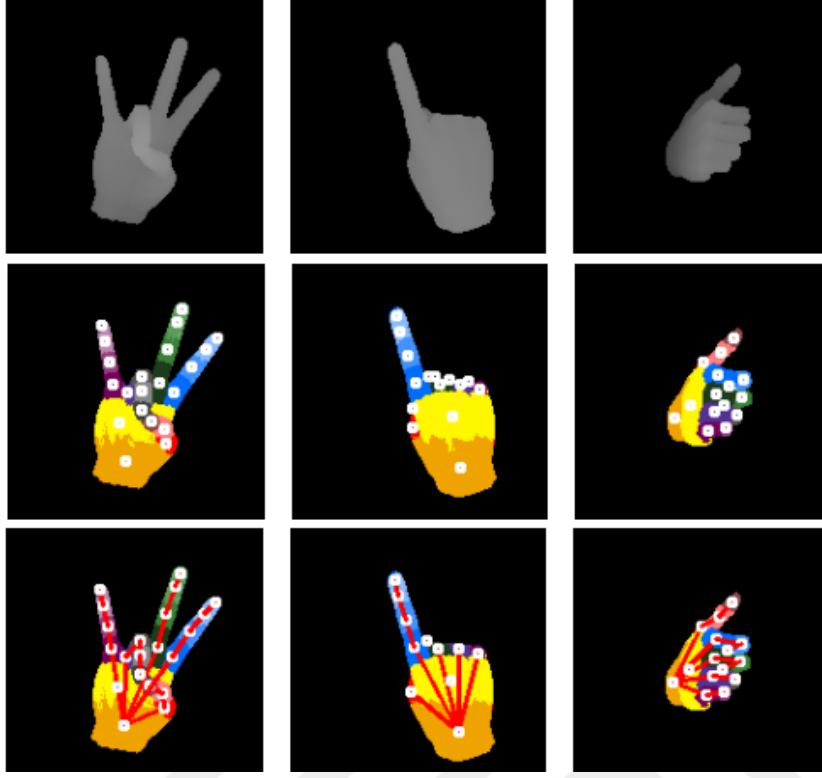
Bu aşamadan sonra sistem parmak sayılarını sayarak baş ve işaret parmaklarını işaretlemiştir. Baş parmak algılaması için algoritma parmağın kameraya olan uzaklığını ve şekil eşleşme yöntemlerini kullanmıştır. Daha sonra baş parmağa olan uzaklıklarına göre sistem diğer bulunan parmakları isimlendirmiştir. Bu aşamadan sonra parmak uçları her iki elde de takip edilmiş ve bunların kullanıldığı etkileşim uygulamaları geliştirilmiştir. Bu uygulamalardan örnekler Şekil 2.10'da gösterilmiştir.



Şekil 2.10- Parmak ucu hareketleri takip edilerek geliştirilen uygulamalar [36].

Bu alanda yapılan diğer çalışmalarda da arařtırmacılar benzer yaklařımlar izlemiřlerdir. [37] 'de Liang ve alıřma grubu el tespiti yapmak iin derinlik grntleri zerinde morfolojik kısıtlar uygulayarak kmeleme algoritması alıřtırmıř ve el evritini segmente etmiřtir. Daha sonra ise segmente edilen el evriti zerinde mesafe dnřm yapılarak avu ii ve elin orta noktası hesaplanmıřtır. Bir diđer alıřmada ise nceki alıřmalara gre farklı bir yntem izlenerek el tespiti yapmak yerine derinlik grnts incelenmiř ve boru benzeri veya ince ulu ğeler tespit edilmeye alıřılmıřtır [38]. Daha sonra bu ğeler parmak ve avu ii adayları olarak tespit edilmiř ve uzaysal koordinatları beraber incelenerek belirli bir řekil uyumu oluřturup oluřturmadıkları tespit edilmiřtir. Belirli bir řekil uyumu sađlayan adaylar avu ii ve parmaklar olarak tanımlanmıřlardır.

Kinect ile yapılan iskelet takibi alıřmaları gibi el iskeletini ıkarma arařtırmaları da literatrde alıřılmıřtır. Burada ama elin iskeletini ıkararak vcutta yapıldıđı gibi poz kestirimi yaparak hareket takibi yapabilmektir. Bu alanda yapılan alıřmalardan birinde [39] arařtırmacılar [30]'te kullanılan yntemi benimseyerek aynı iřlemi el iskeletini oluřturmak iin kullanmıřlardır. Sentetik el grntleri yaratılarak her bir el eklemi parası etiketlenen blmlerin ortasına gelecek řekilde grntdeki paralar etiketlenmiřtir. Daha sonra oluřan bu veri seti ile rasgeleleřtirilmiř karar ađacı eđitilerek eř zamanlı olarak alınan derinlik grntlerinde sınıflandırma yapmak iin kullanılmıřtır. Sınıflandırma yapılırken grntde bulunan her bir piksel ayrı olarak sınıflandırılmıř ve bir el parasına atanmıřtır. Son olarak da eklem merkezlerinin hesaplanması iin ortalama kaydırma algoritması kullanılmıřtır. alıřma ile elde edilen el pozu takip sistemi Amerikan iřaret dili rakamlarını algılamak iin kullanılmıřtır. Daha sonraki alıřmalarında ise ok katmanlı bir RDF yapısı kullanarak bir nceki alıřmalarındaki algoritmayı geliřtirmiřlerdir [40]. Bu alıřmada oluřturulan el řekli sınıflandırıcısı el pozu kestirimi yapan kısım ile derinlik grnts arasında bir ara katman grevini grmřtr. Bu alıřmalar sonucunda elde edilen ASL rakam algılama sonuları řekil 2.11'de gsterilmiřtir.



Şekil 2.11- Derinlik görüntüsünden algılanan el (üst), tahmin edilen eklem pozisyonları (orta), oluşturulan el iskeleti (alt) [39].

Kinect ile derinlik görüntüsü işlenerek yapılan çalışmalar ile daha önceden RGB görüntü üzerinden yapılan çalışmalar karşılaştırıldığında, derinlik görüntüsü işlenerek yapılan analizlerin RGB görüntü işlenerek yapılan analizlere göre daha kolay ve hızlı bir şekilde sonuç verebildiğini göstermiştir. Elde edilen bu sonuçlar birçok alanda Kinect ile yapılan çalışmaları hızlandırmıştır.

2.3 Kinect ile Robotik Alanında Yapılan Görüntü İşleme Çalışmaları

Görüntü işleme insan-makine etkileşimi konusunda özellikle robotik alanında yoğunlukla kullanılmaktadır. Bu çalışmalarda uzaktan kontrol işlemleri için Kinect ve karşısında bulunan operatör merkez birim olarak kabul edilmiş, kontrol edilen robot ise uydu birim olarak kabul edilerek merkez-uydu eşleşmesi ile uzaktan kontrol üzerine gidilmiştir. Özellikle Kinect yazılım geliştirme kitinin geliştiricilere açılması ve bu kit ile birlikte gelen iskelet takibi algoritmasının yayınlanması ile birlikte bu

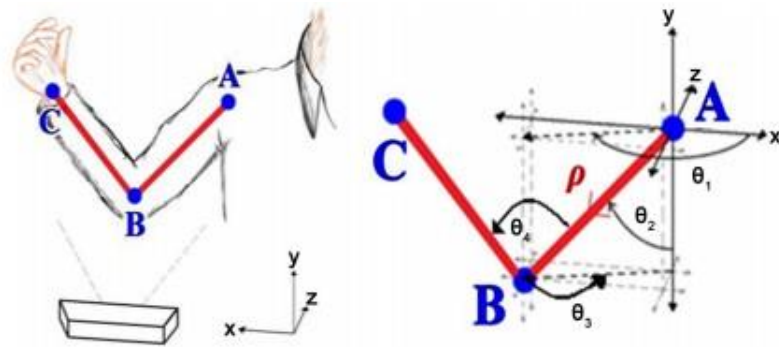
alanda yapılan çalışmalar artmıştır. Bu bölümde bu çalışmalardan örnekler anlatılmaktadır.

Robotik alanında yapılan çalışmalar humanoid robotlar ve endüstriyel robotlar başlıkları altında incelenebilir. Humanoid robotlar insan vücut yapısına yakın oldukları için eklemler ve iskelet pozisyonundan alınan veriler çoğu zaman herhangi bir değişiklik yapmadan robota iletilebilir. Ancak endüstriyel robotların kinematik zincirleri insan iskelet sisteminden farklı olduğu için bu çalışmalarda insan-robot etkileşimini sağlamak için ara katmanlara ihtiyaç duyulabilmektedir. Bu bölümde bu iki alanda da yapılan çalışmalar incelenecektir.

2.3.1 Kinect Kullanılarak Humanoid Robotlar Üzerinde Yapılan Çalışmalar

Humanoid robotlar eklem yapısı olarak insan benzeri kinematik zincir içerirler. Bu robotların eklemleri insan eklemlerine benzer açı aralıklarında çalışmaktadır. Humanoid robotlar ile yapılan çalışmalarda ortak nokta iskelet takibi kullanılarak eklem pozisyonları ve uzuv vektörlerinin çıkarılmasıdır.

Örneğin yapılan bir çalışmada araştırmacılar insan üst uzuvlarını takip ederek eklem açılarını hesaplamış ve bunları doğrudan aynı eklem yapısına sahip bir robota göndermişlerdir [41]. Kinect tarafından takip edilen insan operatörün kolunun eklem pozisyonları ve kol kemiklerine denk gelen uzuv vektörleri oryantasyonları kullanılarak ters kinematik çözümü yapılmış ve eklem açıları hesaplanmıştır. Hesaplamalarda omuz üç eklem, dirsek ise bir eklem ile temsil edilmiştir. Açıların gönderildiği robot kolu da bu dört serbestlik derecesinden oluşmaktadır. Şekil 2.12’de insan kolu üzerinde hesaplanan açılar gösterilmiştir.



Şekil 2.12- İnsan kolu üzerinden hesaplanan açılar [41].

Açılar aşağıdaki gösterilen eşitlikler kullanılarak hesaplanmıştır.

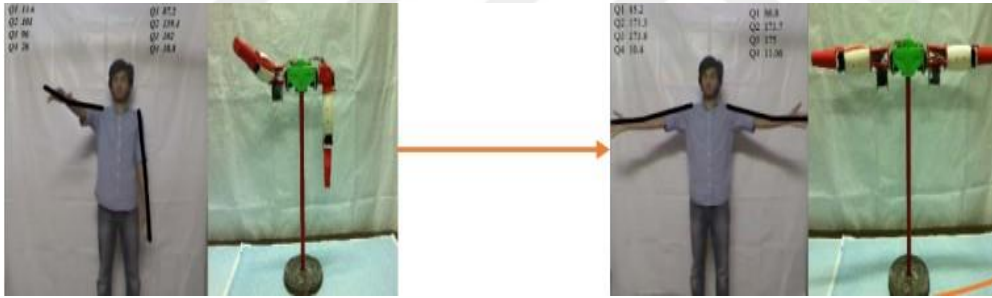
$$\theta_1 = \sin^{-1} \left(\frac{AB_x}{\sqrt{AB_x^2 + AB_y^2 + AB_z^2}} \right) \quad (2.1)$$

$$\theta_2 = \sin^{-1} \left(\frac{AB_y}{\sqrt{AB_x^2 + AB_y^2 + AB_z^2}} \right) \quad (2.2)$$

$$\theta_3 = \sin^{-1} \left(\frac{BC_x}{\sqrt{BC_x^2 + BC_y^2 + BC_z^2}} \right) \quad (2.3)$$

$$\theta_4 = \cos^{-1}(AC^2 - AB^2 - BC^2) \quad (2.4)$$

Gösterilen açılar ters kinematik eşitlikler ile çözülerek robotun insan kolunda denk gelen eklemlerine gönderilmiştir. Robot ile operatörün pozisyonları Şekil 2.13'te gösterilmiştir.



Şekil 2.13- Direk besleme ile gönderilen açılar sonrası operatör ve robotun pozisyonları [41].

Bir diğer çalışmada da humanoid bir robot olan NAO [42], Kinect aracılığı ile kontrol edilmiştir. Bu çalışmada robotun kontrolünü sağlamak için üç farklı yöntem kullanılmıştır [43]. Kullanılan yöntemlerden ilki direkt açı besleme yöntemidir. Kinect aracılığı ile operatörün eklem koordinatları ve uzuv vektörlerinden hesaplanan eklem açıları robotun karşılık gelen eklem açılarına direkt olarak gönderilmiştir. Bu yöntemlerde kullanılmak üzere ise NAO robotun düz kinematik eşitlikleri modifiye Denavit Hartenberg [44] yöntemi kullanılarak hesaplanmış ve uç işlemcinin uzaysal koordinatları eklem açıları ve uzuv uzunlukları açısından ifade

edilmiştir. Bu eşitliklerde kullanılan l_1, l_2 robotun üst kol ve alt kol boyutlarını $\theta_1 \dots \theta_4$ ise sırasıyla kolun eklem açılarını belirtmektedir.

$$p_x = -l_2(\sin\theta_4(\sin\theta_1\sin\theta_3 + \cos\theta_1\cos\theta_3\sin\theta_2 - \cos\theta_1\cos\theta_4\cos\theta_2) + l_1\cos\theta_1\cos\theta_2) \quad (2.4)$$

$$p_y = -l_2(\cos\theta_4\sin\theta_2 + \cos\theta_3\sin\theta_4\cos\theta_2) + l_1\sin\theta_2 \quad (2.5)$$

$$p_z = -l_2(\sin\theta_4(\cos\theta_1\sin\theta_3 - \cos\theta_3\sin\theta_2\sin\theta_1) + \cos\theta_4\sin\theta_1\cos\theta_2) - l_1\sin\theta_1\cos\theta_2 \quad (2.6)$$

İkinci yöntemde ise özyineli Jacobian metodu kullanılarak ters kinematik eşitliği çözülmüştür. Uç işlemcinin uzaysal koordinatlarındaki değişimini, eklem açılarının değişimi ile ilişkilendirilerek Jacobian matrisi oluşturulmuştur.

$$\vec{X} = f(\theta) \quad (2.7)$$

$$d\vec{X} = J(\theta)d\vec{\theta} \quad (2.8)$$

$$dx_i = \sum_{j=1}^n \frac{\partial f_i}{\partial \theta_j} d\theta_j, i = 1 \dots m \quad (2.9)$$

$$j_{ij} = \frac{\partial f_i}{\partial \theta_j} \quad (2.10)$$

(2.7)'de verilen eşitliğin sağ tarafındaki fonksiyon NAO robotun düz kinematik eşitliklerini veren fonksiyon olduğu varsayılmıştır. Bu durumda (2.9)'da uç işlemcinin uzaysal koordinatlarındaki değişim miktarı ifade edilmiştir. Jacobian matrisinin her bir elemanın değeri ise (2.10) kullanılarak hesaplanmıştır. Her adımda Jacobian matrisinin sapması hesaplanmış ve hata belirlenen limite gelinceye kadar uç işlemcinin hesaplanan uzaysal koordinatı azaltılarak hesaplamalar tekrarlanmıştır. Uç işlemcinin yeni uzaysal koordinatlarını elde etmek için eklem açılarında olması gereken değişim (2.11) ve (2.12) eşitlikleri kullanılarak hesaplanmıştır.

$$\Delta\theta = J^+\Delta X \quad (2.11)$$

$$\Delta X = X_{Hedef} - X_{Mevcut} \quad (2.12)$$

Bu prosedür (2.12)'de hesaplanan hata kabul edilebilir seviyeye gelene kadar özyineli olarak tekrarlanarak eklem açıları bulunmuştur.

Üçüncü yöntemde ise MATLAB Fuzzy Logic Toolbox kullanılarak Yapay Bulanık Sinir Ağı (Artificial Neuro-Fuzzy Inference Systems- ANFIS) eğitilmiştir [45]. Bu sinir ağının eğitilmesi için robotun düz kinematik eşitlikleri kullanılmıştır. Daha sonra eğitilen ANFIS'e operatörün bileğinin omzuna göre olan uzaysal koordinatları verilerek eklem açıları alınmış ve robota gönderilmiştir.

Çalışmada kullanılan yöntemler performans olarak farklı kriterler altında değerlendirilmiştir. Direk açı besleme metodu Kinect'ten gelen olası gürültülü veriler göz önünde bulundurulduğunda diğer iki metoda göre daha az güvenilir bulunmuştur. Ters kinematik eşitliklerinin özyineleme kullanılarak Jacobian matrisi ile çözülme yöntemi ise çok fazla işlem gücü gerektirdiği için eş zamanlı bir işlem için uygun olmadığı belirtilmiştir. ANFIS sistemini kullanan ters kinematik yöntemi ise bu yöntemler arasında eş zamanlı olarak kullanılması açısından en uygun yöntem olarak seçilmiştir. Ancak bu yöntemin kullanılabilmesi için de yapay bulanık sinir ağının eğitiminin uzun sürebildiği belirtilmiştir.

Bir diğer çalışmada araştırmacılar daha önceki çalışmalardan farklı olarak benzerlik çıkarımı yapabilmek için sadece eklemlerin uzaysal koordinatlarını takip etmek yerine eklemler arasında bulunan uzuvların normalize vektörlerini de kullanmışlardır [46].

$$B_S = \sum_{i=1}^n w_i \left\| \frac{1}{\|B_{H_i}\|} B_{H_i} - \frac{1}{\|B_{V_i(\theta)}\|} B_{V_i(\theta)} \right\|^2 \quad (2.13)$$

(2.13)'te geçen B_{H_i} eklemler arasında bulunan insan uzuvuna ait vektör, B_{V_i} de bu uzuvun robotta karşılığına gelen uzuvu belirtmektedir. B_{H_i} Kinect'ten alınan eklem verilerinden elde edilirken B_{V_i} robotun düz kinematik eşitliklerinden elde edilmiştir.

Çalışmada yapılan katkı (2.13)'te belirtilen hata fonksiyonunun minimize edilerek ters kinematik çözümünün optimize edilmesi olarak gösterilmiştir. Bunun için yapılan adımlar sağ kol modelinden örneklendirilerek gösterilmiştir. Kinect iskelet takibi algoritmasında sağ kol üç eklem ile modellenmiştir. Bu üç eklem arasında iki uzuv vektörü bulunmaktadır. Elde edilen uzuv vektörleri ve bu vektörlere denk gelen robot uzvu vektörleri hesaplanarak (2.13)'teki hata fonksiyonu çözülmüştür. Buradaki hata optimizasyonu problemi bir minimizasyon problemi olarak ele alınmış

ve eklem açıları özyineleme ile Levenberg-Marquard algoritması kullanılarak çözülmüştür [47]. Bu optimizasyon probleminin çözümü sonucunda elde edilen açılar robotun eklemlerine gönderilmiştir.

Yapılan diğer çalışmalarda da benzer yaklaşımlar benimsenmiştir. Örneğin [48]'de Kinect'ten alınan eklem koordinatları arasında kalan uzuv vektörleri bulunmuş ve ters kinematik çözümleri bu vektörler üzerinden yapılmıştır.

$$\theta_1 = \sin^{-1} \left(\frac{\overline{L_{sey}}}{l_2} \right) \quad (2.14)$$

$$\theta_2 = \cos^{-1} \left(\frac{\overline{L_{sex}}}{l_1} \right) \quad (2.15)$$

$$\theta_3 = \cos^{-1} \left(\frac{\overline{L_{se}L_{eh}}}{\|L_{se}L_{eh}\|} \right) \quad (2.16)$$

Bu eşitliklerde θ_1 ve θ_2 omuz açıları θ_3 ise dirsek açısını ifade etmektedir. Ayrıca L_{se} omuz-dirsek vektörünü, L_{eh} dirsek-el vektörünü, l_1 ve l_2 de omuz vektörünün eksenler üzerindeki iz düşüm uzunluğunu belirtmektedir. Daha sonra hesaplanan açılar robotun kontrolcüsüne gönderilmiştir.

Benzer bir çalışmada [49] yine uzuv vektörleri üzerinden geometrik eşitlikler kullanılarak açılar bulunmuştur. Ancak bu çalışmada farklı olarak robot eklem açıları ile operatör eklem açıları arasında bir ölçeklendirme faktörü hesaplanmış ve her bir eklem için robota gönderilen açılar, robotun çalışma alanına göre ölçeklendirilerek hesaplanmıştır. Buradaki ölçeklendirme faktörü her bir eklem için robotun belirtilen eklem açısının alabileceği değerler ile insanda bulunan aynı eklem alabileceği açı değerleri ile ilişkilendirilmesiyle hesaplanmıştır. Ayrıca robotun eklemlerini hareket ettiren motorların aralarındaki uzaklığın sebep olduğu eklem ofsetleri de açılara eklenmiştir.

Humanoid robotlar üzerinde yapılan çalışmalardan bir diğerinde [50] ise yine Denavit-Hartenberg parametreleri ile hesaplanan düz kinematik eşitlikleri ve bunların ters kinematik çözümleri ile direkt açı gönderme yöntemleri denenmiştir. Bu çalışmada da ters kinematik hesaplamaların çok karmaşık olduğundan ve çok fazla hesaplama zamanı aldığından bahsedilmiştir. Bu nedenle bu yöntemi kullanmak yerine direk açı hesaplama ve gönderme yöntemi kullanılmıştır. Bu yöntemde önceki

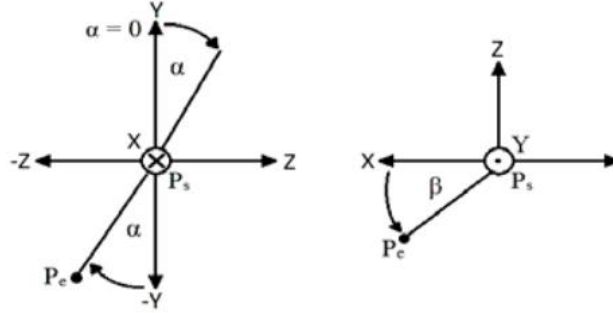
çalışmalarda olduğu gibi açılar robotun taklit etmesinin istendiği operatörün eklem koordinatları ve uzuv vektörlerinden hesaplanmaktadır. Diğer çalışmalardan farklı açı eşlemek için robot ve insan eklem açıları arasında bir eşleştirme tablosu oluşturulmuştur. Bu tablonun oluşturulmasının nedeni kullanılan humanoid robot ile insan iskelet sisteminin fiziksel farklılıkları olarak gösterilmiştir. Kullanılan NAO robotun dirseğini iki eklem temsil ederken insan iskelet sisteminde dirsek eklemi tektir. Ayrıca insan omuzu üç eksene sahipken kullanılan robotun omuz eklemi sadece iki serbestlik derecesine sahip bir eklem yapısından oluşmaktadır. Araştırmacılar direk açı gönderme işlemi sırasında bu sorunun üstesinden gelmek için bu çalışmada oluşturulan açı eşleştirme tablosunu kullanmışlardır.

Bir diğer çalışma ise Zeno Robot [51] üzerine yapılmış ve otizm spektrum bozukluğu terapisinde kullanılmıştır. Burada amaçlanan çalışma Zeno'nun terapi gören hasta ile iletişime geçerek hareketlerini hastaya tekrarlatmasını sağlamaktır. Zeno'nun gerçekleştirdiği hareketler ise terapist tarafından gerçekleştirilen hareketlerin Kinect aracılığıyla robota aktarılması ile oluşmaktadır [52]. Sistemin çalışma prensibi daha önceki çalışmalara benzerlik göstermektedir. Terapistin omuz, dirsek ve el eklemlerinin koordinatları Kinect tarafından algılanarak ters kinematik algoritmasına gönderilmektedir. Diğer çalışmalardan farklı olarak alınan koordinatlar gönderilmeden önce 25 Hz örnekleme hızı ve 2 Hz kesme frekansı olan alçak geçiren filtreden geçirilmiştir. Bu işlem sayesinde Kinect'ten gelen veriler üzerindeki gürültü filtrelenmiştir. Ters kinematik algoritması ise eklem koordinatları, uzuv vektörleri ve bu vektörlerin oryantasyonlarını kullanarak robota gönderilecek açıları belirlemektedir. Zeno kol için dört, boyun için üç, üst gövde için de bir serbestlik derecesine sahip eklemlerden oluşmaktadır. Bu açılardan omuzlarda bulunan açıların hesaplanmasında Euler açıları notasyonu kullanılmıştır.

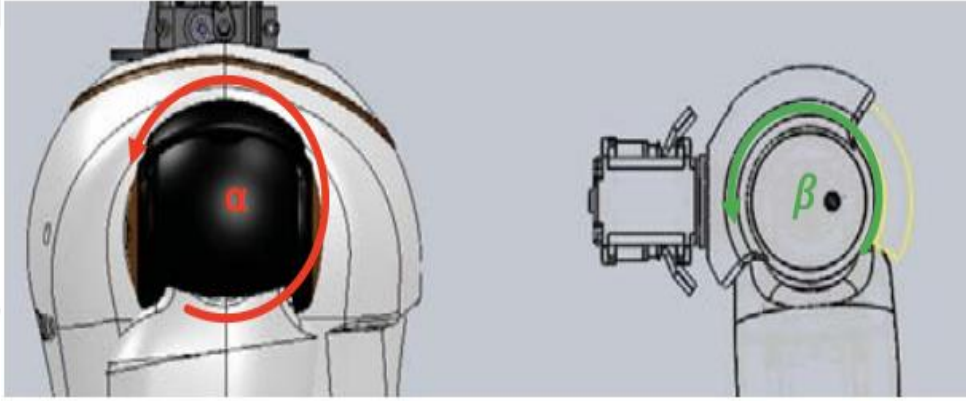
$$\alpha = \text{Atan2} \left(\frac{-z_e}{-y_e} \right) \quad (2.17)$$

$$\beta = \cos^{-1} \left(\frac{x_e}{\sqrt{(x_e^2 + y_e^2 + z_e^2)}} \right) \quad (2.18)$$

(2.17) ve (2.18)'de hesaplanan açılar ZYZ notasyonu kullanılarak hesaplanmıştır. Bu açıların koordinat sistemi üzerinde ve robot üzerinde gösterimi sırasıyla Şekil 2.14 ve Şekil 2.15'te gösterilmiştir.



Şekil 2.14- Omuz açılarının ZYZ Euler açıları notasyonu ile gösterilmesi [52]



Şekil 2.15- Hesaplanan açıların robot üzerinde gösterilmesi [52].

Daha sonra (2.19)'da omuz-dirsek vektörü (V_{se}) ve dirsek-el vektörünün (V_{eh}) oluşturduğu yüzeye normal olan vektör hesaplanmıştır. (2.17) ve (2.18)'de hesaplanan açılarla (2.20)'de gösterilen rotasyon matrisi hesaplanmıştır. Oluşturulan rotasyon matrisi ile yüzey normalinin dönüş eksenini etrafında ne kadar döndüğü (2.21)'de hesaplanmıştır. (2.22)'de omuz ekleminin üçüncü açısı, (2.19) ve (2.20)'de hesaplanan yüzey normalinin nokta çarpımlarının genliklerine bölünmesi ve ters kosinüsünün alınması ile bulunmuştur. Daha sonra alt kolun oryantasyonu, bulunan iki yüzey normalinin nokta çarpımı hesaplanarak bulunmuştur. (2.23)'te dirsek ekleminin açısı, V_{se} ve V_{eh} vektörlerinin nokta çarpımlarının genliklerinin çarpımına bölünerek ters kosinüsünün alınması ile bulunmuştur. Daha sonra bulunan bu açıları robotun eklemlerine gönderilerek operatörün hareketlerini takip etmesi sağlanmıştır.

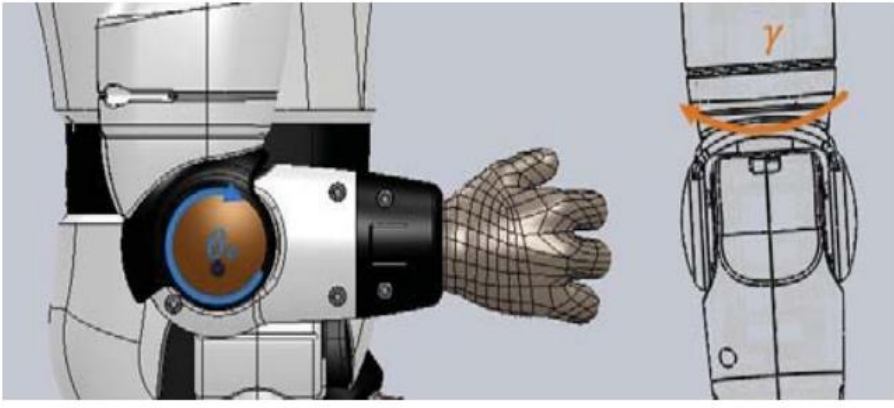
$$n_c = V_{se} \times V_{eh} \quad (2.19)$$

$$R_{\alpha\beta} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \cos\alpha\sin\beta & \cos\alpha\cos\beta & -\sin\alpha \\ \sin\alpha\sin\beta & \sin\alpha\cos\beta & \cos\alpha \end{bmatrix} \quad (2.20)$$

$$n_i = R_{\alpha\beta} * [0 \ 0 \ 1]^T \quad (2.21)$$

$$|\gamma| = \cos^{-1} \left(\frac{n_i \cdot n_c}{|n_i| |n_c|} \right) \quad (2.22)$$

$$\theta_e = \cos^{-1} \left(\frac{V_{eh} \cdot V_{se}}{|V_{eh}| |V_{se}|} \right) \quad (2.23)$$



Şekil 2.16- Dirsek eklemi açısı ve alt kol oryantasyonunun robot üzerinde gösterimi [52].

Bu alanda yapılan çalışmalardan biri de [41]'da olduğu gibi Baxter robotun Kinect ile teleoperasyonu çalışmasıdır [53]. Diğer çalışmalarda olduğu gibi bu çalışmada da ters kinematik hesaplamalarından eklem açıları hesaplanmıştır. Bu çalışmanın diğer çalışmalardan farkı ise insan uzuv vektörleri ile robotun uzuv vektörleri arasında ölçekleme yaparak eşleştirme yapmasıdır. İnsan kolu boyutunun robot koluyla aynı ölçüde olmamasından dolayı ters kinematik algoritmasının düzgün çalışabilmesi için robotun ölçüleri ile insan uzuv vektörleri arasında ölçeklendirme yapılmıştır. Aynı zamanda robotun çalışma alanını anlatan koordinat sistemi Kinect'in sunduğu koordinat sisteminden farklı olduğu için hesaplamalar yapılırken koordinat transformasyonu da hesaba katılmıştır.

Bu bölümde incelenen çalışmalara genel olarak bakıldığında Kinect etkileşiminin humanoid robotlar ve insan eklem yapısını yansıtan robotlar üzerinde sıkça

çalışıldığı görülmektedir. Bunun başlıca nedenlerinden biri humanoid robotların eklem yapıları insana çok benzer olduğu için bulunan açılar direk olarak robota gönderilebilmektedir. Çalışmalarda izlenen metodoloji genel olarak ters kinematik kullanılarak koordinat pozisyonlarından eklem açılarının bulunması ve bu açıların robota gönderilmesidir. Bunun yanı sıra ters kinematik hesaplamaların eş zamanlı operasyon için çok fazla işlem gerektirdiği ve çok zaman aldığı hemen hemen bütün çalışmalarda belirtilmiştir. Bu nedenle çalışmalardan birinde farklı bir metodoloji kullanılmış ve eş zamanlı operasyonda kullanılmak için yapay sinir ağı eğitilmiştir.

Bu bölümde taranan çalışmalar için humanoid robotlar ile yapılan çalışmalarda kullanılan teleoperasyon yöntemleri Çizelge 2.1'de özetlenmiştir. Bu çizelgede teleoperasyon yöntemleri ve bu yöntemlerin oluşturulmasında ve yöntemlerin sonucunda oluşan verilerin robota aktarılmasında kullanılan yardımcı yöntemler beraber gösterilmiştir. Bu yöntemlerden düz kinematik yöntemi analitik ters kinematik çözümü yönteminde kullanılmak üzere dönüşüm matrisini oluşturmak için kullanılmaktadır. Diğer yardımcı yöntemlerden ölçekleme yöntemi robot ile operatör arasında eklem açıları veya uzuv vektörleri arasında ölçekleme yaptıktan sonra açıları robota gönderen çalışmaları göstermektedir. Direkt besleme yöntemi ise teleoperasyon yöntemlerinin herhangi biri ile hesaplanan verilen ölçekleme yapılmadan robota aktarılmasını kullanan çalışmaları göstermektedir.

Çizelge 2.1'den de görüleceği gibi humanoid robotlar üzerine yapılan çalışmalarda kullanılan teleoperasyon yöntemleri ters kinematik çözümü ağırlıklı yöntemlerdir. Bu yöntemlere alternatif olarak bir çalışmada yapay sinir ağı yöntemi kullanılmış ve eş zamanlı operasyon performansı olarak ters kinematik yöntemine göre daha iyi sonuç verdiği gösterilmiştir. Bu sonuçtan yola çıkılarak bizim çalışmamızda da yapay sinir ağı yöntemi geliştirilmiş ve performans açısından ters kinematik çözümü yöntemi ile karşılaştırılmıştır.

Çizelge 2.1- Taranan Çalışmalarda Kullanılan Yöntemler

Ref.	Teleoperasyon Yöntemleri			Yardımcı Yöntemler		
	Ters Kin. (Analitik)	Ters Kin. (Geometrik)	Yapay Sinir Ağı	Direkt Açı Besleme	Düz Kin.	Ölçekleme
40		X		X		
42	X	X	X	X	X	
45	X				X	
47		X		X		
48		X				X
49	X				X	X
50		X		X		
52		X				X

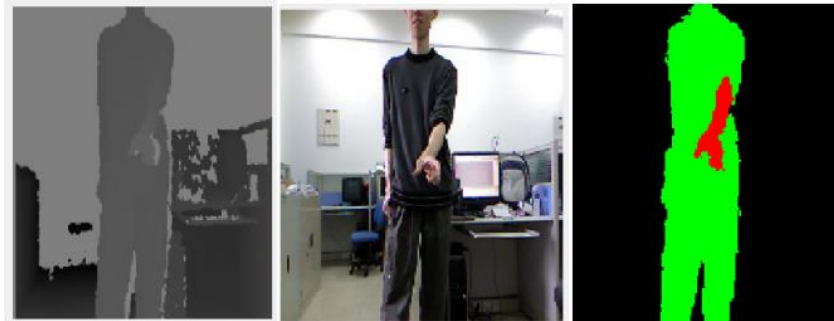
2.3.2 Kinect Kullanılarak Endüstriyel Robotlar Üzerinde Yapılan Çalışmalar

Bu bölümde Kinect kullanılarak yapılan endüstriyel robotik alanındaki çalışmalar anlatılacaktır. Bu bölümdeki çalışmalar bir önceki bölümdeki çalışmalardan sayıca daha az ve daha karmaşıktır. Bunun nedeni endüstriyel robotların eklem yapısının insan eklemleri ile birebir örtüşmemesi olarak görülebilir. Aynı zamanda endüstriyel robotların kullanımı daha çok aynı hareketleri üst üste yüksek hassasiyette yapmak üzerine olması da yine operatör tarafından Kinect kullanılarak teleoperasyon gerçekleştirme senaryosu genel olarak bu robotların kullanım amaçlarının dışındadır. Buna ek olarak yüksek hassasiyetle işlem yapmanın gerektirdiği durumlarda Kinect tarafından sağlanan konum hassasiyeti robotu kontrol etmek için yeterli değildir. Ancak bu senaryoların dışında kalan, tekrarlı hareket yerine teleoperasyon içeren ve düşük hassasiyetin kabul edilebileceği işlerde bu yöntemin kullanıldığı gözlenmiştir.

[54]'te araştırmacılar kontrol edilen robot ile operatör arasında bir merkez-uydu yapısı oluşturmak için Kinect kullanmışlardır. Burada merkez sistemi olarak operatör tarafından kullanılan Kinect sistemi gösterilmiş, uydu ise kontrol edilen Kuka KR6 6 serbestlik derecesine sahip [55] endüstriyel robot olarak tanımlanmıştır. Yapılan çalışmada iki tür operasyon tipi belirlenmiş. İlk operasyon türünde operatörün

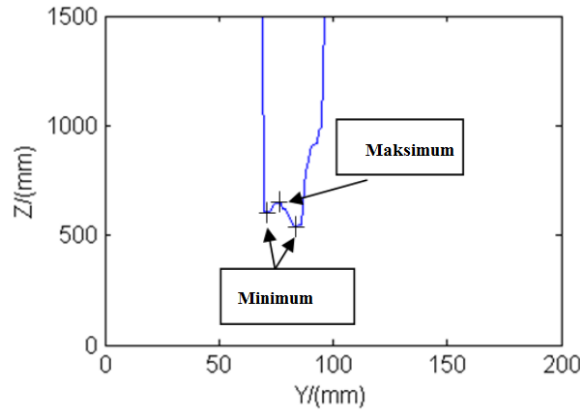
hareketleri eklem bazında robota gönderilmiştir. İkinci operasyon türünde ise robotun belirli bir nokta etrafında rotasyon yapması sağlanmıştır. Operatörün sağ elinin koordinatları robotun çalışma alanına eşleştirilerek robotun operatörün hareketlerini taklit etmesi sağlanmıştır.

Bir diğer çalışmada [56], araştırmacılar Kinect el pozlarını algılamak için kullanılmıştır. Derinlik görüntülerinden algılanan işaret parmağı ve baş parmak kullanılarak robotun uç işlemcinin açma-kapama hareketi kontrol edilmiştir. El algılama işlemi öncelikle kullanıcının kolunun ayırt edilmesi ve segmente edilmesi ile başlamıştır. Bunun için derinlik görüntüsünde eşik değer filtreleme kullanılmıştır. Kullanıcı kolunu kameraya doğru uzattığı zaman kameraya derinlik olarak en yakın bölgesi kolu olmaktadır. Bu kolu segmente edebilmek için öncelikle bütün vücudunun ortalama derinlik değeri hesaplanmış, daha sonra da bu değere göre bir eşik değeri belirlenmiştir. Belirlenen eşik değeri kullanılarak kolun derinlik görüntüsü üzerinde vücudun geri kalanından segmente edilmesi Şekil 2.17’de gösterilmiştir. Daha sonra segmente edilen üç boyutlu el görüntüsünün üzerinde üç adet nokta belirlenmiştir. Bu noktalardan I işaret parmağının uç noktasını, T başparmağın uç noktasını ve B bu iki nokta arasında bir noktayı belirlemektedir. Bu noktaların belirlenmesinde de kameraya olan uzaklık kullanılmış, kullanıcı elini öne uzatıp operasyona başladığında bu noktalar karşılaştırılmıştır. I, T ve B noktalarının sırasıyla kameraya en yakından en uzağa olacak şekilde oldukları gözlemlenmiştir. Bu noktaların üç boyutlu koordinatlarının bulunması için RGB görüntüsünden gelen her bir piksel için derinlik görüntüsünden alınan derinlik verisi kullanılmıştır. Oluşturulan üç boyutlu kol bölgesi A_{3D} bölgesi olarak adlandırılmıştır.



Şekil 2.17- Kullanıcının kolunun derinlik görüntüsünden segmente edilmesi [56].

Bu bölgedeki üç boyutlu koordinatların YZ düzlemindeki izdüşümü alınmıştır ve A_{YOZ} olarak adlandırılmıştır. Bu iz düşüm Şekil 2.18’de gösterilmiştir.



Şekil 2.18- YZ düzleminde hesaplanan maksimum ve minimum noktaları [56].

Alınan iz düşüm üzerinde minimize iz düşüm fonksiyonu tanımlanmış, bu fonksiyona göre y eksenindeki her bir nokta için z eksenine taranarak iki lokal minimum ve bir maksimum noktası hesaplanmıştır. Daha sonra bu fonksiyondan alınan çıktılarından maksimum olan I ve T noktaları arasındaki B ($y=y1$) noktası, minimumlar da I ($y=y2$) ve T ($y=y3$) noktaları olarak alınmıştır. Bu noktaların üç boyutlu koordinatlarını veren eşitlikler (2.24) – (2.27) arasında gösterilmiştir.

$$f(y) = \min_{z=1,2\dots m}(A_{YOZ}(y, z)) \quad (2.24)$$

$$I(x, y, z) = \begin{cases} x = \sum_{x=1}^n A_{3D}(x, y2, f(y2)) \\ y = y2 \\ z = f(y2) \end{cases} \quad (2.25)$$

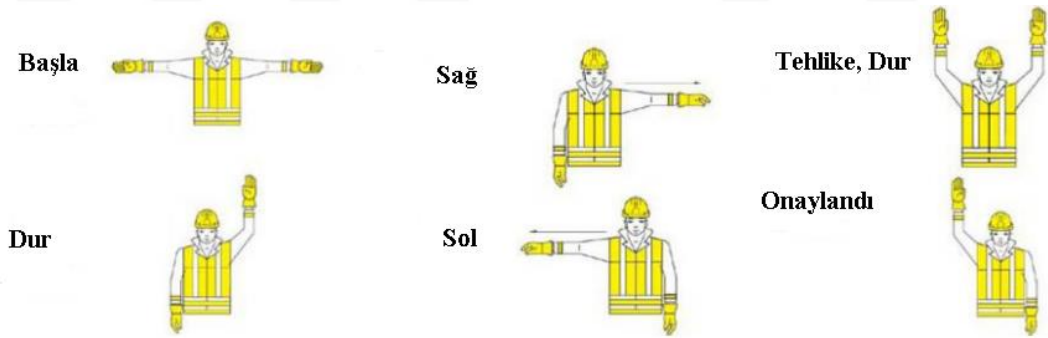
$$T(x, y, z) = \begin{cases} x = \sum_{x=1}^n A_{3D}(x, y3, f(y3)) \\ y = y3 \\ z = f(y3) \end{cases} \quad (2.26)$$

$$B(x, y, z) = \begin{cases} x = \sum_{x=1}^n A_{3D}(x, y1, f(y1)) \\ y = y1 \\ z = f(y1) \end{cases} \quad (2.27)$$

Daha sonra operatörün elindeki I ve T noktasının açıklığı ile ve robotun uç işlemcisinin açıklığı eşlenerek tutma-bırakma işlemi için kontrol sağlanmıştır. Kinect’in koordinat sistemi ile operatörün elinin koordinat sistemi arasındaki

dönüşüm matrisi hesaplanarak robotun baz eklemi ile uç işlemcisinin eklemi arasına uygulanmıştır. Daha sonra robotun Denavit-Hartenberg parametreleri ile oluşturulan baz eklem koordinatı ile uç işlemci arasındaki bağlantıyı veren dönüşüm matrisi bir önceki işlemden operatörün elinin koordinat sistemi ile Kinect koordinat sistemi arasındaki bağlantıyı veren transformasyon matrisine eşitlenerek eklem açıları hesaplanmıştır. Hesaplanan eklem açıları ve uç işlemci operasyonu robota gönderilerek tele-operasyon sağlanmıştır.

[57]'de araştırmacılar derinlik görüntülerinden statik poz algılama algoritması geliştirmişlerdir. Statik pozlar endüstride yaygın olarak vinç operatörleri ile iletişim için kullanılan pozlar olarak seçilmiştir. Statik pozların hareket algılama algoritması tarafından algılanmasını sağlamak ve pozlara göre algoritmanın eğitilmesinin eş zamanda dinamik olarak pozların çıkarılmasına göre daha kolay bir işlem olduğundan bahsedilmiştir. Aynı zamanda robot ile iletişime geçecek operatörlerin kimlik doğrulamasının yapılabilmesi için de yine Kinect ile geliştirilen yüz algılama sistemi önerilmiştir. Eş zamanlı olarak operatörün üst vücudunda bulunan eklemlerin 2B koordinatları takip edilerek daha önceden belirlenen statik pozlar ile karşılaştırılmıştır. Kullanılan statik pozlar Şekil 2.19'da gösterilmiştir.



Şekil 2.19- Çalışmada kullanılan statik pozlar [57].

Statik pozların algılanmasında basit koordinat karşılaştırmaları kullanılmıştır. Örneğin tehlike, dur pozisyonunun algılanması için kullanılan denklemler (2.28) ve (2.29)'da gösterilmiştir.

$$\alpha_1 = \begin{cases} |x_n > x_s > x_e| \\ |x_n > x_s > x_w| \\ |y_w > y_e > y_s > y_n| \end{cases} \quad (2.28)$$

$$\gamma_1 = |x_w - x_e < k| \quad (2.29)$$

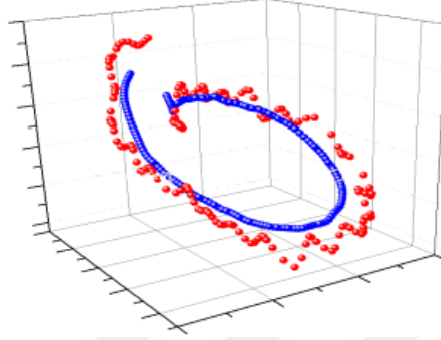
(2.28) ve (2.29)'da gösterilen n indisi boyun, s indisi omuz, e indisi dirsek, w indisi ise bilek uzuvlarını, x ve y de bu uzuvların belirtilen koordinatlardaki konumlarını belirtmektedir. (2.28)'de eklemlerin x ve y koordinatları karşılaştırılarak duruşun statik poza olan benzerliğine uyup uymadığı kontrol edilmiştir. (2.29) ise dikey hizalama için bir sabit kullanılmıştır. Bu yöntemle diğer statik duruşların eşitlikleri de çıkarılmıştır ve operatörün pozu eş zamanlı olarak bu pozların eşitlikleri ile karşılaştırılmıştır.

Çalışmada aynı zamanda eş zamanlı el koordinatı takibi de yapılmıştır. Bu konuda yapılan diğer çalışmalardan farklı olarak kol pozisyonundan açıları hesaplayıp robota göndermek yerine operatörün çalışma alanı ile robotun çalışma alanı ölçeklendirilerek birbirine eşlenmiştir. Bu eşlemeyi yapabilmek için kullanıcının elinin uzanabileceği maksimum ve minimum noktalar ile robotun uç işlemcisinin ulaşabileceği maksimum ve minimum noktalar belirlenmiştir. Daha sonra operatörün elinin çalışma alanı içerisindeki pozisyonu minimum ve maksimum noktaları ile karşılaştırılarak robot uç işlemcisinin kendi çalışma alanında bu pozisyona eşleşen koordinatı bulunarak robota gönderilmiştir.

[58]'de endüstriyel robotun ile operatör arasındaki etkileşimi sağlamak için dinamik olarak yaratılan ve robotun çalışma alanının bir kopyası olan sanal ortam yaratılmıştır. Diğer çalışmalardan farklı olarak robot ile etkileşim robotu kullanıcının kol açılarına göre hareket ettirmek yerine oluşturulan sanal ortamda daha önceden belirlenen spesifik görevleri robota yaptırarak sağlanmıştır. Aynı zamanda bu çalışmada derinlik görüntülerinin oluşturulması için iki farklı kamera türü kullanılmıştır. İlk deneylerde stereoskopik bir kamera kullanılarak derinlik görüntüleri oluşturulurken ikinci deneyde Kinect kullanılarak derinlik görüntüleri elde edilmiştir.

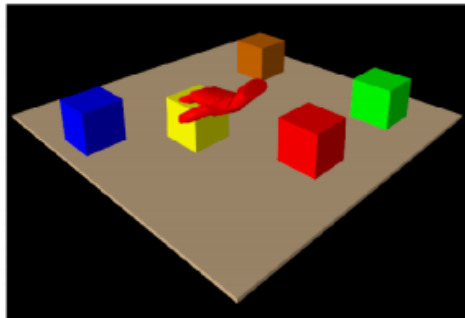
Çalışmada operatörün elleri yaratılan üç boyutlu sanal ortam içerisinde takip edilmektedir. Kullanıcının elleri tespit edildikten sonra hareket tespiti yapabilmek için Kalman filtresi kullanılarak tespit edilen el koordinatlarında düzleştirme yapılmıştır. Uygulanan üç boyutlu Kalman filtresi sonucunda elde edilen hareket yörüngesi Şekil 2.20'de gösterilmiştir. Elin sanal ortamda takibi sağlandıktan sonra

operatörün yapabileceği görevler sekiz farklı hareket ile belirlenmiştir. Bu hareketlerin her birinin tanımlanabilmesi için hareketin gerçekleştirildiği 300 örnekten oluşan veri seti ile eğitim kümesi oluşturulmuştur. Hareketler etkileşim başla/bitir, rotasyon başla/bitir, yaklaşma başla/bitir ve obje seç, obje yerleştir olarak belirlenmiştir.



Şekil 2.20- (Kırmızı) Filtre uygulanmadan önce hareket yörüngesi, (Mavi) Filtre uygulandıktan sonra düzleştirilmiş hareket yörüngesi [58].

Eğitilen hareket yörüngeleri ile kullanıcının eş zamanlı el hareketinin filtrelenmiş yörüngesi karşılaştırılarak yapılacak harekete karar verilmiştir. Bu hareketleri kullanarak operatör etkileşimi başlattıktan sonra sanal ortamda bulunan bir objeyi seçebilmekte, objeyi döndürebilmekte ve farklı bir pozisyona yerleştirdikten sonra etkileşimi bitirerek robota gönderebilmektedir. Daha sonra robot operatörün gerçekleştirdiği hareketleri kendi çalışma ortamında gerçek objeler üzerinde gerçekleştirmekte ve bu süre içerisinde operatörün sistemle etkileşimi durdurulmaktadır. Operatörün sanal ortamda objeler ile etkileşime geçerek gerçekleştirdiği hareket Şekil 2.21’de gösterilmiştir.



Şekil 2.21- Operatörün sanal ortamda bulunan bir objeyi seçme hareketinin algılanması [58].

[59] 'da arařtırmacılar operatörün el pozisyonunu bulmak için Kinect'ten alınan derinlik görüntülerini kullanmışlardır. Aynı zamanda bu çalışmada operatör elinde bulunan ataletsel ölçüm birimi(AÖB) kullanılarak elin oryantasyonu hesaplanmıştır. Elin koordinatlarının belirlenmesi için diğerk çalışmalardan farklı olarak Camshift metodu kullanılmıştır [60]. Camshift algoritmasında resimde aranan hedef kümeleme metodunu kullanarak bulunmaktadır. Algoritma bölgedeki renk bilgilerini kullanarak renk eşleřtirmesi yaparak hedefi bulmayı amaçlar. Camshift algoritması kullanılarak el takibinin yapılması ve daha sonra derinlik görüntüsünden AÖB koordinatlarının çıkarılması Şekil 2.22'de gösterilmiştir.



Şekil 2.22- Derinlik görüntüsünden IMU koordinatlarının elde edilmesi (sol), Camshift algoritması kullanılarak elin takip edilmesi (sağ) [59].

Daha sonra [61]'de anlatılan adaptif çoklu uzay dönüşümü yöntemi kullanılarak operatörün çalışma alanı ile robotun çalışma alanı birbirine eşlenmiştir. Bu yöntemde operatörün çalışma uzayı iki ölçkleme ile sanal ortamda robotun çalışma uzayına eşlenmiştir. İlk eşleme operatör tarafından gerçekleştirilen hareketin robot uç işlemcisinin sanal pozisyonuna ölçklenmesidir. İkinci eşlemede ise robotun dikey eksenindeki birim vektörü operatör hareketlerine göre ölçklenmiştir. Bu şekilde operatörün kendi çalışma alanındaki el koordinatları robotun çalışma alanındaki koordinatlara eşlenmiş, uç işlemcinin oryantasyonu için de AÖB ile hesaplanan el oryantasyonu verileri kullanılmıştır.

Bu alanda incelenen son çalışmada da Kinect çift kollu endüstriyel bir robota teleoperasyon yaptırmak için kullanılmıştır. Çalışmada teleoperasyon robotun kolları arasında tuttuğu bir obje için sağlanmıştır [62]. Robotun objeyi tutacağı yerler etiketlenerek belirlenmiştir. Daha sonra robotun 2 kolunun uç işlemcileri belirlenen

oryantasyonda sabit hızla objeye yaklaşarak önceden belirlenen temas kuvveti kriteri sağlanana kadar hareketine devam ettirilmiştir. Operatörün ellerinin üç boyutlu koordinatları Kinect kullanılarak eş zamanlı olarak takip edilmiş, ellerin birbirlerine göre koordinatları göz önüne alınarak objenin oryantasyonu sağlanmıştır. Bu operasyonlara örnek olarak robotun kavradığı objenin operatör tarafından belirlenen yörüngede hareket ettirilmesi ve son olarak üretim bandına konulması verilmiştir.

Bu bölümde özetlenen çalışmalara genel olarak bakıldığında Kinect kullanılarak endüstriyel robotik alanında henüz çalışmaların humanoid robotlar ile olduğu kadar yaygın olmadığı görülmektedir. Yapılan çalışmalar metot olarak birbirinden farklı yaklaşımlara sahiptir. Genel olarak bu alanda yapılan çalışmaları humanoid robot alanındaki gibi açı takibi ve açı gönderme değil, el takibi ve çalışma alanı eşleştirmesi şeklindedir.



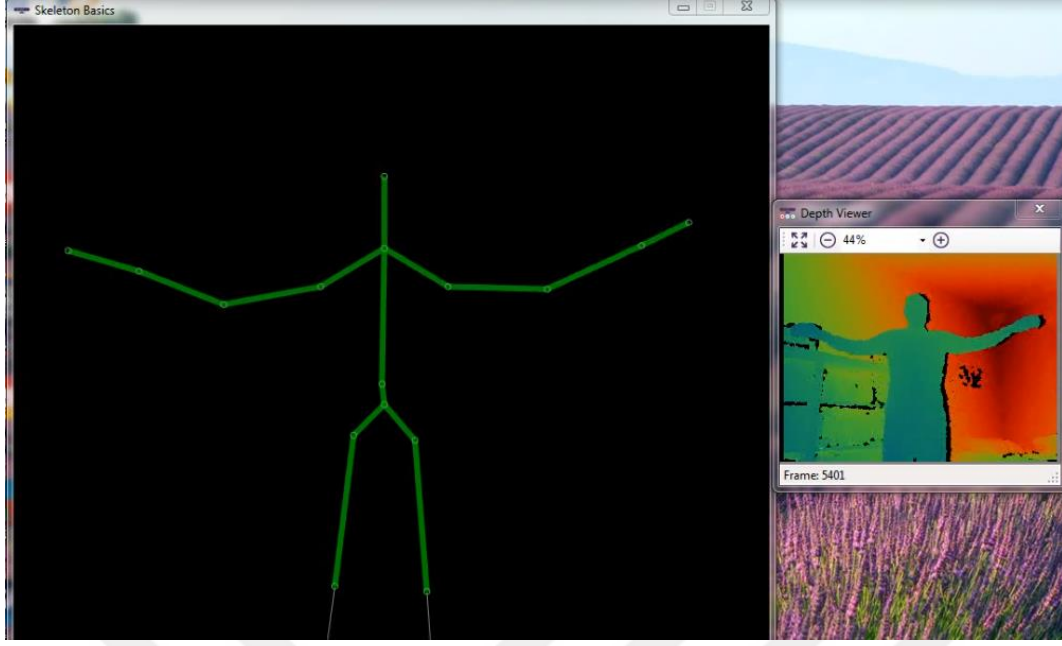
3. DERİNLİK HARİTASI KULLANARAK İSKELET TAKİBİ

Microsoft Kinect kamerasının sahip olduğu ToF kamera sayesinde kameranın çalışma alanı içerisinde kalan objelerin kameraya göre derinlik bilgileri Microsoft Kinect SDK aracılığıyla alınabilmektedir. Bu çalışmada da Microsoft Kinect SDK kullanılarak eklem takibi sistemi geliştirilmiştir.

Microsoft Kinect SDK ile birlikte sunulan örnek uygulamalardan biri iskelet takibi uygulamasıdır. Bu uygulama sayesinde kamera karşısında duran insanların eklem pozisyonları algılanarak iskelet çıkarımı yapılmaktadır. İskelet takibi sistemi Kinect SDK içerisinde bulunan doğal kullanıcı arayüzü (Natural User Interface –NUI) tarafından sağlanmaktadır. İskelet takibini etkinleştirmek ve kullanıcıların iskelet verilerini alabilmek için izlenmesi gereken adımlar aşağıda anlatılmıştır.

1. *INuiSensor::NuiSkeletonTrackingEnable* metodu çağırılarak iskelet takibi aktif hale getirilir.
2. *INuiSensor::NuiSkeletonGetNextFrame* metodu çağırılarak kamera önünde bulunan ve kamera tarafından algılanan kullanıcıların bilgileri alınır.
3. Bir önceki adımdan gelen her geçerli karede *NUI_SKELETON_FRAME* yapısı alınır. Bu yapı içerisinde kullanıcıların iskelet verilerini içeren *NUI_SKELETON_DATA* yapısını bulundurur.
4. Takip edilen iskeletler *NUI_SKELETON_DATA* yapısından dönen dizi içerisinde gezilerek bulunur ve *SkeletonPositions* dizisine erişilerek eklemlerin pozisyonları elde edilir.

Yukarıda belirtilen adımlardan 2-4 arasındaki adımlar cihazdan gelen her kare için tekrarlı bir şekilde yapılarak eklem pozisyonları elde edilir. Cihaz saniyede 30 kare hızda çalışarak 640x480 piksel çözünürlüğünde derinlik görüntüsü oluşturmaktadır. Bu adımlar izlenilerek elde edilen eklem pozisyonları arasına çizgiler çizilerek elde edilen iskelet görüntüsü Şekil 3.1’de gösterilmiştir.



Şekil 3.1- Microsoft Kinect'ten alınan derinlik görüntüsü ve iskelet bilgisi.

Bu noktada iskelet takip sisteminden elde edilen eklem koordinatları ve iskelet bilgisi gürültü içermektedir. Eklem verilerini kullanmadan önce gürültüyü azaltmak daha doğru veriler ile çalışma yapılmasını sağlamaktadır. Bu amaçla Kinect SDK tarafından sunulan eklem düzleme sistemi kullanılmıştır [63]. Microsoft Kinect SDK tarafından sağlanan düzleme filtresi ekonomi verileri ve istatistiksel analiz için kullanılan Holt Double Exponential düzleme filtresi tabanlıdır [64]. Bu filtre verilen beş parametre tarafından kontrol edilmektedir. Parametreler aşağıda tanımlanmıştır:

Smoothing : Düzleme parametresidir. 0 - 1.0 aralığında değer alabilir. Bu parametrenin yüksek değerleri eklem pozisyonlarının düzleme değerlerini artırır. Bu değer sistemin gecikme süresi ile ters orantılıdır. Önerilen parametre 0.5'tir.

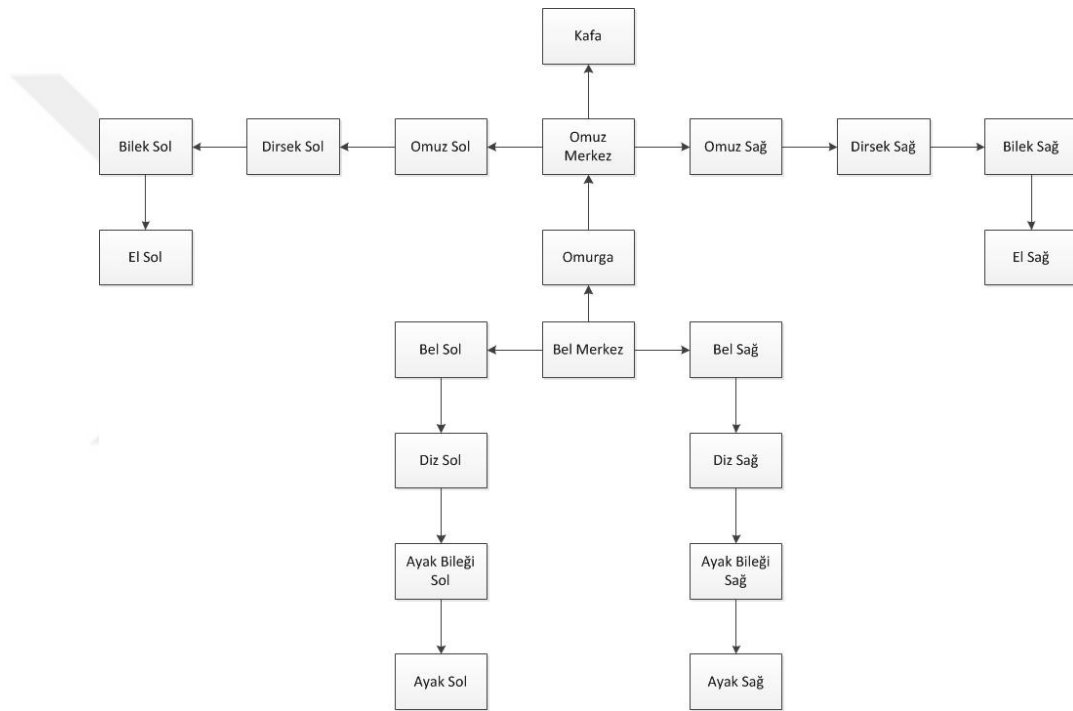
Correction: 0 -1.0 arası değer alır. Düşük değerler daha düzgün değerler verirken sistemin gecikme süresini artırır. Önerilen parametre 0.5'tir

Prediction: Gelecekte tahmin yürütülecek kare sayısını belirleyen parametredir. 0'dan büyük değerler alır. 0.5'ten büyük değerler için takip edilen kişi çok hızlı hareket ettiği zaman aşırı salınma neden olabilir. Önerilen parametre 0.5'tir.

JitterRadius: Sapma azaltma için kullanılan parametredir. Girilen parametre metre cinsinden yarıçapı ifade eder. Önerilen parametre 0.05'tir.

MaxDeviationRadius: Filtrelenmiş pozisyonların ham veriden sapma yarıçapını metre cinsinden belirten parametredir. Bu parametrenin üzerinde olan sapmalar belirtilen parametre değerinde tutulur. Önerilen parametre 0.04'tür.

Bu parametreler önerilen parametreler olarak ayarlanmış ve düzleme işlemi uygulanarak eklem koordinatları iskelet verisinden alınmıştır. Bu noktadan sonra elimizde 20 eklem koordinatından oluşan iskelet ağacı bulunmaktadır. Bu iskelet ağacı belirli bir eklem hiyerarşisine göre oluşturulmuştur. Hiyerarşinin kökünde bel merkez eklemi bulunmaktadır ve buradan kafa ve ayaklara kadar hiyerarşi devam etmektedir. Bu hiyerarşi Şekil 3.2'de gösterilmiştir.



Şekil 3.2- İskelet eklemlerinin hiyerarşik yapısı.

Hiyerarşik eklem yapısından alınan merkez omuz, sağ omuz, sağ dirsek ve sağ bilek koordinatları açı hesaplamaları yapılmak üzere bir sonraki aşamaya aktarılmıştır.

3.1 Eklem Açılarının Hesaplanması

İnsan kolunun kinematik modeli ve ters kinematik çözümleri özellikle biyomekanik alanında birçok çalışmada araştırılmıştır. Örneğin [65] 'de araştırmacılar rehabilitasyon robotunda kullanılmak üzere yedi serbestlik derecesine sahip insan

kolunun ters kinematik çözümlerini göstermişlerdir. Bu serbestlik açıları omuz için küresel eklemi gösteren üç açı, dirsek için döner eklemi gösteren bir açı, bilek için de omuz gibi küresel eklemi gösteren üç açıdan oluşmaktadır. Çalışmada ters kinematik çözümler el pozisyonu ve el oryantasyonu ile beraber üst kolun radyal ivmesi ölçülerek oluşturulmuştur. Daha sonra oluşturulan analitik model empirik yaklaşım yöntemi ile denenerek doğrulanmıştır.

Bir diğer çalışmada araştırmacılar kolun ters kinematığını eş zamanlı çözebilmek için basitleştirilmiş bir model kullanmışlardır [66]. Çalışmada omuza sabit bir koordinat sistemi yerleştirilerek diğer eklemlerin koordinatları omuza göre 4x4 homojen koordinat dönüşümü matrisleriyle ifade edilmiştir. Çalışmada bilek eklemi dönüşüm matrisi bilinmekte ve buna göre diğer transformasyon matrisleri bulunarak eklem açıları için çözüm yapılmaktadır.

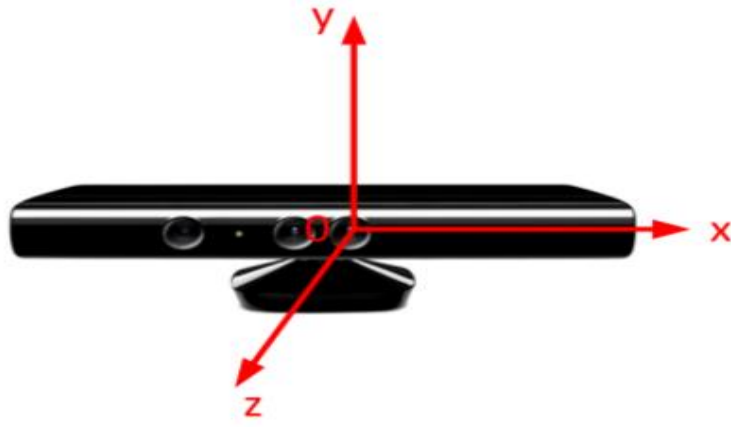
Başka bir çalışmada [67] araştırmacılar hareket takip sistemi ve eklemlere yerleştirilen yansıtıcı etiketler kullanarak kolun günlük hareketleri gerçekleştirdiği sırada dinamiklerini incelemişlerdir. Gerçekleştirilen her hareket esnasında yansıtıcı etiketlerden alınan kartezyen koordinatlar kullanılarak eklemlerin Euler açıları hesaplanmıştır. Yapılan hareketler kayıt altına alınarak, hareketlerin takibi ile elde edilen açı verileri ile oluşturulan simülasyon ortamı ile kaydedilen görüntüler karşılaştırılarak doğrulama yapılmıştır.

Bu çalışmalarda ortak olan nokta kolda takip edilen veya bilinen eklem pozisyonlarından açılarının bulunmasıdır. Bu yöntem literatürde ters kinematik yöntemi olarak bilinmektedir. Ters kinematik yönteminde gerçekleşen hareket üzerinden uzuv vektörleri ve eklemlerin kartezyen koordinatları kullanılarak uç noktanın baz noktaya göre bulunduğu pozisyonuna ulaşması için kapalı kinematik zincirde eklemlerin alması gereken açılar hesaplanmaktadır.

Bizim çalışmamızda insan kolunun dört eklemi için ters kinematik çözümleri ile hesaplanmıştır. Ters kinematik eşitliklerin oluşturulması için gerekli olan veriler ise Microsoft Kinect kullanılarak bir önceki bölümde anlatılan iskelet ve eklem takibi algoritması aracılığıyla alınmıştır.

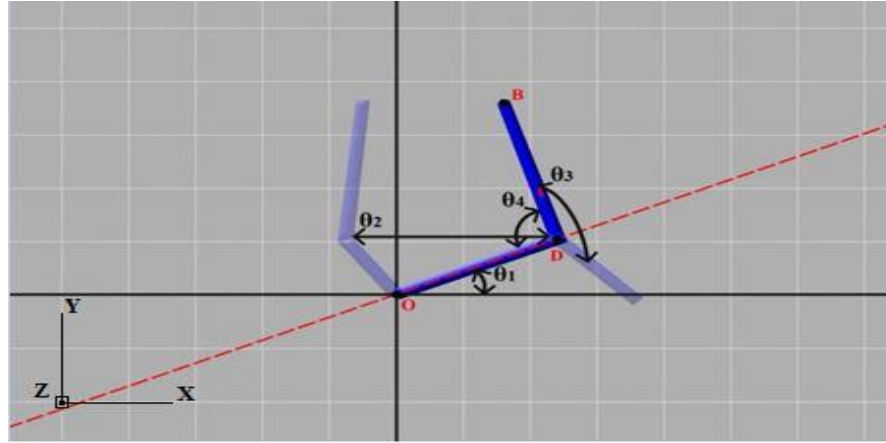
Eklemler arası açılar hiyerarşik iskelet ağacından gelen eklem koordinatları ve eklemler arası uzuv vektörleri kullanılarak hesaplanmıştır. Açılar Microsoft

Kinect'in tanımladığı koordinat eksenlerine göre hesaplanmıştır. Kinect'in kullandığı koordinat eksenleri Şekil 3.3'te gösterilmiştir.



Şekil 3.3- Kinect koordinat sistemi [68].

Açıların hesaplanabilmesi için eklem koordinatları, bu koordinatlar arasında bulunan uzuv vektörleri ve eksen vektörleri kullanılmıştır. Şekil 3.4'te gösterilen açıların hesaplanmaları (3.1) - (3.12) arasında gösterilmiştir. Şekilde gösterilen B noktası bilek, D noktası dirsek, O noktası da omuz pozisyonunu belirtmektedir. Ayrıca görsel olarak açıların gösterimi zor olduğu için şekilde gösterilen açılar, belirtilen açı değerinin iki farklı değeri arasındaki değişimi göstermektedir. θ_1 , θ_2 ve θ_3 değerleri omuz eklemının açılarını göstermektedir. θ_4 açısı ise dirsek eklemının açısını ifade etmektedir.



Şekil 3.4- İskelet modeli kol açıları.

$$\vec{V}_{od} = Pd_{x,y,z} - Po_{x,y,z} \quad (3.1)$$

$$\vec{V}_{db} = Pb_{x,y,z} - Pd_{x,y,z} \quad (3.2)$$

$$\vec{X} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \vec{Y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \vec{Z} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (3.3)$$

$$m_{od} = \frac{\vec{V}_{ody}}{\vec{V}_{odx}} \quad (3.4)$$

$$m_{\vec{V}_{od} x \vec{V}_{db}} = \frac{\vec{V}_{od} x \vec{V}_{db_y}}{\vec{V}_{od} x \vec{V}_{db_x}} \quad (3.5)$$

$$\theta_2 = \tan^{-1} \frac{\|\vec{V}_{od} x \vec{Z}\|}{\vec{V}_{od} \cdot \vec{Z}} \quad (3.6)$$

$$R_y = \begin{bmatrix} \cos(\frac{\pi}{2} - \theta_2) & 0 & \sin(\frac{\pi}{2} - \theta_2) \\ 0 & 1 & 0 \\ \sin(\frac{\pi}{2} - \theta_2) & 0 & \cos(\frac{\pi}{2} - \theta_2) \end{bmatrix} \quad (3.7)$$

$$\vec{V}_{rotx} = R_y \cdot \vec{X} \quad (3.8)$$

$$\vec{V}_{rotz} = R_y \cdot \vec{Z} \quad (3.9)$$

$$\theta_1 = \begin{cases} \tan^{-1} \frac{\|\vec{V}_{od} x \vec{V}_{rotx}\|}{\vec{V}_{od} \cdot \vec{V}_{rotx}}, & m_{od} > 0 \\ \tan^{-1} \frac{-\|\vec{V}_{od} x \vec{V}_{rotx}\|}{\vec{V}_{od} \cdot \vec{V}_{rotx}}, & m_{od} < 0 \end{cases} \quad (3.10)$$

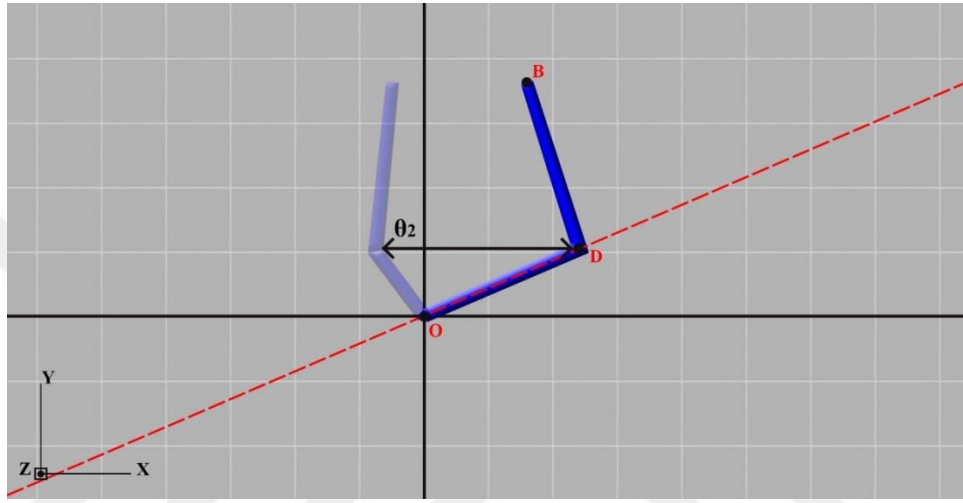
$$\theta_3 = \begin{cases} \tan^{-1} \frac{\|\vec{V}_{db} x \vec{V}_{rotz}\|}{\vec{V}_{db} \cdot \vec{V}_{rotz}}, & m_{\vec{V}_{od} x \vec{V}_{db}} < 0 \parallel P_{bx} > P_{dx} \\ \tan^{-1} \frac{-\|\vec{V}_{db} x \vec{V}_{rotz}\|}{\vec{V}_{db} \cdot \vec{V}_{rotz}}, & m_{\vec{V}_{od} x \vec{V}_{db}} < 0 \end{cases} \quad (3.11)$$

$$\theta_4 = \begin{cases} \tan^{-1} \frac{\|\vec{V}_{od} x \vec{V}_{db}\|}{\vec{V}_{od} x \vec{V}_{db}}, & m_{\vec{V}_{od} x \vec{V}_{db}} > 0 \\ \tan^{-1} \frac{-\|\vec{V}_{od} x \vec{V}_{db}\|}{\vec{V}_{od} x \vec{V}_{db}}, & m_{\vec{V}_{od} x \vec{V}_{db}} < 0 \parallel P_{bx} > 0 \end{cases} \quad (3.12)$$

(3.1) ve (3.2)'de sırasıyla omuz-dirsek ve dirsek-bilek vektörleri tanımlanmıştır. Bu vektörlerin hesaplanmasında kullanılan $Po_{x,y,z}$ omuz, $Pd_{x,y,z}$ dirsek, $Pb_{x,y,z}$ de bilek eklemlerinin Kinect'ten alınan uzaysal koordinatlarıdır. (3.3)'te x, y ve z eksenlerinin birim vektörleri tanımlanmıştır. Kinect kameranın koordinat sisteminde z ekseninin kameradan dışarı doğru uzandığı kabul edildiği için z eksenini negatif yönde

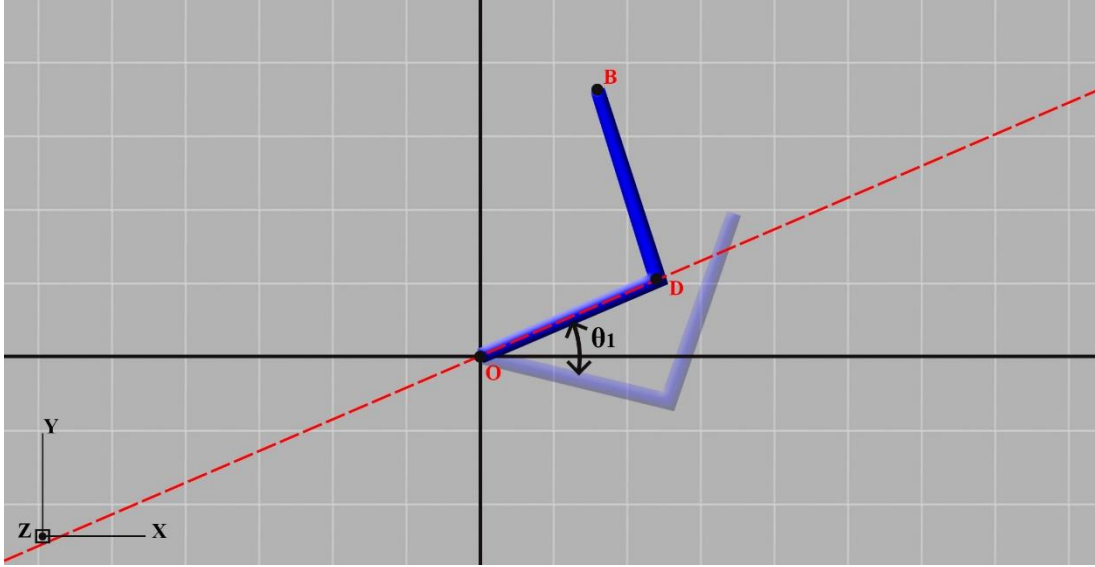
tanımlanmıştır. (3.4) omuz-dirsek vektörünün eğimini vermektedir. Daha sonra bu değer eşitlik (3.10)'da θ_1 açısının hesaplanmasında kullanılmıştır.

θ_2 açısı (3.6)'da hesaplanmıştır. Bu açı omuz-dirsek vektörünün z eksenini ile yaptığı küçük açıdır. Şekil 3.5 θ_2 açısının z eksenini etrafında tarayabileceği değerleri göstermektedir. θ_2 açısı daha sonra θ_1 ve θ_3 açılarının hesaplanmasında da kullanılmıştır. Bu açıların hesaplandığı eksenler y eksenini etrafında (3.7), (3.8) ve (3.9)'da gösterildiği gibi döndürülmüşlerdir.



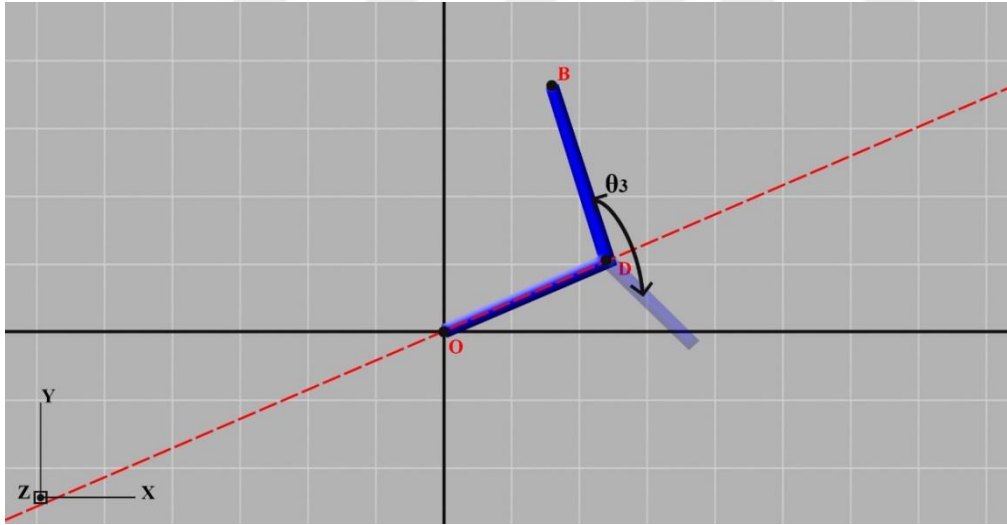
Şekil 3.5- θ_2 açısını gösteren hareket yörüngesi.

θ_1 açısı omuz-dirsek vektörünün x ekseninin (3.7) ve (3.8)'de gösterilen eşitlikler kullanılarak y eksenini etrafında döndürülmesi ile elde edilen eksen ile yaptığı açıyı temsil etmektedir. Aynı zamanda bu açı 3 serbestlik derecesine sahip omuz ekleminin ilk açısıdır. İnsan omuzunun hareket açısını x eksenini üzerinde 180 derece olarak düşünürsek x ekseninde 1. bölgede kalan açılar pozitif, 4. bölgede olan açılar da negatif olarak tanımlanmıştır. θ_1 açısı (3.4)'te verilen m_{od} değerinin işaretine göre işaret değiştirmektedir. θ_1 açısının x eksenini etrafında alabileceği değerler Şekil 3.6'da gösterilmiştir. Bu gösterimde x eksenini üzerinde kalan yerer pozitif değerleri, altında kalan yerler de negatif değerleri göstermektedir.



Şekil 3.6- θ_1 açısının negatif ve pozitif değerleri.

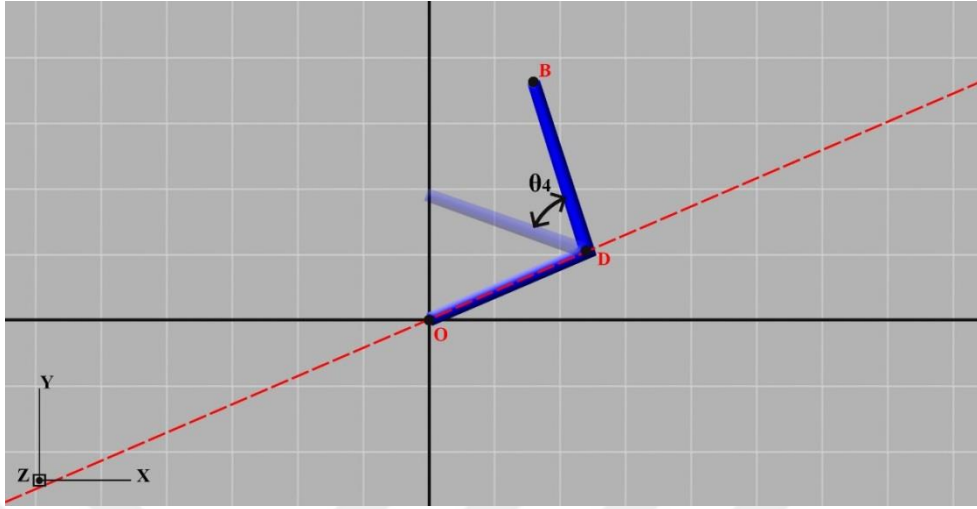
θ_3 açısı \vec{V}_{ab} vektörünün z ekseninin (3.7) ve (3.9)'da gösterilen eşitlikler ile y eksenini etrafında döndürülmesi ile oluşan eksen ile arasındaki açıdır. (3.11) kullanılarak θ_3 açısı hesaplanmıştır. θ_3 açısı Şekil 3.7'de gösterilmiştir.



Şekil 3.7- θ_3 açısını gösteren hareket yörüngesi.

θ_4 açısı ise \vec{V}_{od} vektörü ile \vec{V}_{ab} vektörü arasındaki açıyı göstermektedir. Bu iki vektör arasında birbirlerini 360° 'ye tamamlayan iki açı bulunmaktadır. Kolun kinematik yapısı göz önüne alındığında bulunması gereken açı bu açılardan küçük olanıdır. θ_4 açısını hesaplayan eşitlik (3.12)'de gösterilmiştir. θ_4 açısı (3.7)'de verilen $\vec{V}_{od} \times \vec{V}_{ab}$ vektörel çarpımı sonucunda oluşan vektörün eğimi ve P_{bx} noktasının değerine göre

işaret değiştirmektedir. Hesaplanan θ_4 açısını gösteren kol hareketi Şekil 3.8’de gösterilmektedir.



Şekil 3.8- θ_4 açısını gösteren kol hareketi.

Operatör kol açıları Microsoft Kinect’ten gelen her bir derinlik görüntüsü için hesaplanmaktadır, bu da saniyede 30 kere hesaplandıkları anlamına gelmektedir. Hesaplanan açılar bir sonraki bölümde anlatılan insan kolunun kinematik modeli kullanılarak kontrol edilmiştir. Sadece kinematik model ile doğrulanan açı değerleri sistemin geri kalanında kullanılmaktadır.

3.2 İnsan Kolu Düz Kinematiği

Bir önceki bölümde hesaplanan açılar Kinect’ten gelen her karede hesaplanmaktadır. Ancak gelen derinlik görüntüsü karelerinde gürültü olma ihtimali her zaman vardır. Örneğin Kinect iskelet takibi uygulaması çalışırken uzuvların birbirlerinin üzerine geldiği ya da karışık eklem pozisyonları gibi durumlarda eklem pozisyonlarında anlık sapmalar oluşabilmektedir. Bu durumda oluşabilecek yanlış hesaplamaları önlemek amacıyla insan kolunun düz kinematik denklemleri oluşturulmuş ve bileğin üç boyutlu pozisyonu omuzun üç boyutlu pozisyonuna göre hesaplanmıştır. Düz kinematik eşitlikler robotik alanında kullanımı yaygın olan Denavit-Hartenberg(DH) metodu takip edilerek oluşturulmuştur. Kullanılan DH parametreleri Çizelge 3.1’de gösterilmiştir. Bu parametrelerden a_i bağ uzunluğunu, d_i bağ ofsetini, α_i eksen açısını, θ_i de eklem açısını ifade etmektedir.

Çizelge 3.1- İnsan kolu DH parametreleri.

i	a	d	α	θ
1	0	0	$\pi/2$	θ_1
2	0	0	$\pi/2$	θ_2
3	0	$ V_{OD} $	$\pi/2$	θ_3
4	0	0	$-\pi/2$	θ_4
5	0	$ V_{DB} $	$\pi/2$	0

Çizelge 3.1’de gösterilen $|V_{OD}|$ ve $|V_{DB}|$ parametreleri omuz-dirsek ve dirsek-bilek vektörlerinin uzunluklarını göstermektedir. DH parametreleri ile oluşturulan düz kinematik denklemleri (3.13) – (3.19) arasında gösterilmiştir.

$$\begin{aligned}
 A_i &= R_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} R_{x,\alpha_i} \\
 &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & c\alpha_i & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)
 \end{aligned}$$

$$A_1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 \sin\alpha_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 \sin\alpha_1 & 0 \\ 0 & \sin\alpha_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

$$A_2 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \sin\alpha_2 & 0 \\ \sin\theta_2 & 0 & -\cos\theta_2 \sin\alpha_2 & 0 \\ 0 & \sin\alpha_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

$$A_3 = \begin{bmatrix} \cos\theta_3 & 0 & \sin\theta_3 \sin\alpha_3 & 0 \\ \sin\theta_3 & 0 & -\cos\theta_3 \sin\alpha_3 & 0 \\ 0 & \sin\alpha_3 & 0 & |V_{OD}| \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$$A_4 = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 \sin\alpha_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 \sin\alpha_4 & 0 \\ 0 & -\sin\alpha_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$$A_5 = \begin{bmatrix} \cos \theta_5 & 0 & \sin \theta_5 \sin \alpha_5 & 0 \\ \sin \theta_5 & 0 & -\cos \theta_5 \sin \alpha_5 & 0 \\ 0 & \sin \alpha_5 & 0 & |V_{DB}| \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

$$A_{OB} = A_1 A_2 A_3 A_4 A_5 \quad (3.19)$$

$$A_{OB} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

(3.13) 'te T_i homojen dönüşüm matrisinin dört basit dönüşümün çarpımından oluştuğu gösterilmektedir. Bu dönüşüm matrisleri çarpım sırasıyla z ekseninde θ_i kadar dönmeyi, z_i ekseninde d_i kadarlık ötelemeyi, x_i ekseninde a_i kadarlık ötelemeyi ve x_i ekseninde α_i kadarlık dönmeyi ifade eder. Daha sonra bunlar birinci çerçeveden beşinci çerçeveye kadar olan dönüşüm matrisleri eşitlik (3.14) – (3.18) arasında gösterilmektedir. (3.19) baz eklem olan omuza göre son eklem olan bilek eklemine transformasyon matrisini vermektedir. (3.20) ve düz kinematik eşitlikleri kullanılarak bilek pozisyonu hesaplanmaktadır. Aynı zamanda Microsoft Kinect'ten alınan bilek eklemi koordinatları ile kolun kinematik eşitliklerinden hesaplanan bilek koordinat değeri karşılaştırılarak doğrulama fonksiyonu oluşturulmuştur.

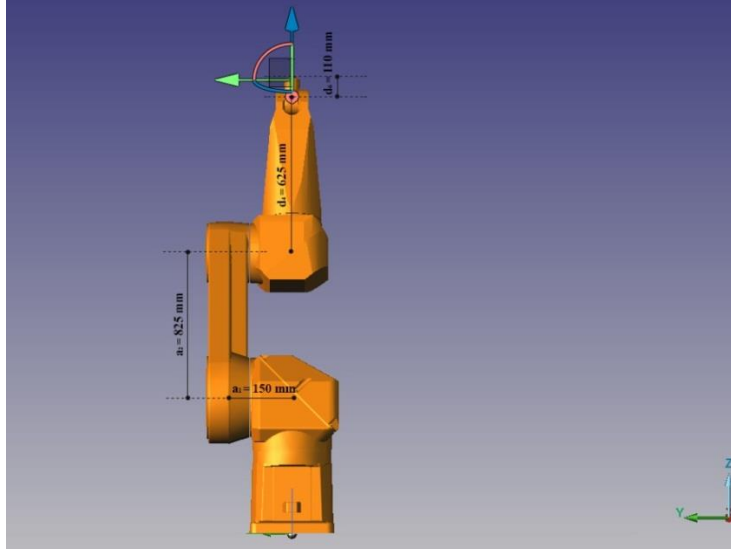
Çizelge 3.2'de doğrulanan eklem koordinatları ve bu koordinatlardan hesaplanan eklem açıları insan kolu çalışma alanının robotun çalışma alanına eşlenmesi için bir sonraki aşamaya gönderilir.

Çizelge 3.2- Doğrulama Fonksiyonu Sahte Kodu

```
while (kinectSkeleton.nextFrame)
Kinect'ten gelen omuz eklemi koordinatlarını al  $\rightarrow P_O$ 
Kinect'ten gelen bilek eklemi koordinatlarını al  $\rightarrow P_B$ 
Düz kinematik eşitlikleri ile omuz eklemi pozisyonuna göre bilek
ekleminin pozisyonunu hesapla  $\rightarrow P_{BH}$ 
 $P_B$  ile  $P_{BH}$  karşılaştır  $\rightarrow |P_{BH_i} - P_{B_i}| < 0.05, i = x, y, z$ 
```

3.3 İnsan Kolu Çalışma Alanının Robot Çalışma Uzayına Eşlenmesi

İnsan kolu kinematiğinin DH parametrelerinden görüleceği üzere insan kolunun bağ uzunluğu parametreleri Kinect'ten alınan eklem koordinatları arasında tanımlanan V_{OD} ve V_{DB} vektörlerinin uzunlukları olarak tanımlanmıştır. Bu nedenle operatörün çalışma uzayı omuz noktası merkez noktası olan ve yarıçapı üst kol ve alt kol vektörlerinin toplamı olan bir küre olarak gösterilebilir. Ancak kontrol edilen robotun çalışma uzayı boyutlarından dolayı insan kolunun çalışma uzayından çok daha geniştir. Kontrol senaryosu olarak operatör çalışma uzayının genişletilmesi ve bu sayede insanın uzularını kullanarak erişemeyeceği noktalara erişerek operasyon yapabilmesi seçildiği için çalışma uzayı eşlemesi operatörün çalışma uzayını genişletme senaryosu üzerine kurulmuştur. Çalışmada kullanılan Stäubli Rx160 robot kolu yere monte olarak konfigüre edildiği için çalışma uzayı, insan kolunun çalışma uzayının koordinat sisteminde bulunan dikey eksen birbirine ters yöndedir. Kol modelinde dikey ekseninde negatif yönde yapılan bir hareket birebire eşleme yönteminde robot için pozitif dikey ekseninde oluşmaktadır. Operatörün çalışma uzayının genişletilmesi için Stäubli Rx160 robot kolunun bağ uzunlukları kullanılmıştır. Bu uzunluklar üretici firmanın yayınladığı broşürden alınmıştır [69]. Robotun koordinat eksenleri ve bağ uzunlukları arasındaki Şekil 3.9'da gösterilmiştir.



Şekil 3.9- Stäubli Rx160 bağ uzunluğu ve bağ ofseti parametreleri

$$d_1 = 150 \text{ mm} \quad (3.21)$$

$$|V_{OD_R}| = 825 \text{ mm} \quad (3.22)$$

$$|V_{DB_R}| = 625 + 110 = 735 \text{ mm} \quad (3.23)$$

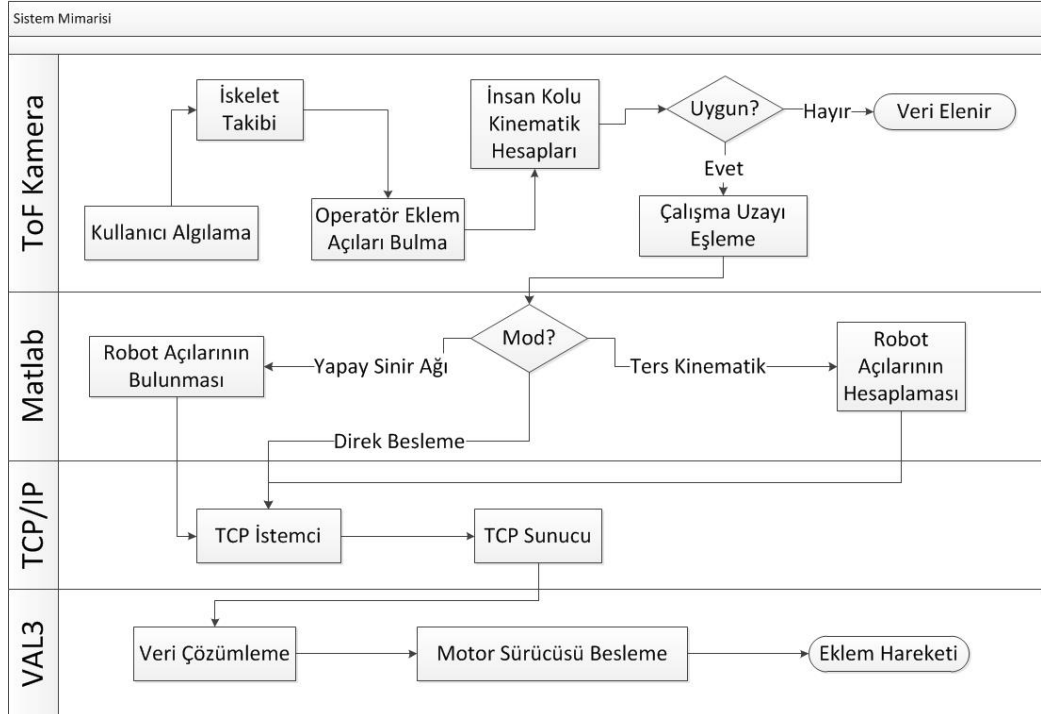
Robotun bađ ofseti ve bađ uzunluđu parametreleri kullanılarak izelge 3.1’de gsterilen insan kolunun DH parametreleri geniřletilmiřtir. (3.21)’de gsterilen robotun baz bađ erevesi ve birinci eklemi arasındaki bađ uzunluđu insan kolunun birinci bađ erevesinin bađ uzunluđu olarak deđiřtirilmiřtir. Kolun nc ekleminin bađ uzunluđu olan V_{OD} vektrnn uzunluđu (3.22)’ de verilen robotun ikinci ve nc eklemleri arasında kalan bađ uzunluđu deđeri ile deđiřtirilmiřtir. Kolun beřinci bađ erevesinde bulunan V_{DB} vektrnn uzunluđu ise (3.23)’te robotun nc ile drdnc ve drdnc ile beřinci eklemleri arasında kalan bađ uzunluklarının toplamı olarak deđiřtirilmiřtir.

Operatr alıřma uzayı ile robot alıřma uzayı eřlendikten sonra bulunan aılar ile farklı teleoperasyon yntemleri denenmiřtir. Bu yntemler Blm 4’te anlatılmıřtır.



4. TELEOPERASYON YÖNTEMLERİ

Bu bölümde 3. Bölüm’de bulunan operatör açıları ile robotun merkez-uydu yapısı ile kontrol edilmesini sağlayan yöntemler anlatılmıştır. Üç farklı yöntem kullanılarak teleoperasyon yöntemleri test edilmiştir. Bunlardan ilki operatör açılarının robot kolunun açlarına direkt olarak eşlenerek üç serbestlik dereceli bir yöntem denenmiştir. Diğer iki yöntemde operatör açıları yerine uç işlemci pozisyonu eşlenerek bu pozisyonu sağlayacak robot eklem açlarına ulaşılmıştır. Robota aktarılan altı açıdan θ_4 , θ_5 ve θ_6 açıları bileğin yönelimi ile ilgili açılardır. Kinect’ten alınan derinlik görüntüsü üzerinde bilek yönelimi ile ilgili herhangi bir hesaplama yapılmadığı için bu açılar uç işlemcinin açık olacağı şekilde sıfıra eşitlenerek robota gönderilmiştir. Bu nedenle diğer iki teleoperasyon yöntemi de üç serbestlik derecesine sahip yöntemlerdir.



Şekil 4.1- Genel Sistem Mimarisi

Bölüm 4.1 direkt eşleme yöntemini, Bölüm 4.2 uç işlemci pozisyonundan açıların hesaplandığı eş zamanlı ters kinematik yöntemi, Bölüm 4.3 de ters kinematik yönteminin sonuçlarından oluşturulmuş veri seti ile eğitilen yapay sinir ağı yöntemini anlatılmaktadır. Sonuçlar Bölüm 5'te karşılaştırılacaktır.

4.1 Direkt Eşleme Yöntemi

İnsan kolu yedi, çalışmada kullanılan Stäubli Rx160 robotu ise altı serbestlik derecesine sahiptir. Bu nedenle direkt besleme yönteminde bir serbestlik derecesi kaybolacaktır. Bu da açıların eşlenmesi durumunda dahi uç işlemci ve eşlenmiş bilek pozisyonlarında farklılık olmasına neden olacaktır.

İnsan kolu modeli ve Stäubli Rx160 robotu oluşturulan modeller üzerinden karşılaştırılmış ve (4.1) – (4.3) arasında gösterilen açı eşleme eşitlikleri elde edilmiştir. Bu karşılaştırma yapılırken kullanılan modellerin koordinat sistemleri arasındaki fark ve mekanizmaların dikey ekseninde birbirlerinin simetriği oldukları da hesaba katılmıştır.

$$\theta_{1R} = -90 - \theta_3 \quad (4.1)$$

$$\theta_{2R} = 90 - \theta_1 \quad (4.2)$$

$$\theta_{3R} = -\theta_4 \quad (4.3)$$

(4.1) – (4.3) arasında bulunan eşitlikler robotun ilk üç eklem açısı olan θ_{1R} , θ_{2R} ve θ_{3R} açılarını göstermektedir. Robotun ilk 3 eklemindeki hareketler iki omuz ve bir dirsek açısından oluşmaktadır. Bu eşitliklerde insan kinematiğinden hesaplanan θ_2 kullanılmamıştır. Bunun nedeni robotun omuz ekleminin insandan farklı olarak iki serbestlik derecesine sahip olmasıdır. Direkt açı besleme yöntemi kullanıldığında insan kolu, robotun kinematik yapısından dolayı gerçekleştirilemeyeceği hareketler yapabilmektedir. Bu nedenle bu yöntem sadece diğer iki yöntemle karşılaştırma amacıyla kullanılmıştır. Direkt açı besleme ile elde edilen sonuçlar, diğer yöntemlerle elde edilen sonuçlarla beraber Bölüm 5'te karşılaştırılmıştır.

4.2 Eş zamanlı Ters Kinematik Çözümü Yöntemi

Düz kinematik eşitliklerini oluşturmak için sistematik yaklaşımlar vardır, ancak ters kinematik için bunun gibi sistematik yöntemler bulunmamaktadır. Bu nedenle ters kinematik çözümleri analitik olarak yapılmaktadır.

Yapılan literatür araştırmasında Stäubli Rx160 robotun ters kinematik çözümlerinin yapıldığı bir yüksek lisans tezi çalışması incelenmiştir [70]. Bu çalışmada ters kinematik çözüm altıncı eklemin baz ekleme göre modifiye edilmiş DH parametreleri [71] kullanılarak oluşturulmuş dönüşüm matrisinden yola çıkılarak yapılmıştır. Modifiye edilmiş DH yöntemi klasik DH yöntemi gibi sistematik adımlardan oluşmaktadır, ancak bu yöntemde parametrelerin elde edilme yöntemleri klasik yöntemle göre farklılık göstermektedir. Modifiye edilmiş DH yöntemi ile klasik DH yöntemi [72]'de karşılaştırılmıştır. Bizim çalışmamızda modifiye edilmiş DH parametreleri kullanılmıştır. Çizelge 4.1'de Stäubli Rx160 robotunun modifiye edilmiş DH parametreleri gösterilmiştir.

Çizelge 4.1- Stäubli Rx160 modifiye edilmiş DH parametreleri [70].

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	$\frac{-\pi}{2}$	150 mm	0	θ_2
3	0	825 mm	0	θ_3
4	$\frac{-\pi}{2}$	0	625 mm	θ_4
5	$\frac{\pi}{2}$	0	0	θ_5
6	$\frac{-\pi}{2}$	0	110 mm	θ_6

Çizelge 4.1'de verilen modifiye edilmiş DH parametreleri ile yine modifiye edilmiş DH metodu kullanılarak oluşturulan düz kinematik eşitlikleri (4.4) - (3.55) arasında gösterilmiştir.

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$${}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_2 & -\cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

$${}^2T_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_2 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

$${}^3T_4 = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -\sin \theta_4 & -\cos \theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

$${}^4T_5 = \begin{bmatrix} \cos \theta_5 & -\sin \theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

$${}^5T_6 = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ -\sin \theta_6 & -\cos \theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

$${}^0T_6 = {}^0T_1(\theta_1) {}^1T_2(\theta_2) {}^2T_3(\theta_3) {}^3T_4(\theta_4) {}^4T_5(\theta_5) {}^5T_6(\theta_6) \quad (4.11)$$

$${}^0T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

$$c_i = \cos \theta_i \quad (4.13)$$

$$s_i = \sin \theta_i \quad (4.14)$$

$$c_{ij} = \cos \theta_i \cos \theta_j - \sin \theta_i \sin \theta_j \quad (4.15)$$

$$s_{ij} = \cos \theta_i \sin \theta_j + \sin \theta_i \cos \theta_j \quad (4.16)$$

$$r_{11} = c_1[c_{23}(c_4 c_5 c_6 - s_4 s_6) - s_{23} s_5 c_6] + s_{23}(s_4 c_5 c_6 + c_4 s_6) \quad (4.17)$$

$$r_{12} = c_1[c_{23}(-c_4 c_5 s_6 - s_4 c_6) + s_{23} s_5 s_6] + s_1(c_4 c_6 - s_4 c_5 s_6) \quad (4.18)$$

$$r_{13} = -c_1(c_{23} c_4 s_5 + s_{23} c_5) - s_1 s_4 s_5 \quad (4.19)$$

$$r_{21} = s_1[c_{23}(c_4 c_5 c_6 - s_4 s_6) - s_{23} s_5 c_6] - c_1(s_4 c_5 c_6 + c_4 s_6) \quad (4.20)$$

$$r_{22} = s_1[c_{23}(-c_4 c_5 s_6 - s_4 c_6) + s_{23} s_5 s_6] - c_1(c_4 c_6 - s_4 c_5 s_6) \quad (4.21)$$

$$r_{23} = -s_1(c_{23} c_4 s_5 + s_{23} c_5) + c_1 s_4 s_5 \quad (4.22)$$

$$r_{31} = -s_{23}(c_4 c_5 c_6 - s_4 s_6) - c_{23} s_5 c_6 \quad (4.23)$$

$$r_{32} = s_{23}(c_4 c_5 s_6 + s_4 c_6) + c_{23} s_5 s_6 \quad (4.24)$$

$$r_{33} = s_{23} c_4 s_5 - c_{23} c_5 \quad (4.25)$$

$$p_x = c_1[a_1 + a_2 c_2 - d_4 s_{23} - d_6(s_{23} c_5 + c_{23} c_4 s_5)] - d_6 s_1 s_4 s_5 \quad (4.26)$$

$$p_y = s_1[a_1 + a_2 c_2 - d_4 s_{23} - d_6(s_{23} c_5 + c_{23} c_4 s_5)] + d_6 c_1 s_4 s_5 \quad (4.27)$$

$$p_z = -a_2 s_2 - d_4 c_{23} - d_6(c_{23} c_5 - s_{23} c_4 s_5) \quad (4.28)$$

(4.4) – (4.28) arasında gösterilen düz kinematik eşitliklerden yola çıkılarak eklem açıları için çözüm kümeleri oluşturulmuştur. İlk olarak (4.12)'nin her iki tarafı (4.5)'te gösterilen 0_1T matrisinin tersi ile çarpılmıştır.

$$[{}^0_1T(\theta_1)]^{-1}{}^0_6T = {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) {}^4_5T(\theta_5) {}^5_6T(\theta_6) \quad (4.29)$$

$$[{}^0_1T(\theta_1)]^{-1}{}^0_6T = \begin{bmatrix} c_1 r_{11} + r_{21} s_1 & c_1 r_{12} + r_{21} s_1 & c_1 r_{13} + r_{23} s_1 & c_1 p_x + p_y s_1 \\ c_1 r_{21} - r_{11} s_1 & c_1 r_{22} - r_{12} s_1 & c_1 r_{23} - r_{13} s_1 & c_1 p_y - p_x s_1 \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} r_{11}^1 & r_{12}^1 & r_{13}^1 & p_x^1 \\ r_{21}^1 & r_{22}^1 & r_{23}^1 & p_y^1 \\ r_{31}^1 & r_{32}^1 & r_{33}^1 & p_z^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.30)$$

(4.30)'un sağ tarafında bulunan dönüşüm matrisi birinci eklemden altıncı ekleme kadar olan transformasyonu göstermektedir. Bu matrisi bulmak için baz eklem transformasyon matrisi (0_1T) dışındaki transformasyon matrisleri çarpılmalıdır. Bu

çarpımlar sonucunda oluşan matrisin elemanları (4.31) – (4.43) arasında gösterilmektedir.

$${}^1T = {}_2T_3 {}_3T_4 {}_4T_5 {}_5T_6 = \begin{bmatrix} r_{11}^1 & r_{12}^1 & r_{13}^1 & p_x^1 \\ r_{21}^1 & r_{22}^1 & r_{23}^1 & p_y^1 \\ r_{31}^1 & r_{32}^1 & r_{33}^1 & p_z^1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.31)$$

$$r_{11}^1 = c_{23}(c_4 c_5 c_6 - s_4 s_6) - s_{23} s_5 c_6 \quad (4.32)$$

$$r_{12}^1 = -c_{23}(c_4 c_5 s_6 + s_4 c_6) + s_{23} s_5 s_6 \quad (4.33)$$

$$r_{13}^1 = -c_{23} c_4 s_5 - s_{23} c_5 \quad (4.34)$$

$$r_{21}^1 = -s_4 c_5 c_6 - c_4 s_6 \quad (4.35)$$

$$r_{22}^1 = s_4 c_5 s_6 - c_4 c_6 \quad (4.36)$$

$$r_{23}^1 = s_4 s_5 \quad (4.37)$$

$$r_{31}^1 = s_{23}(s_4 s_6 - c_4 c_5 c_6) - c_{23} s_5 c_6 \quad (4.38)$$

$$r_{32}^1 = s_{23}(c_4 c_5 s_6 + s_4 c_6) + c_{23} s_5 s_6 \quad (4.39)$$

$$r_{33}^1 = s_{23} c_4 s_5 - c_{23} c_5 \quad (4.40)$$

$$p_x^1 = a_1 + a_2 c_2 - d_4 s_{23} - d_6 (s_{23} c_5 + c_{23} c_4 s_5) \quad (4.41)$$

$$p_y^1 = d_6 s_4 s_5 \quad (4.42)$$

$$p_z^1 = -a_2 s_2 - d_4 c_{23} - d_6 (c_{23} c_5 - s_{23} c_4 s_5) \quad (4.43)$$

(4.30) ve (4.31)'de verilen matrisler birbirlerine eşittir. Buradan iki matrisin (2,3) ve (2,4) elemanları (4.44) ve (4.45) 'deki denklemleri oluşturur.

$$c_1 r_{23} - r_{13} s_1 = s_4 s_5 \quad (4.44)$$

$$c_1 p_y - p_x s_1 = d_6 s_4 s_5 \quad (4.45)$$

(4.44)'deki ifade (4.45)'te yerine konup sadeleştirme yapıldığında (4.46) ve (4.47)'deki ifadeler elde edilir.

$$c_1 p_y - p_x s_1 = d_6 (c_1 r_{23} - r_{13} s_1) \quad (4.46)$$

$$s_1 (d_6 r_{13} - p_x) = c_1 (d_6 r_{23} - p_y) \quad (4.47)$$

Daha sonra (4.48) kullanılarak θ_1 elde edilir. θ_1 açısı diğer eklem açılarının hesaplanmasında bilinen olarak kullanılmıştır.

$$\tan(\theta_1) = \frac{d_6 r_{23} - p_y}{d_6 r_{13} - p_x} = \frac{p_y - d_6 r_{23}}{p_x - d_6 r_{13}} \quad (4.48)$$

$$\theta_1 = \begin{cases} \text{atan2}((d_6 r_{13} - p_x), (d_6 r_{13} - p_y)) \\ \text{atan2}((p_x - d_6 r_{13}), (p_y - d_6 r_{13})) \end{cases} \quad (4.49)$$

(4.30)'da gösterilen matrislerin (1,3), (1,4), (3,3) ve (3,4) elemanlarının eşitliklerinden oluşan denklemler (4.50) – (4.55) arasında gösterilmiştir.

$$c_1 r_{13} + r_{23} s_1 = -c_{23} c_4 s_5 - s_{23} c_5 \quad (4.50)$$

$$c_1 p_x + p_y s_1 = a_1 + a_2 c_2 - d_4 s_{23} - d_6 (s_{23} c_5 + c_{23} c_4 s_5) \quad (4.51)$$

$$c_1 p_x + p_y s_1 = a_1 + a_2 c_2 - d_4 s_{23} + d_6 (c_1 r_{13} + r_{23} s_1) \quad (4.52)$$

$$r_{33} = s_{23} c_4 s_5 - c_{23} c_5 \quad (4.53)$$

$$p_z = -a_2 s_2 - d_4 c_{23} - d_6 (c_{23} c_5 - s_{23} c_4 s_5) \quad (4.54)$$

$$p_z = -a_2 s_2 - d_4 c_{23} + d_6 r_{33} \quad (4.55)$$

(4.52), (4.50)'nin (4.51)'de , (4.55) de (4.53)'ün (4.54)'te yerine koyulması ile oluşturulmuştur. Daha sonra bu iki denklem düzenlenerek bilinenler sol tarafa alınmış ve η ve μ olarak isimlendirilmiştir. Daha sonra oluşan denklemlerin her iki tarafın karesi alınarak taraf tarafa toplanılmış ve (4.61) ve (4.62)'de verilen trigonometrik formüller uygulanmıştır.

$$c_1 p_x + p_y s_1 - a_1 - d_4 s_{23} + d_6 (c_1 r_{13} + r_{23} s_1) = \eta = a_2 c_2 - d_4 s_{23} \quad (4.56)$$

$$p_z - d_6 r_{33} = \mu = -a_2 s_2 - d_4 c_{23} \quad (4.57)$$

$$\eta^2 = d_4^2 s_{23}^2 + 2a_2 c_2 d_4 s_{23} + a_2^2 c_2^2 \quad (4.58)$$

$$\mu^2 = d_4^2 c_{23}^2 + 2d_4 c_{23} a_2 s_2 + a_2^2 s_2^2 \quad (4.59)$$

$$\eta^2 + \mu^2 = a_2^2 (s_2^2 + c_2^2) + d_4^2 (s_{23}^2 + c_{23}^2) - 2a_2 d_4 (s_{23} c_2 - c_{23} s_2) \quad (4.60)$$

$$\sin^2 x + \cos^2 x = 1 \quad (4.61)$$

$$\sin(x - y) = \sin(x) \cos(y) - \cos(x) \sin(y) \quad (4.62)$$

$$2a_2d_4\sin(\theta_2 + \theta_3 - \theta_2) = a_2^2 + d_4^2 - \eta^2 - \mu^2 \quad (4.63)$$

$$\sin \theta_3 = \frac{a_2^2 + d_4^2 - \eta^2 - \mu^2}{2a_2d_4} = \rho \quad (4.64)$$

$$\theta_3 = \text{atan2}(\rho, \sqrt{1 - \rho^2}) \vee \text{atan2}(\rho, -\sqrt{1 - \rho^2}) \quad (4.65)$$

(4.61) ve (4.62)'de gösterilen trigonometrik denklemler (4.60) üzerine uygulanmış ve denklem basitleştirilmiştir. Daha sonra sadeleştirilen eşitlikten θ_3 açısı bulunmuştur. θ_3 açısı bulunduktan sonra (4.11)'de verilen 0_6T dönüşüm matrisi ile $[{}^0_3T]^{-1}$ matrisi çarpılarak (4.66) elde edilmiştir.

$${}^0_3T = {}^0_1T(\theta_1){}^1_2T(\theta_2){}^2_3T(\theta_3) \quad (4.66)$$

$${}^0_3T^{-1} = \begin{bmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & -a_1c_{23} - a_2c_3 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & a_1s_{23} + a_2s_3 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.67)$$

$${}^0_3T^{-1} {}^0_6T = {}^3_4T(\theta_4){}^4_5T(\theta_5){}^5_6T(\theta_6) \quad (4.68)$$

$$\begin{bmatrix} c_1c_{23} & s_1c_{23} & -s_{23} & -a_1c_{23} - a_2c_3 \\ -c_1s_{23} & -s_1s_{23} & -c_{23} & a_1s_{23} + a_2s_3 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & -d_6c_4s_5 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 + d_6c_5 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & d_6s_4s_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.69)$$

(4.69)'un sol tarafında gösterilen matris çarpımının sonucundan gelen (1,4) ve (2,4) elemanlarının eşitliğin sağ tarafında bulunan matrisin aynı elemanlarına olan eşitliği kullanılarak (4.70) ve (4.71) eşitlikleri oluşturulmuştur.

$$c_1c_{23}c_{23}p_x + s_1c_{23}p_y - s_{23}p_z - a_1c_{23} - a_2c_3 = -d_6c_4s_5 \quad (4.70)$$

$$-c_1s_{23}p_x - s_1s_{23}p_y - c_{23}p_z + a_1s_{23} + a_2s_3 = d_4 + d_6c_5 \quad (4.71)$$

(4.70) ve (4.71) sırasıyla c_{23} ve s_{23} ile çarpılarak (4.72) ve (4.73) eşitliklerine ulaşılır. Daha sonra bu eşitlikler (3.101) – (3.103) arasındaki eşitliklerde taraf tarafa çıkarılıp sadeleştirilerek (3.103)'te gösterilen eşitliğe ulaşılır.

$$c_1 c_{23}^2 p_x + s_1 c_{23}^2 p_y - c_{23} s_{23} p_z - a_1 c_{23}^2 - a_2 c_{23} c_3 = -d_6 c_4 s_5 \quad (4.72)$$

$$-c_1 s_{23}^2 p_x - s_1 s_{23}^2 p_y - c_{23} s_{23} p_z + a_1 s_{23}^2 + a_2 s_{23} c_3 = d_4 s_{23} + d_6 s_{23} c_5 \quad (4.73)$$

$$\begin{aligned} p_x c_1 (c_{23}^2 + s_{23}^2) + p_y s_1 (c_{23}^2 + s_{23}^2) - a_1 (c_{23}^2 + s_{23}^2) - a_2 (c_{23} c_3 + s_{23} c_3) \\ = -d_6 (c_4 s_5 + s_{23} c_5) - d_4 s_{23} \end{aligned} \quad (4.74)$$

$$c_{23} c_3 + s_{23} c_3 = c_2 c_3^2 - s_2 s_3 c_3 + s_2 s_3 c_3 + c_2 s_3^2 = c_2 (c_3^2 + s_3^2) = c_2 \quad (4.75)$$

$$p_x c_1 + p_y s_1 - a_1 - a_2 c_2 = -d_6 (c_{23} c_4 s_5 + s_{23} c_5) - d_4 s_{23} \quad (4.76)$$

(4.50)'de gösterilen ifade (4.76)'da yerine koyulup s_{23} ifadesi de (4.16)'da gösterilen şekilde tekrar yazıldığında (4.77)'de gösterilen eşitlik elde edilmiştir.

$$p_x c_1 + p_y s_1 - a_1 - d_6 (c_1 r_{13} + r_{23} s_1) = d_4 (s_2 c_3 + c_2 s_3) + a_2 c_2 \quad (4.77)$$

(4.77)'de elde edilen eşitlik (4.78)'de gösterilen formatta düzenlenerek (4.79) – (4.81) arasında gösterilen parametreler elde edilir.

$$A \sin(\theta_2) + B \cos(\theta_2) + C = 0 \quad (4.78)$$

$$A = \frac{d_4 c_3}{p_x c_1 + p_y s_1 - a_1 - d_6 (c_1 r_{13} + r_{23} s_1)} \quad (4.79)$$

$$B = \frac{d_4 s_3 - a_2}{p_x c_1 + p_y s_1 - a_1 - d_6 (c_1 r_{13} + r_{23} s_1)} \quad (4.80)$$

$$C = 1 \quad (4.81)$$

(4.78) tekrar düzenlenerek $\cos(\theta_2)$ yerine $\sqrt{1 - \sin^2(\theta_2)}$ yazılıp her iki tarafın karesi alınarak (4.83)'te gösterilen ikinci dereceden tek bilinmeyenli bir eşitlik elde edilmiştir. Daha sonra bu denklem $\sin(\theta_2)$ için çözümlenerek θ_2 açısı bulunmuştur.

$$A \sin(\theta_2) + 1 = -B \sqrt{1 - \sin^2(\theta_2)} \quad (4.82)$$

$$\sin^2(\theta_2) (A^2 + B^2) + 2A \sin(\theta_2) + 1 - B^2 = 0 \quad (4.83)$$

$$\sin(\theta_2) = \frac{-2A \pm \sqrt{4A^2 - 4(A^2 + B^2)(1 - B^2)}}{2(A^2 + B^2)} \quad (4.84)$$

$$\sin(\theta_2) = \frac{-A \pm B\sqrt{A^2 + B^2 - 1}}{A^2 + B^2} = \sigma_{1,2} \quad (4.85)$$

$$\theta_2 = \text{atan2}(\sigma, \sqrt{1 - \sigma^2}) \vee \text{atan2}(\sigma, -\sqrt{1 - \sigma^2}) \quad (4.86)$$

(4.84)'te θ_2 için iki farklı değer gösterilmiştir. Bunun nedeni eşitlik (4.83)'ün iki kökü olmasıdır. Bu köklerden biri (4.82)'de gösterilen eşitliği sağlamaktadır. Bu nedenle (4.84)'te hesaplanan θ_2 açısı (4.82)'de doğrulanarak seçilmiştir.

θ_2 açısı bulunduğundan sonra (4.69)'da gösterilen matrislerin (1,3) ve (1,4) elemanlarının eşitliği kullanılarak (4.87) ve (4.88) elde edilmiştir.

$$c_1 c_{23} r_{13} + s_1 c_{23} r_{23} - s_{23} r_{33} = -c_4 s_5 \quad (4.87)$$

$$-s_1 r_{13} + c_1 r_{23} = s_4 s_5 \quad (4.88)$$

Bu eşitliklerden θ_4 açısının bulunması (4.89)'te gösterilmiştir. θ_4 açısının değerinin hesaplanabilmesi için (4.87) ve (4.88) taraf tarafa bölünerek sadeleştirme yapılmıştır. θ_4 açısının hesaplanabilmesi için s_5 değerinin 0'a eşit olmaması gerekmektedir.

$$\theta_4 = \text{atan2}((-s_1 r_{13} + c_1 r_{23}), (-c_1 c_{23} r_{13} - s_1 c_{23} r_{23} + s_{23} r_{33})) \quad (4.89)$$

(4.12)'de verilen eşitliğin her iki tarafının bilinen açıların dönüşüm matrislerinin tersi olan ve (4.90)'da gösterilen ${}^0_4T^{-1}$ matrisi ile çarpılması ile elde edilen matris (4.91)'de gösterilmiştir.

$${}^0_4T = {}^0_1T(\theta_1) {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) \quad (4.90)$$

$${}^0_4T = \begin{bmatrix} s_1 s_4 + c_{23} c_1 c_4 & s_1 c_4 + c_{23} c_1 s_4 & -s_{23} c_1 & {}^0_4p_x \\ -c_1 s_4 + c_{23} s_1 c_4 & -c_1 c_4 + c_{23} s_1 s_4 & -s_{23} s_1 & {}^0_4p_y \\ -s_{23} c_4 & s_{23} s_4 & -c_{23} & {}^0_4p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.91)$$

$${}^0_4T^{-1} = \begin{bmatrix} c_{23} c_1 c_4 + s_1 c_4 & -c_1 s_4 + c_{23} s_1 c_4 & -s_{23} c_4 & {}^0_4p'_x \\ s_1 c_4 - c_{23} c_1 s_4 & -c_1 c_4 - c_{23} s_1 s_4 & s_{23} s_4 & {}^0_4p'_y \\ -s_{23} c_1 & -s_{23} s_1 & -c_{23} & {}^0_4p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.92)$$

$${}^0T^{-1}{}^0T = {}^4T(\theta_5){}^5T(\theta_6) \quad (4.93)$$

$${}^0T^{-1}{}^0T = \begin{bmatrix} c_{23}c_1c_4 + s_1c_4 & -c_1s_4 + c_{23}s_1c_4 & -s_{23}c_4 & {}^0p'_x \\ s_1c_4 - c_{23}c_1s_4 & -c_1c_4 - c_{23}s_1s_4 & s_{23}s_4 & {}^0p'_y \\ -s_{23}c_1 & -s_{23}s_1 & -c_{23} & {}^0p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_5c_6 & -c_5s_6 & -s_5 & -d_6s_5 \\ s_6 & c_6 & 0 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & d_6c_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.94)$$

(4.94)'te gösterilen matrislerin (1,3), (3,3) (2,1) ve (2,2) elemanlarının eşitliğinden (4.95) – (4.98) arasındaki eşitlikler çıkarılmıştır. Bu eşitlikler sonucunda (4.99) ve (4.100)'de sırasıyla θ_5 ve θ_6 açıları da bulunarak robotun bütün eklem açıları bulunmuştur.

$$r_{13}(c_{23}c_1c_4 + s_1c_4) + r_{23}(-c_1s_4 + c_{23}s_1c_4) - r_{33}(s_{23}c_4) = -s_5 \quad (4.95)$$

$$r_{13}(-s_{23}c_1) + r_{23}(-s_{23}s_1) - r_{33}(c_{23}) = c_5 \quad (4.96)$$

$$r_{11}(s_1c_4 - c_{23}c_1s_4) + r_{21}(-c_1c_4 - c_{23}s_1s_4) + r_{31}(s_{23}s_4) = s_6 \quad (4.97)$$

$$r_{12}(s_1c_4 - c_{23}c_1s_4) + r_{22}(-c_1c_4 - c_{23}s_1s_4) + r_{32}(s_{23}s_4) = c_6 \quad (4.98)$$

$$\theta_5 = \text{atan2}(s_5, c_5) \quad (4.99)$$

$$\theta_6 = \text{atan2}(s_6, c_6) \quad (4.100)$$

Bu aşamadan sonra [70]'deki çalışma kullanılarak hesaplanan eklem açılarından çoklu değere sahip olan θ_1 ve θ_3 açıları, robotun insan kinematiğine en uygun yönelimle belirtilen uzay koordinatlarına ulaşacağı şekilde seçilerek robota gönderilmeye hazır altı eklem açısı elde edilmiştir.

Bu açılardan son üçü, $\theta_4 - \theta_6$, yönelimi belirleyen açılardır. Kinect'ten alınan derinlik haritasında bileğin yönelimi ile ilgili herhangi bir hesaplama yapılmadığından operatörün çalışma uzayı ile robotun çalışma uzayı Bölüm 3.3'e birbirine eşitlenirken operatörün kolu robotun uç işlemcisi açık duracak şekilde genişletilmiştir. Bu işlem (3.23)'te gösterilmiştir. Bu nedenle ters kinematik çözümler yapılırken d_4 ve d_6 parametreleri de aynı şekilde modifiye edilmiştir. Robotun istenilen uç işlemci pozisyonuna giderken herhangi bir bilek yönelimi göstermemesi için Çizelge 4.1'de gösterilen d_6 parametresi sıfıra eşitlenmiş, d_4

parametresi de d_6 parametresi kadar artırılarak (3.23)'te gösterilen parametre genişletme ile eşleşecek bir ters kinematik çözüm yapılması sağlanmıştır.

Bu yöntem 3.4 bölümünde anlatılan insan-robot kolu eşlemesi yapılarak oluşturulan homojen dönüşüm matrisinin her örneği için eş zamanlı olarak tekrarlanmaktadır. Elde edilen sonuçlar, diğer yöntemlerin sonuçları ile beraber 5. Bölümde gösterilmektedir.

4.3 Yapay Sinir Ağı Yöntemi

Yapay sinir ağları genellikle kapalı form eşitliklerle modellenmesi zor olan, karmaşık ve doğrusal olmayan sistemleri modellemek için kullanılan yapılardır. Bu yapılarda sistemi modellemek için tüme varım yöntemi yerine tümden gelim yöntemi kullanılmaktadır. Modellenmek istenilen sistemin yeterli sayıdaki girdi ve çıktı kümeleri kullanılarak farklı eğitim algoritmaları ile yapay sinir ağları oluşturmak mümkündür.

Bu çalışmada eğitilen yapay sinir ağı için Matlab Neural Network Fitting (NNF) aracı kullanılmıştır [73]. Kinect'ten alınan operatör eklem açıları ve operatör bilek eklemi pozisyonu ile robotun eklem açılarının oluşturduğu veri seti üzerinde eğri uydurma yapılarak 4.2'da anlatılan eş zamanlı ters kinematik çözüme yöntemine alternatif olabilecek bir çözüm geliştirilmiştir.

Matlab NNF eğri uydurma amaçlı yapay sinir ağı oluşturmak için kullanılmaktadır. Literatürde kullanılan birçok yapay sinir ağı türü bulunmaktadır, ancak bu türlerinin hepsinin temel prensipleri benzerdir. Sinir ağında bulunan her bir nöron girdi alır, bu girdileri işler ve çıktı verir. Her bir nöron en az başka bir nöron ile bağlantılıdır ve bu bağlantı ağırlık katsayısı ile belirtilir. Yapay sinir ağlarının eğitilmesi için iki temel yöntem bulunmaktadır. Bunlar denetimli eğitim ve denetimsiz eğitim yöntemleridir. Denetimli eğitim yönteminde yapay sinir ağına ulaşılması istenen çıktılar verilir, nöronlar arasındaki ağırlık katsayıları eğitim sırasında bu çıktılara en yakın şekilde ulaşmak için güncellenir. Denetimsiz eğitim yönteminde ise ulaşmak istenilen çıktı bilinmez, sisteme bir model sunulur ve belli sayıda tekrardan sonra istikrarlı bir duruma gelmesi beklenir.

Çok katmanlı ileri beslemeli sinir ağları denetimli eğitim yöntemi ile eğitilir ve literatürde en çok kullanılan sinir ağı türlerinden biridir. Bu tür bir sinir ağında nöronlar katmanlara yayılmış bir şekilde bulunurlar. İlk katman girdi katmanı olarak adlandırılır. Bu katmanı bir veya birden çok gizli katman takip eder. Son olarak da çıktı katmanı bulunur. Bir katmanda bulunan bütün nöronlar bir sonraki katmanda bulunan nöronların hepsi ile bağlantılıdır. Farklı katmanlarda bulunan i ve j nöronları arasında bağlantı nöronlar arasında bulunan ağırlık katsayısı, w_{ij} , ve i nöronunun eşik değeri katsayısı, v_i , ile karakterize edilir. Yapay sinir ağının eğitimi nöronlar arasında bulunan ağırlık katsayıları ve eşik değerlerinin değiştirilerek, sinir ağının performans fonksiyonunun optimize edilmesi ile yapılır. İleri beslemeli sinir ağlarında kullanılan performans fonksiyonlarından biri ortalama karesel hatanın hesaplanmasıdır. Bu performans fonksiyonunun optimize edilmesi sırasında herhangi bir sayısal optimizasyon algoritması kullanılabilir. Bu optimizasyon algoritmalarının bazıları sinir ağı performans fonksiyonunun nöronların ağırlık katsayılarına göre gradyanlarını kullanırken bir kısmı ise sinir ağının nöronların ağırlık katsayılarına göre hatalarının Jacobian matrisini kullanır [74].

Gradyan ve Jacobian matrisi hesaplamaları geri yayılım algoritması kullanılarak yapılmaktadır. Bu yöntemde hata fonksiyonunun gradyan değeri sinir ağındaki nöronların ağırlık katsayısına göre hesaplanır. Daha sonra sayısal bir optimizasyon algoritması kullanılarak hata fonksiyonu minimize edilmeye çalışılır. Bu minimizasyon sırasında her iterasyonda ağırlık katsayıları güncellenerek işlem belirli bir hata değeri ya da iterasyon sayısına kadar devam eder.

Bu optimizasyon algoritmalarından bazılarının gerçeklemeleri Matlab NNF aracı tarafından sunulmaktadır. Levenberg-Marquardt optimizasyon algoritması bunlardan biridir. Levenberg-Marquardt algoritması ikinci dereceden eğitim hızına Hessain matrisini hesaplamadan ulaşmaya çalışır. Performans fonksiyonunun kareler toplamı türünden belirtildiği sinir ağlarında Hessain matrisi (4.99)'da gösterildiği gibi varsayılabilir.

$$H = J^T J \quad (4.101)$$

$$g = J^T e \quad (4.102)$$

(4.102)'de gradyan hesaplaması sinir ağının hatalarının nöronların ağırlık katsayılarına göre birinci dereceden türevinden oluşan Jacobian matrisi ve sinir ağının hatalarını içeren vektör, e , cinsinden verilmiştir. Levenberg-Marquardt algoritması (4.101)'de gösterilen Hessain matrisi yaklaşımını kullanarak (4.103)'te gösterilen güncelleme fonksiyonunu kullanır.

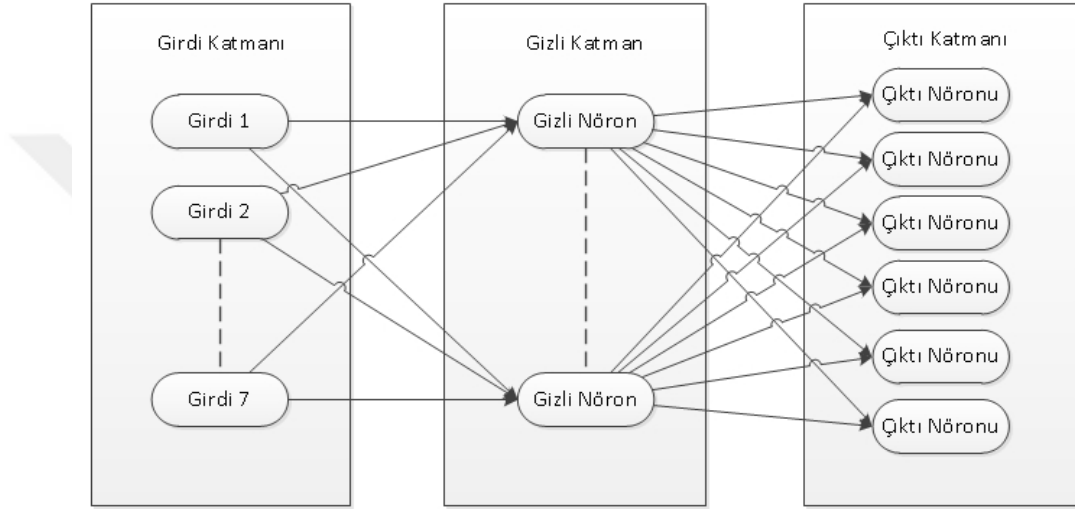
$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (4.103)$$

Burada skalar değer olan μ 0 olduğunda güncelleme fonksiyonu Hessain matrisinin (4.101)'de gösterilen varsayımını kullanan Newton yöntemine dönüşür [75]. Bu skalar büyük olduğunda ise küçük adım aralıkları kullanan gradyan düşüşü metoduna dönüşür. Newton yönteminin daha hızlı ve hassasiyetinin daha yüksek olduğu göz önüne alındığında, (4.103)'ü kullanırken amaç olabildiğince hızlı bir şekilde Newton yöntemine yaklaşılmaktadır. Bu nedenle optimizasyonu yapılan sinir ağı performans fonksiyonunun her iterasyondaki düşüşü ile beraber μ değeri azaltılarak Newton yöntemine yaklaşılmaktadır.

Matlab NNF aracı içerisinde bulunan algoritmalarından en hızlı çalışan ve Matlab tarafından önerilen performans fonksiyonu sayısal optimizasyon algoritması yukarıda anlatılan Levenberg-Marquardt yöntemidir. Bu yöntemin aynı zamanda Matlab içerisinde yüksek verimli bir gerçekleştirilmesi bulunmaktadır. Bu nedenle yapay sinir ağının eğitimi sırasında bu algoritma kullanılmıştır.

Yapay sinir ağının eğitilmesi için oluşturulan veri seti dördü operatör kolunun eklem açıları, üçü operatör bilek ekleminin uzaysal koordinatı olmak üzere yedi girdiden ve üç robot eklemi açısı içeren çıktıdan oluşan örneklerden oluşmaktadır. Bu veri seti Bölüm 3.3'te anlatılan insan kolu kinematiği hesaplamalarına uyan örneklerden oluşturulmuştur. Bu örneklerin her biri için operatör eklem açıları ve dönüşüm matrisi hesaplanmış ve 4.2'da gösterilen ters kinematik çözümler uygulanarak robotun eklem açılarına ulaşılmıştır. Ulaşılan eklem açılarından son üçü yönelim ile ilgili olduğu için ve Kinect'ten bilek yönelimi ile ilgili veri alınmadığı için bu açılar robotun uç işlemcisinin açık pozisyonda duracağı şekilde sifira eşitlenmiştir. Bu nedenle yapay sinir ağı oluştururken sadece ilk üç robot ekleminin açısı çıktı olarak kabul edilmiştir.

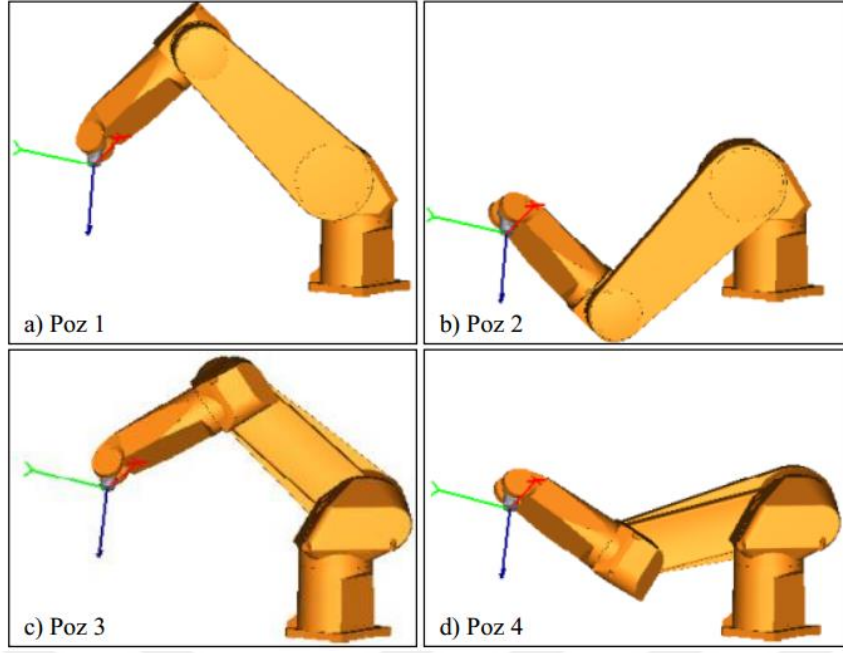
Oluşturulan veri seti üzerinde Matlab'ın NNF aracı kullanılarak 2 katmanlı ileri beslemeli sinir ağı yaratılmıştır. Bu sinir ağının ilk katmanı sigmoid gizli nöronlardan oluşmaktadır. İkinci katman ise çıktı katmanıdır ve doğrusal çıktı nöronlarından oluşmaktadır. Çok katmanlı yapısı sayesinde bu sinir çok boyutlu eşleme problemlerinin çözümünde yaygın olarak kullanılmaktadır. Burada dikkat edilmesi gereken noktalar çok boyutlu eşleme için yeterince veri olması ve ilk katman olan gizli katmanında yeterince nöron olmasıdır. İki katmanlı ileri beslemeli sinir ağı yapısı Şekil 4.2'de gösterilmiştir.



Şekil 4.2- İki katmanlı ileri beslemeli sinir ağı mimarisi.

Bölüm 4.2'da belirtildiği gibi θ_1 ve θ_3 açısının birden fazla değer alabilmesi nedeniyle robotun belirtilen uç işlemci pozisyona gitmesi için kullanabileceği dört farklı konfigürasyon bulunmaktadır. Bu konfigürasyonlar Şekil 4.3'te gösterilmiştir.

Yapay sinir ağı eğitilmeden önce Şekil 4.3'teki gibi oluşabilecek farklı çözümlerin veri setinden ayıklanması gerekmektedir. Aksi takdirde yapay sinir ağı çıktısından alınacak olan robot açıları aynı girdi için birden fazla çözüm oluşma riski vardır. Böyle bir durumda yapay sinir ağından alınan robot açıları, girdi olarak verilen kol açılarındaki küçük değişimlerde bile büyük farklılıklar gösterebilir. Bu durum hem yapay sinir ağının eğitimi sırasında, hem de daha sonra kullanımı sırasında problem oluşturacaktır.



Şekil 4.3- Aynı uç işlemci pozisyonu için oluşan farklı pozlar [70].

Bu durum göz önünde bulundurularak eğitim için oluşturulan veri seti üzerinde bazı tanımlamalara göre filtreler oluşturulmuştur. Tanımlamalar yapılırken aşağıdaki eşitliklerde belirtilen işlem şartları göz önünde bulundurulmuştur.

$$J_{6,z} \geq J_{3,z} \quad (4.104)$$

$$J_{6,z} \geq 0 \text{ mm} \quad (4.105)$$

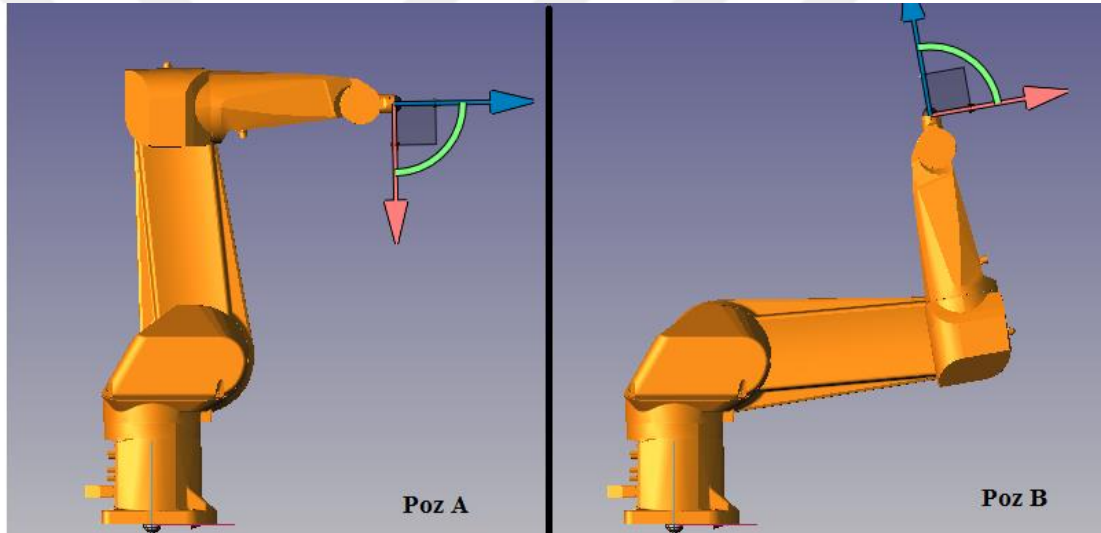
$$J_{3,z} \geq -360 \text{ mm} \quad (4.106)$$

$$\text{sign}(\theta_2) \neq \text{sign}(\theta_3) \quad (4.107)$$

$$J_{3,z} \leq 800 \text{ mm} \wedge (J_{6,z} - J_{3,z}) \geq 0 \quad (4.108)$$

Eşitliklerde gösterilen $J_{i,z}$ parametresi robotun i eklemine z eksenindeki pozisyonunu belirtmektedir. Buna göre (4.104)'te robotun uç işlemcisinin z eksenindeki pozisyonunun, dirsek eklemine z eksenindeki pozisyonuyla olan ilişkisini göstermektedir. Bu eşitlik robotun her zaman dirsek aşağıda çalışma konfigürasyonlarını kullanması için tanımlanmıştır. Bu eşitliğe göre değerlendirildiğinde Şekil 4.3'te gösterilen Pozisyon 1 ve Pozisyon 3 konfigürasyonları veri setinden çıkarılmaktadır. (4.105) uç işlemcinin z eksenindeki pozisyonunun pozitif olması gerektiğini belirtmektedir. Bu kontrol robotun çalışması

esnasında yere çarpmasını engellemek için konulmuştur. Aynı şekilde (4.106) dirsek ekleminin z ekseninde limitini belirlenerek ve dirsek ekleminin yere çarpmasını engellemek için konulmuştur. (4.107) robotun dirsek aşağı konfigürasyonda çalışması için tanımlanmıştır. Yapılan deneylerde Poz 1 ve Poz 3 gibi konfigürasyonlarda robotun θ_2 ve θ_3 açılarının aynı işarete sahip olduğu görülmüştür. Bu nedenle (4.108) tanımlanarak çözüm setinden aynı işarete sahip θ_2 ve θ_3 açılarını içeren çözümler çıkarılmıştır. (4.108)'de gösterilen eşitlik model üzerinde olarak denemeler sırasında karşılaşılan bir durumdan yola çıkılarak oluşturulmuştur. Bazı verilerin (4.104) – (4.107) arasındaki eşitlikleri sağlamasına rağmen hala istenilmeyen pozlara gidebildiği gözlemlenmiştir. Bu gözlem Şekil 4.4'te gösterilmiştir.



Şekil 4.4- Aynı uç işlemci noktasını veren farklı pozlar.

Şekil 4.4'te görüleceği gibi Poz A'da robot hala dirsek yukarı konfigürasyonda hareket etmektedir. Ancak seçilmek istenilen konfigürasyon Poz B'de gösterilen konfigürasyondur. Bu nedenle (4.108) tanımlanarak istenilmeyen pozlar veri seti arasından elenmiştir.

(4.104) - (4.108) arasında belirtilen eşitlikler iki farklı veri setine uygulanmıştır. İlk veri seti 2542 örnekten oluşmaktadır. Filtreler uygulandıktan sonra geçerli veri sayısı 1438'e düşmüştür. İkinci veri seti ise 5383 örnek içermektedir. Bu veri seti filtrelendikten sonra 2999 geçerli veri içerir hale gelmiştir. Bu çözüm setleri yapay sinir ağının eğitiminde kullanılmıştır. Bu veri setinin %70'i eğitim, %15'i test, kalan

%15'i de validasyon verisi olacak şekilde ayrımı yapılmıştır. Şekil 4.2'de gösterilen mimaride gizli katmandaki nöron sayıları değiştirilerek oluşturulan yapay sinir ağlarının regresyon ve hata histogramları incelenmiştir. Daha geniş veri seti ile eğitilen ve 70 nöronlu gizli katmandan oluşan yapay sinir ağı aynı sayıda gizli nöron içeren düşük örnek sayısı ile eğitilmiş sinir ağına göre testlerde daha az hata ile uç işlemci noktasına ulaşan eklem açılarını vermiştir. Sinir ağlarının performans karşılaştırmasını gösteren testler 5. Bölüm'de anlatılmıştır.



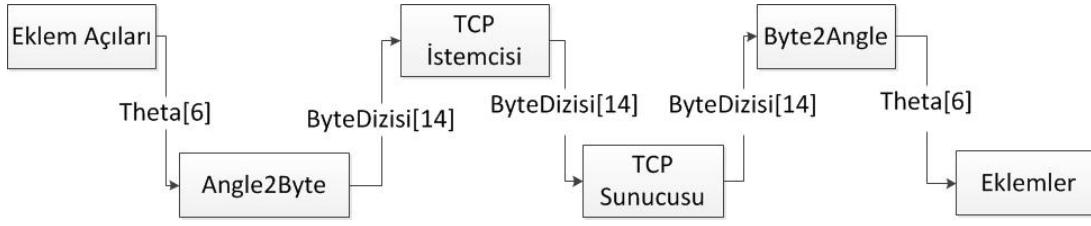
5. SONUÇLAR VE KARŞILAŞTIRMA

Bu bölümde, Bölüm 4'te açıklanan üç farklı teleoperasyon yöntemi ile elde edilen sonuçlar gösterilerek karşılaştırma yapılmıştır. Öncelikli olarak görsel karşılaştırmalar için operatörün dört hareketi tanımlanmış ve bu hareket sonucunda oluşan robot açıları her yöntem için hesaplanarak robota gönderilmiştir. Diğer bölümde ise ters kinematik çözümü ve yapay sinir ağı yöntemleri doğruluk ve zaman performanslarına göre birbirleri ile karşılaştırılmıştır.

Simülasyon ortamını oluşturmak için Stäubli Robotics Suite 2016 emülatörü kullanılmıştır. Bu emülatör simülasyon programına bağlanarak kontrolcü gibi davranmaktadır. Normalde robotu kontrol etmek için gereken parametreler emülatöre verildiği zaman Stäubli Robotics Suite 2016 üzerinden robotun hareketleri görülmektedir.

Robot kontrolcüsü Stäubli 'nin oluşturduğu bir dil olan VAL3 robot programlama dili kullanılarak programlanmaktadır. Kontrolcüye yüklenen uygulamalar bu dil ile geliştirilerek kontrol sağlanmaktadır.

Emülatöre bulunan açıların geçirilmesi ağ üzerinden yapılmıştır. TCP/IP protokolü kullanılarak emülatör ile bağlantı kurulmuştur. Bunun için emülatör TCP sunucusu, Matlab yazılımı ise TCP istemcisi gibi konfigüre edilerek kullanılmıştır. Emülatör üzerinde istemcinin bağlantı bekleyeceği port açılmış ve buraya gelen verileri çözümleyecek bir uygulama yazılmıştır. Burada uygulamaya gelen veriler byte cinsinden gelmektedir. Bu nedenle gelen byte verileri karşılığı olan sayı değerlerine çevrilerek ilgili ekleme gönderilmektedir. Oluşturulan istemci/sunucu mimarisi Şekil 5.1 'de gösterilmiştir.



Şekil 5.1- Açık değerlerin iletmek için oluşturulan istemci/sunucu mimarisi.

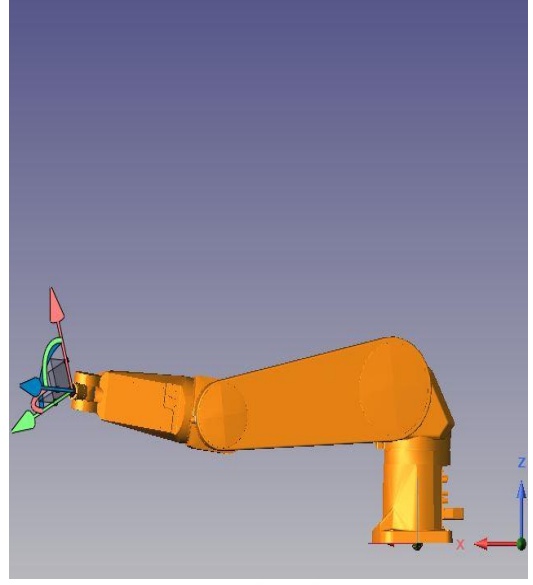
Şekil 5.1 ile gösterilen yapı kurularak Kinect'ten alınan veriler ile hesaplanan operatör kolu açı değerleri Bölüm 4'te anlatılan yöntemlerin her biri ile hesaplanarak robota aktarılmıştır.

5.1 Pozların Görsel Karşılaştırılması

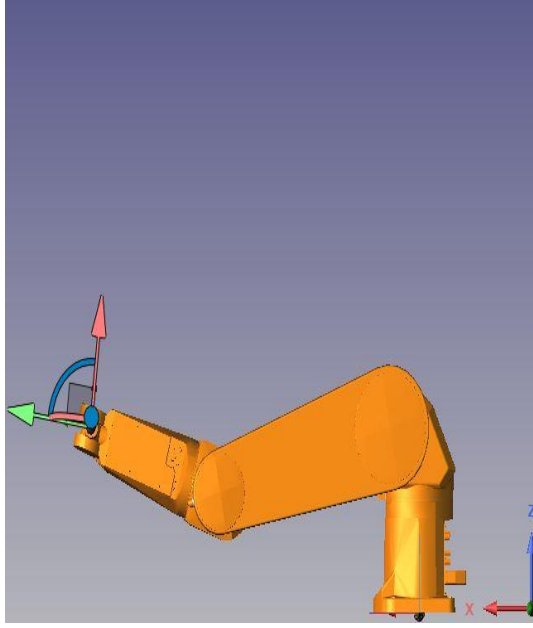
Bu bölümde farklı uzaktan kontrol yöntemlerinin operatör hareketlerine verdikleri tepkiler görsel olarak gösterilmiştir. Bunun için oluşturulan operatör hareketlerin derinlik kamerasından alınmış ve üzerlerine iskelet haritası eklenmiş görselleri ile farklı teleoprasyon yöntemleri kullanılarak elde edilen sonuçlar dört farklı operatör figürü için Şekil 5.2 – Şekil 5.5 arasında gösterilmiştir. Operatör pozisyonlarında z eksenini kamera doğrultusunda kameradan uzaklaştıkça pozitif değer almaktadır. Simülasyonda ise derinlik y eksenindedir ve ekrandan içeri doğru gidildikçe pozitif değer almaktadır. Bu nedenle aşağıdaki şekillerde gösterilecek üç ve dört numaralı figürlerin robot ve operatör pozisyonları arasında yönelim farkı vardır. Operatör tarafından kameraya doğru yapılan bir hareket robot tarafından ekrandan içeri doğru yapılmaktadır. Elde edilen sonuçlarda ilk şekil operatörün yaptığı hareketin derinlik kamerası tarafından algılanarak iskelet modeli üzerine oturtulmuş halini göstermektedir. Diğer şekiller farklı teleoperasyon yöntemleriyle alınan çıktıları göstermektedir.



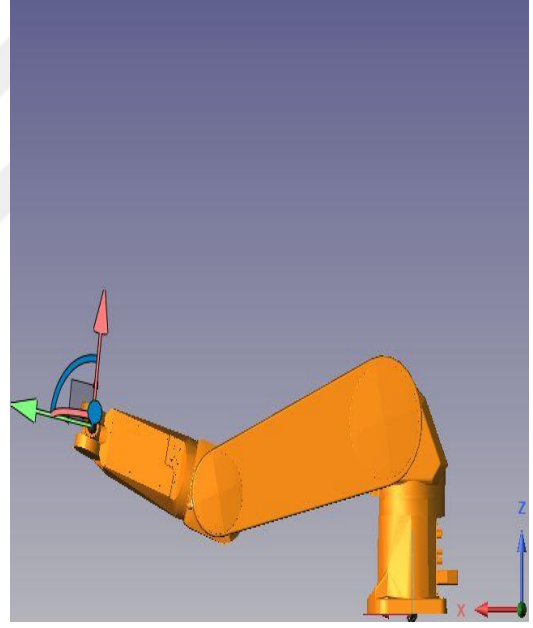
(a)



(b)

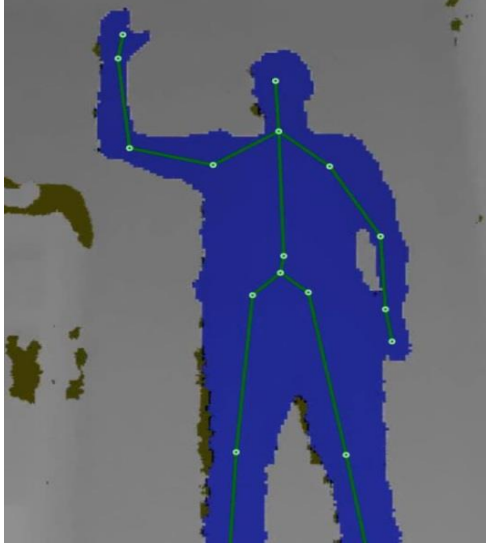


(c)

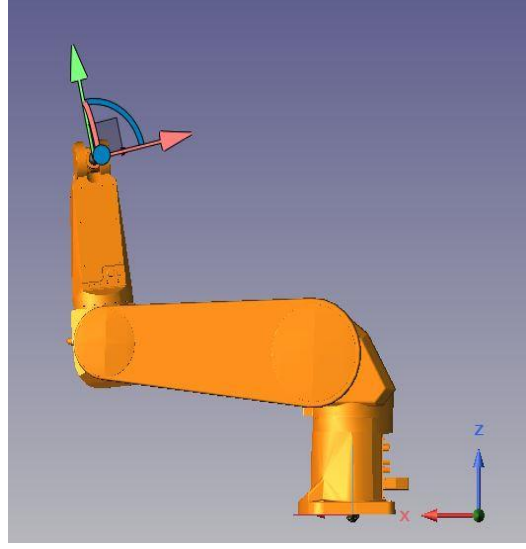


(d)

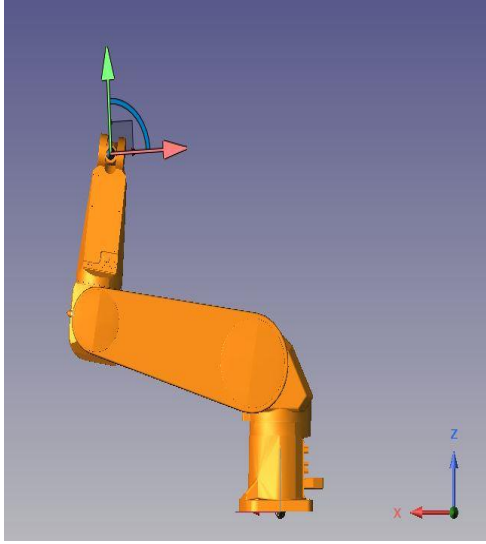
Şekil 5.2- Poz 1: (a) Operatör, (b) Direkt besleme yöntemi, (c) Ters kinematik çözümü, (d) Yapay sinir ağı



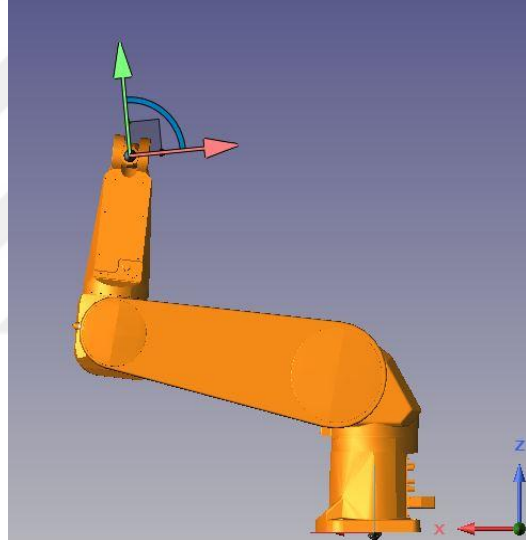
(a)



(b)



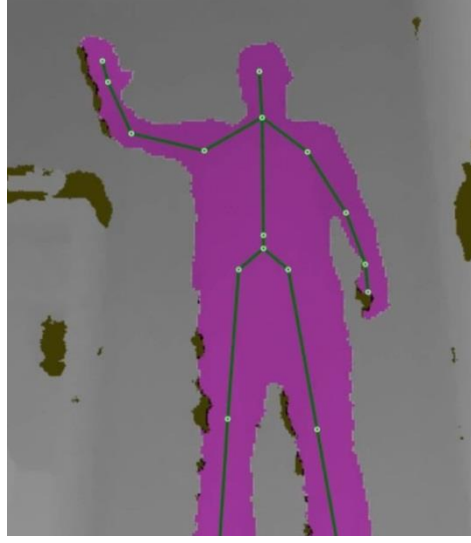
(c)



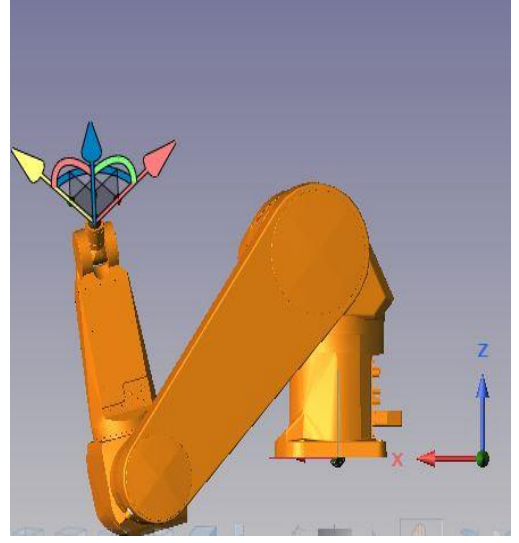
(d)

Şekil 5.3- Poz 2: (a) Operatör, (b) Direkt besleme yöntemi, (c) Ters kinematik çözümü, (d) Yapay sinir ağı

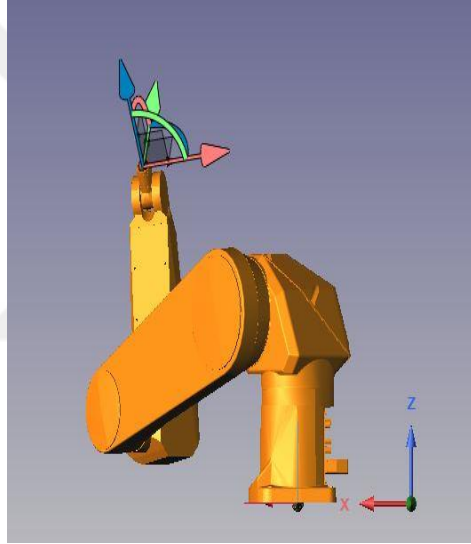
Şekil 5.2 ve Şekil 5.3'te gösterilen pozlar, bilek eklemleri hariç düşünüldüğünde, yalnızca üç eklem açısı ile kontrol edilebilmektedir. Bu nedenle üç yöntem de birbirlerine yakın sonuçlar vermişlerdir. Uç işlemci pozisyonu bu pozlar için operatör bileğini takip etmektedir. Ancak operatörün, θ_2 ve θ_3 açıları 90 dereceden farklı olacak şekilde bir poz vermesi durumunda direkt besleme yöntemi başarılı bir sonuç veremeyecektir. Bu pozlar Şekil 5.4 ve Şekil 5.5'te gösterilmiştir.



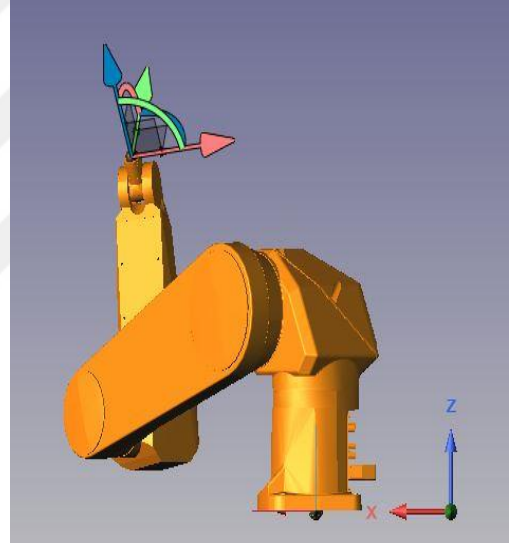
(a)



(b)



(c)

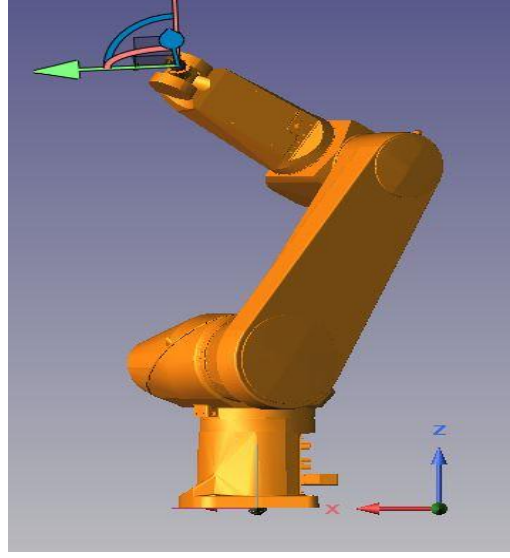


(d)

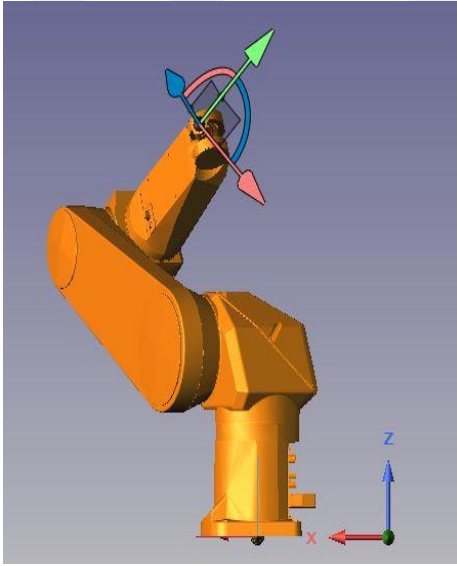
Şekil 5.4- Poz 3: (a) Operatör, (b) Direkt besleme yöntemi, (c) Ters kinematik çözümü, (d) Yapay sinir ağı



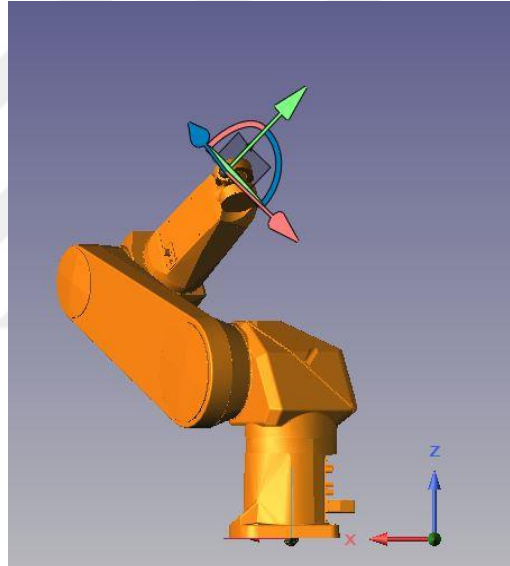
(a)



(b)



(c)



(d)

Şekil 5.5- Poz 4: (a) Operatör, (b) Direkt besleme yöntemi, (c) Ters kinematik çözümü, (d) Yapay sinir ağı

Daha önce 4.1 bölümünde direkt besleme yöntemi için insan kolu ve robotun eklemlerinin eşlenmesi yapılmıştır. Bu eşleme esnasında insan kolunu omuzdan bileğe kadar temsil eden açılar kullanılmıştır, bilek hareketlerini temsil eden son üç eklem açısı bu çözümün dışarısında bırakılmıştır. İnsan kolu ve robot kolu farklı kinematik zincirlere sahip olduğundan birebir eşleme yapılamamış, insan kolunun dirsek eklemindeki eklemden dışarı doğru olan eksen üzerindeki hareketi robot

üzerinde herhangi bir ekleme denk gelmediği gözlemlenmiştir. Ayrıca bu yöntemde birebir açı eşlemesi yapıldığı için ve insan kolunun kinematiğinde bilek eklemının pozisyonu dört eklem açısı ile belirlenirken, robotta buna denk gelen uç işlemci pozisyonunun üç eklem açısı ile belirlenmesinden dolayı son işlemcinin pozisyonu karşılaştırma parametresi olarak alınmamış, sadece poz benzerliğine bakılmıştır.

Ters kinematik çözümü ve yapay sinir ağı yöntemleri sonucunda oluşan pozisyonların birbirlerine çok yakın olduğu gözlemlenmiştir. Bu yöntemlerde robotun uç işlemci pozisyonu ile operatörün sağ el bileğinin robot çalışma uzayına eşlenmiş pozisyonunun birbirlerini takip ettikleri gözlenmiştir. Robot kolun kinematiği ile insan kolunun kinematiği farklı olmasına rağmen çözümlerin baz noktası uç işlemci pozisyonu olduğu için direkt besleme yönteminde olduğu gibi farklı pozisyonlarla karşılaşılmamıştır. Bu nedenle yapay sinir ağı yöntemi ve ters kinematik yöntemi uzaktan operasyon için direkt besleme yöntemine göre daha iyi sonuç veren yöntemlerdir. Bu yöntemler 5.2’de doğruluk ve zaman performanslarına göre birbirleri ile de karşılaştırılmıştır.

5.2 Ters Kinematik Yöntemi ve Yapay Sinir Ağı Yöntemlerinin Performans Karşılaştırmaları

Bu bölümde ters kinematik çözümleri ile yapay sinir ağı kullanılarak elde edilen teleoperasyon yöntemi zaman ve doğruluk yönleri ele alınarak karşılaştırılmıştır. Ters kinematik çözümleri uç işlemci pozisyonu ve oryantasyonunu kullanarak eklem açılarını bulduğu için robotun uç işlemci pozisyonu ve operatörün anlık uç işlemci pozisyonu birebir örtüşmektedir. Bu nedenle karşılaştırmalar yapılırken ters kinematik yöntemi ile elde edilen açılar ve ters kinematik yöntemine girdi olarak verilen operatör bilek konumu açı doğruluğu ve pozisyon doğruluğu karşılaştırmaları için referans değerler olarak kabul edilmiştir.

Ters kinematik çözüm yöntemi ve yapay sinir ağı performansları zaman bazında karşılaştırılması için bir test prosedürü geliştirilmiştir. Bu prosedürde 2000 örnekten oluşan girdi kümesi kullanılmıştır. Yapay sinir ağı için girdiler operatörün kol açıları iken ters kinematik çözümü yöntemi için girdiler 4x4 dönüşüm matrisi olarak verilmiştir.

Karşılaştırmanın sağlıklı yapılabilmesi için iki yöntem de işlem gücü aynı olan bilgisayar üzerinde test edilmiştir. Testler 10 kere tekrarlanarak ortalama değerler üzerinden karşılaştırma yapılmıştır. Test için kullanılan basamaklar Çizelge 5.1’de gösterilmektedir.

Çizelge 5.1- Ters kinematik ve yapay sinir ağı çözümlerinin hesaplama performansı testleri.

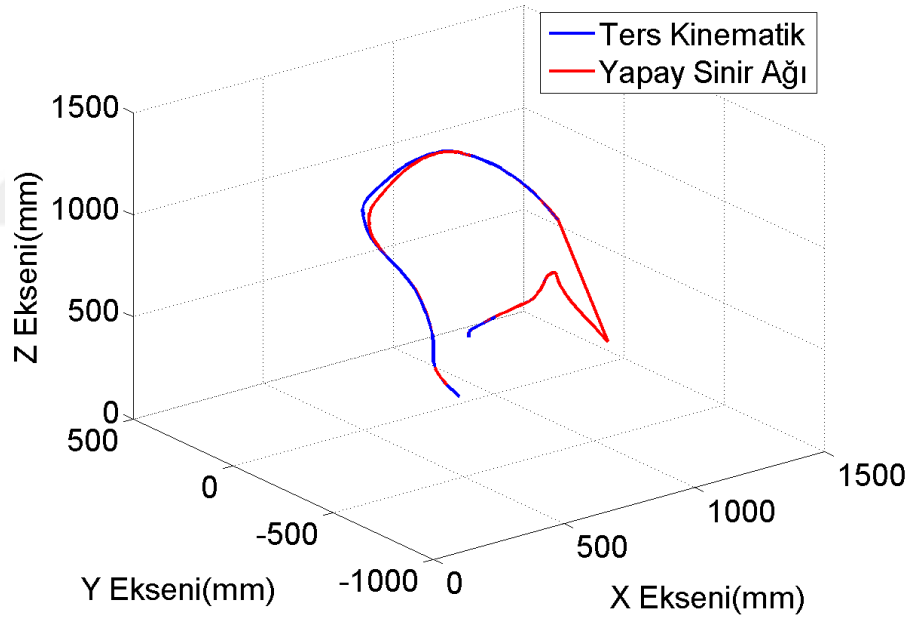
Test	Ters Kinematik	Yapay Sinir Ağı	Performans
1	1.71913	0.04422	38.87100
2	1.71877	0.02400	71.61010
3	1.70825	0.02378	71.81730
4	1.70445	0.02371	71.87972
5	1.70143	0.02349	72.42310
6	1.70336	0.02294	74.24990
7	1.70497	0.02334	73.02710
8	1.70848	0.02350	72.67417
9	1.69891	0.02361	71.95488
10	1.69908	0.02364	71.87069
Ortalama	0.4955	0.0140	69.03779

Çizelge 5.1’de son kolonda gösterilen performans katsayısı parametresi iki yöntemin hesaplama zamanlarının birbirlerine oranlanması ile elde edilmiştir. Elde edilen ortalama sonuçlar karşılaştırıldığında yapay sinir ağı yönteminin ters kinematik yönteminden yaklaşık olarak 70 kat daha hızlı çalıştığı görülmektedir. Test yapılan bilgisayarın işlem gücü göz önüne alındığında iki yöntemin de eş zamanlı kontrol için yeterli performansı sağladığı görülmektedir. Ancak çözüm yöntemleri işlem gücü daha düşük olan bir mikroişlemci üzerine taşınması durumunda işlem karmaşıklığı daha çok olan ters kinematik çözümü yönteminin zaman performansı yönünden yapay sinir ağı yöntemine göre daha düşük performansla çalışacağı öngörülmektedir.**Doğruluk Performansı Ölçümü**

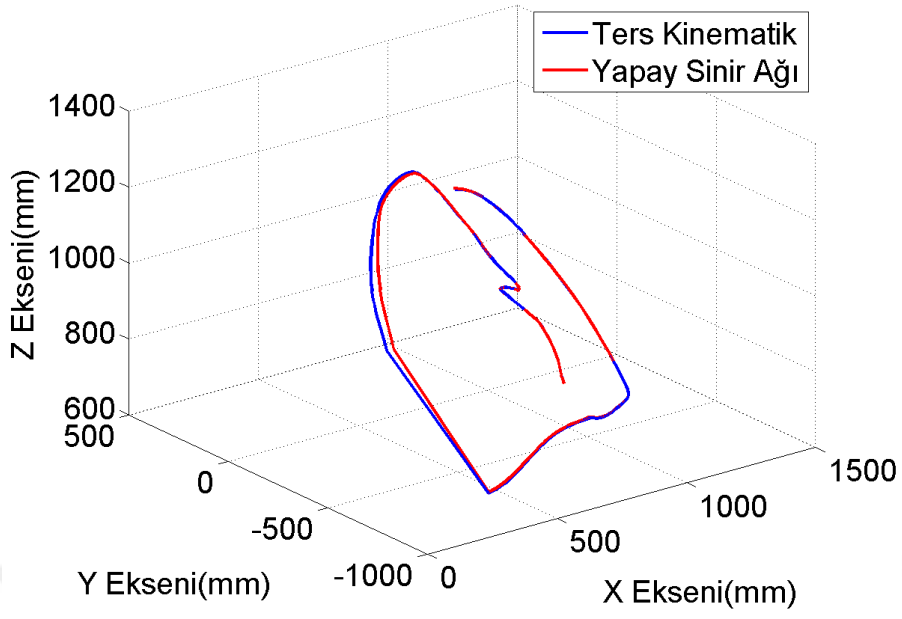
Bu testlerde yapay sinir ağı çözüm yönteminin, ters kinematik yöntemi çözümleri referans alınarak doğruluğu ölçülmüştür. Daha önce de bahsedildiği gibi ters kinematik yönteminin referans olarak alınmasının nedeni uç işlemci pozisyonunu

veren eklem açılarının bu çözüme göre hesaplanmış olmasıdır. Bu nedenle ters kinematik çözüm yönteminden çıkacak eklem açıları her zaman operatörün bilek pozisyonu ile robotun uç işlemcisini her zaman aynı pozisyonda eşleyecektir. Ters kinematik yöntemine girdi olarak verilen dönüşüm matrisinin son kolonundaki değerler ise anlık olarak uç işlemcinin pozisyonunu belirtmektedir.

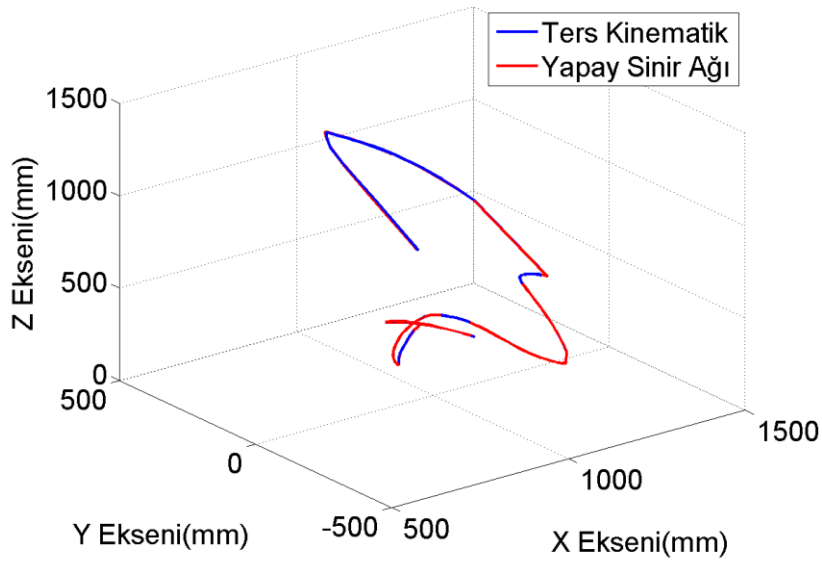
Test için üç boyutlu uzayda farklı şekiller oluşturularak eğitilen yapay sinir ağının ve ters kinematik çözümü yönteminin bu figürleri ne kadar takip ettiği ölçülmüştür. Test için ayrıca iki farklı boyuttaki veri setleri ile eğitilen farklı yapay sinir ağları kullanılmıştır. Filtrelendikten sonra 1438 çözüm örneği içeren veri setinden oluşan yapay sinir ağının farklı figürler için ölçüm sonuçları Şekil 5.6 – 5.9 arasında gösterilmiştir. Bu yapay sinir ağının ters kinematik yöntemi ile arasındaki farkı gösteren hata histogramları Şekil 5.10- 5.13 arasında gösterilmiştir.



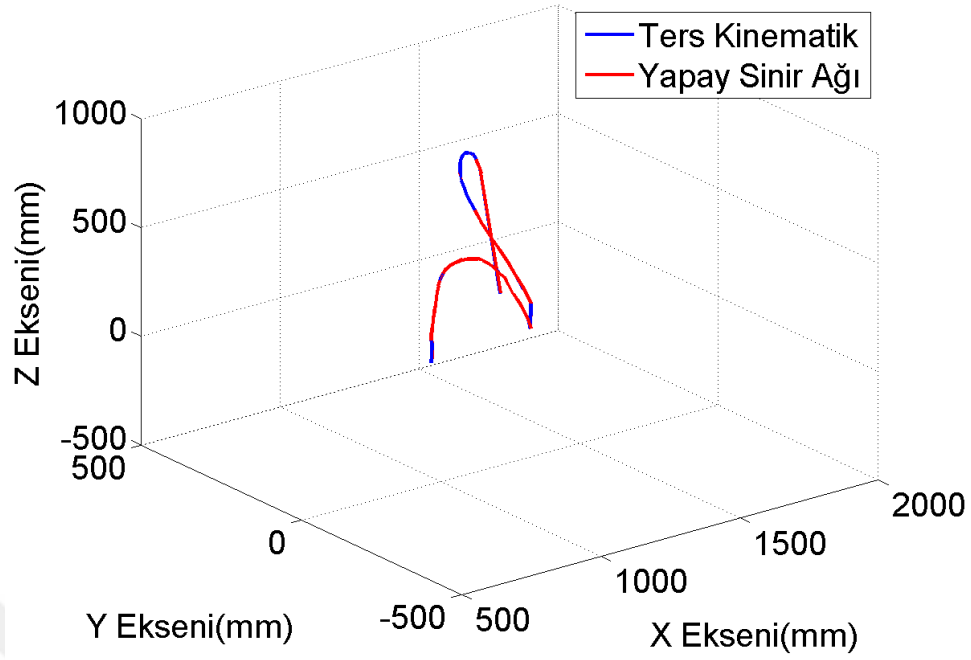
Şekil 5.6- Doğruluk karşılaştırması figür 1.



Şekil 5.7- Doğruluk karşılaştırması figür 2.

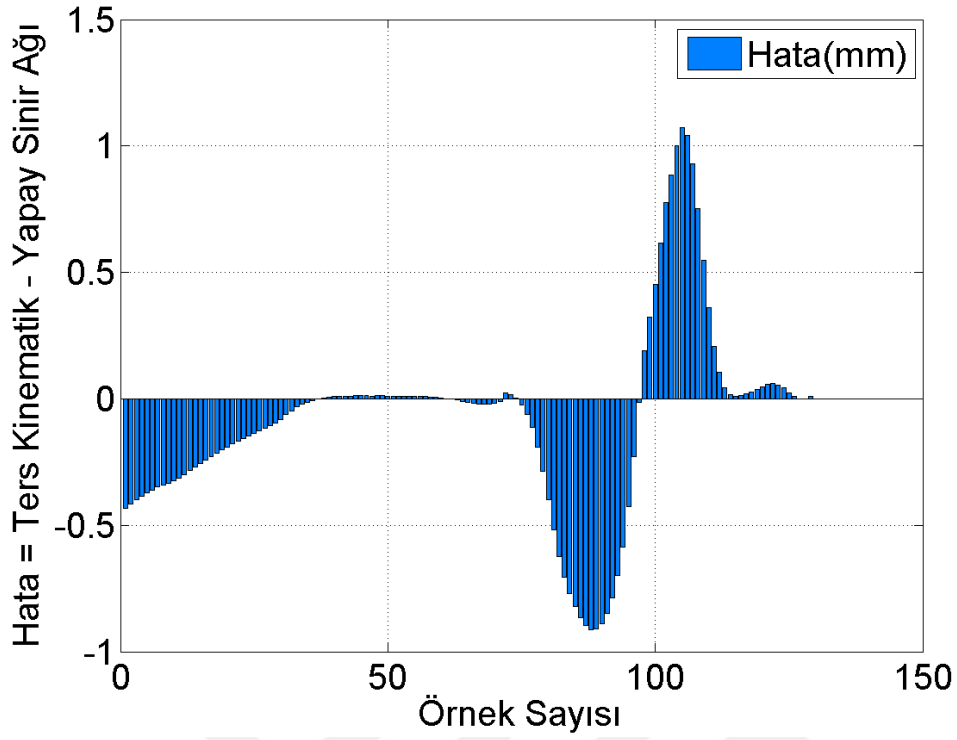


Şekil 5.8- Doğruluk karşılaştırması figür 3.

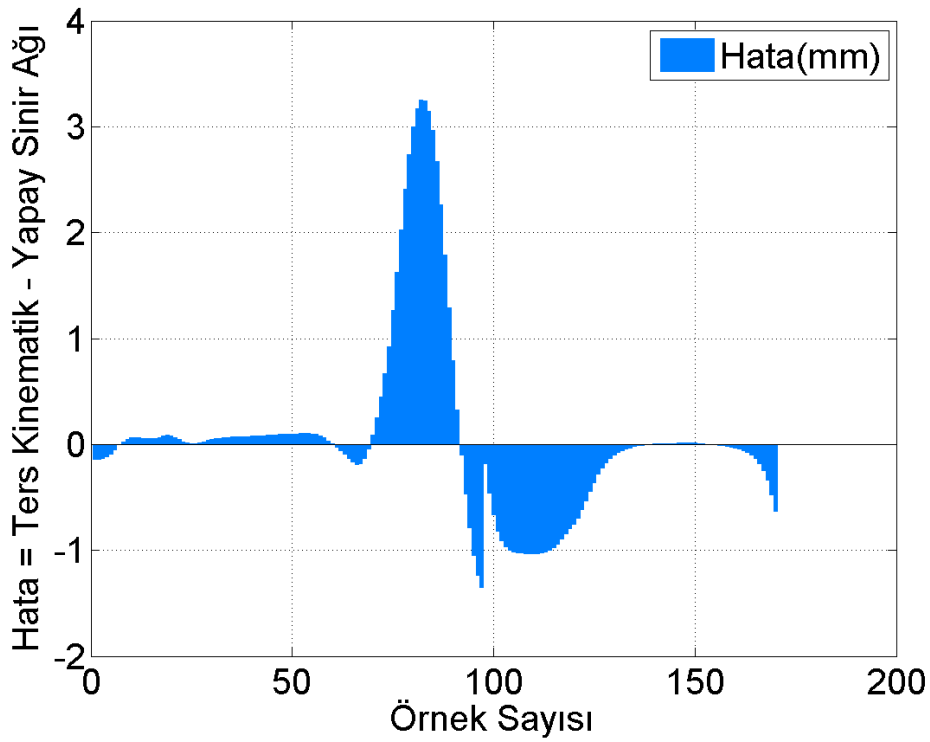


Şekil 5.9- Doğruluk karşılaştırması figür 4.

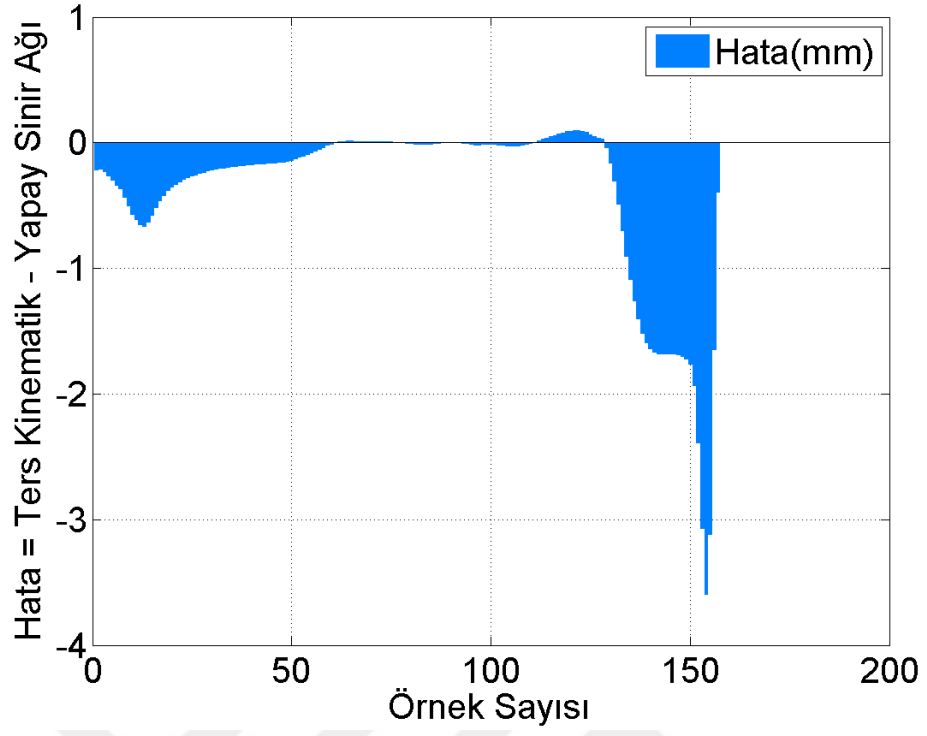
Yukarıdaki figürler ilk yapay sinir ağının ters kinematik çözümü ile karşılaştırıldığında uç işlemci pozisyonlarını göstermektedir. Figürlerden görülebileceği gibi yapay sinir ağı yöntemi ile figürlerin takibi bazı noktalarda hatalara rağmen sağlanmıştır. Yöntemler arasındaki hatalar Şekil 5.10 - 5.13 arasında hata histogramlarında gösterilmiştir. Hata grafikleri deney yapılan figürlerde yapay sinir ağı yönteminin ve ters kinematik çözümü yönteminin verdiği sonuçların farkı olarak tanımlanmıştır. Burada yapay sinir ağı çözümü ile elde edilen açılar (4.11)'de gösterilen düz kinematik eşitliği kullanılarak uç işlemci pozisyonuna çevrilmiştir. Daha sonra elde edilen üç boyutlu uç işlemci pozisyonundan orijine olan bileşke uzaklık tanımlanmıştır. Bu bileşke uzaklık ile ters kinematik yöntemini girdi olarak kullanılan uç işlemci pozisyonunun bileşke uzaklığı karşılaştırılarak hata değeri bulunmuştur.



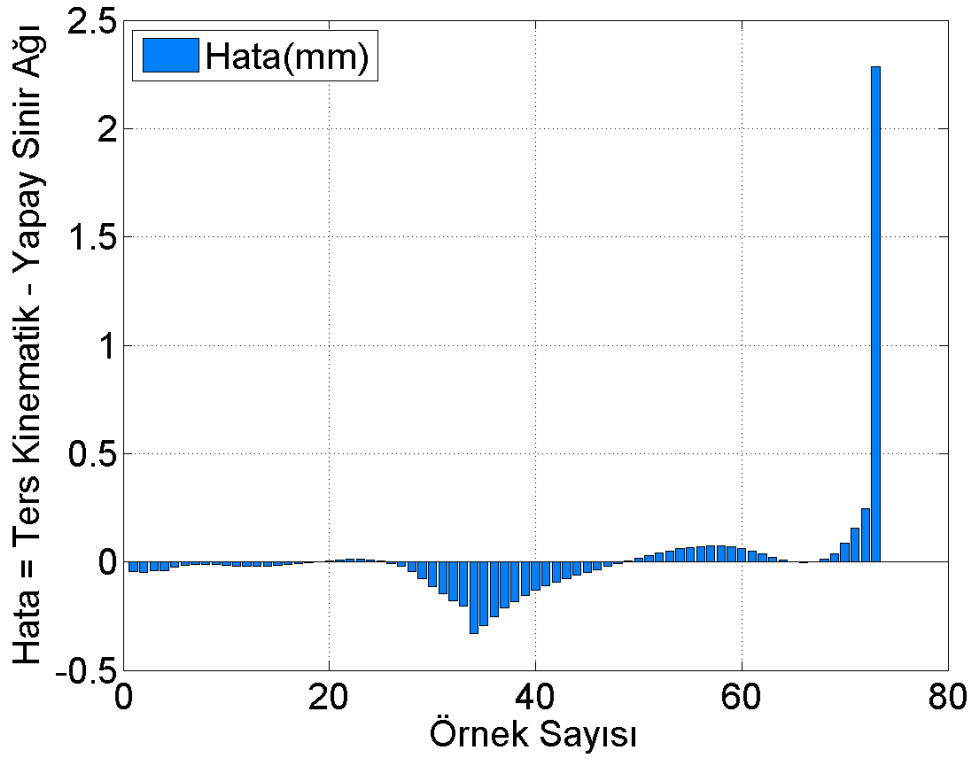
Şekil 5.10- Figür 1 hata histogramı.



Şekil 5.11- Figür 2 hata histogramı.



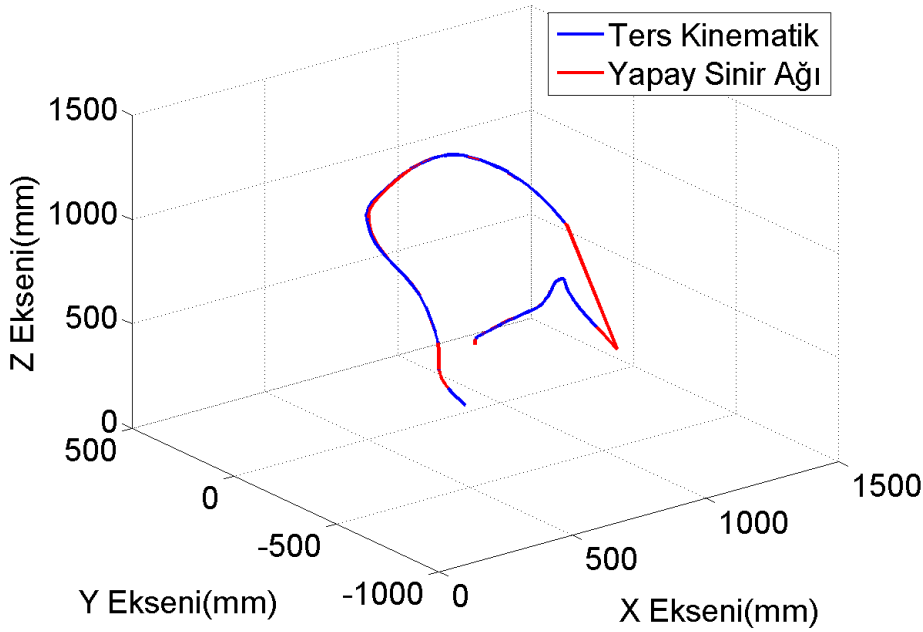
Şekil 5.12- Figür 3 hata histogramı.



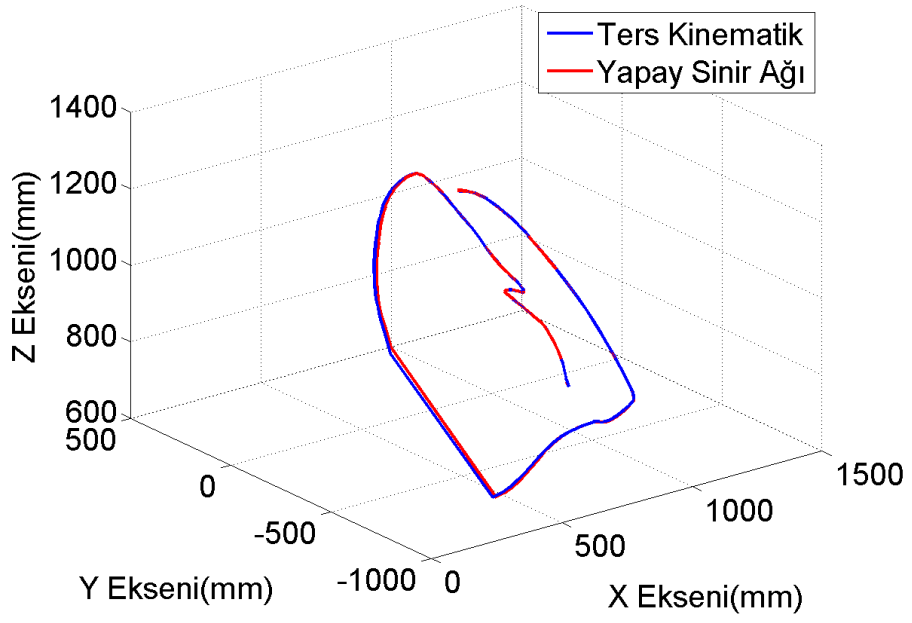
Şekil 5.13- Figür 4 hata histogramı.

Hata histogramlarında görüldüğü gibi daha düşük sayıda veri ile eğitilmiş olan yapay sinir ağının performansı bazı figürleri takip ederken iyi sayılabilecek bir değerde olsa da özellikle Şekil 5.7’de gösterilen figür 2’yi takip ederken hata diğer figürlerden yüksek çıkmıştır. Burada uç işlemci pozisyonunda 3.5 mm’ye varan hata gözlenmiştir. Şekil 5.9’da gösterilen figür 4’te gerçekleştirilen hareketin takibi sırasında ise 2 mm’nin üzerinde bir hata değeri alınmıştır.

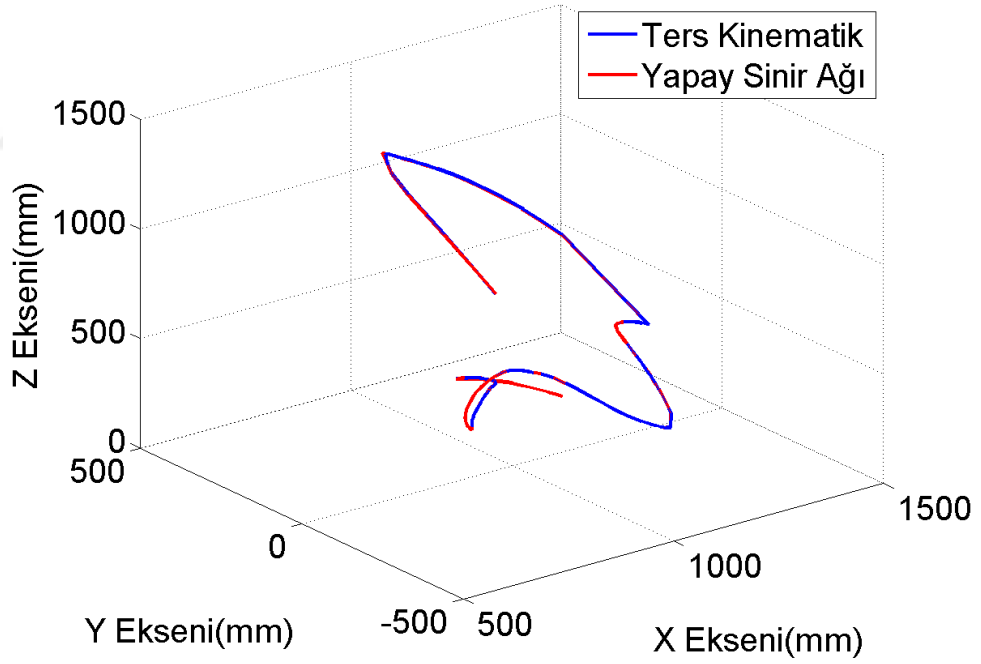
Elde edilen hata değerlerini düşürebilmek ve performansın veri sayısı ile bağıntısını ölçmek için daha yüksek veri sayısı ile başka bir eğitim yapılmıştır. Bu veri seti 5383 örnekten oluşmaktadır. Örnekler üzerinde Bölüm 4.3’te belirtilen filtreler uygulandıktan sonra 2999 çözüm örneği elde edilmiştir. Bu sayı da bir önceki yapay sinir ağının eğitimi sırasında kullanılan örnek sayısının iki katından daha fazladır. Bu çözüm örnekleri ile eğitilmiş olan yapay sinir ağının figür takibi ölçüm sonuçları Şekil 5.14-5.17 arasında gösterilmiştir.



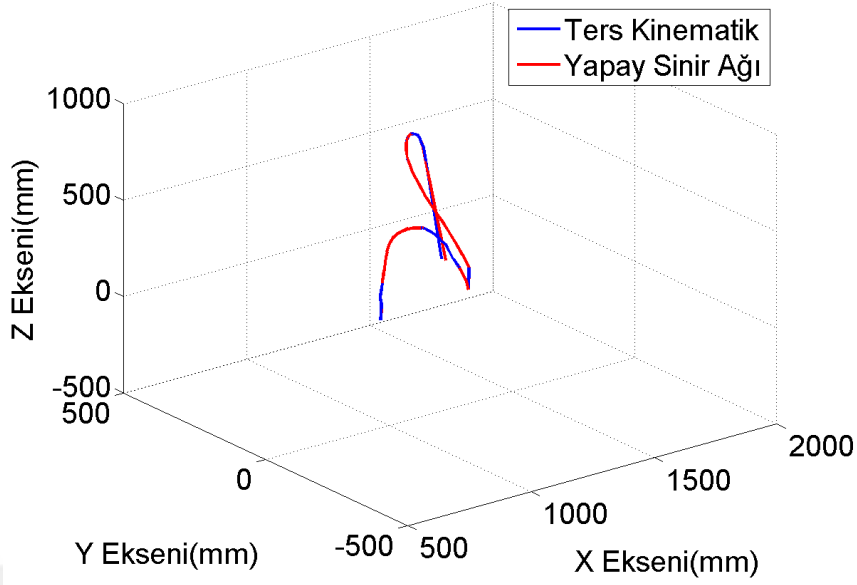
Şekil 5.14- Doğruluk karşılaştırması figür 1- yapay sinir ağı 2.



Şekil 5.15- Doğruluk karşılaştırması figür 2- yapay sinir ađı 2.

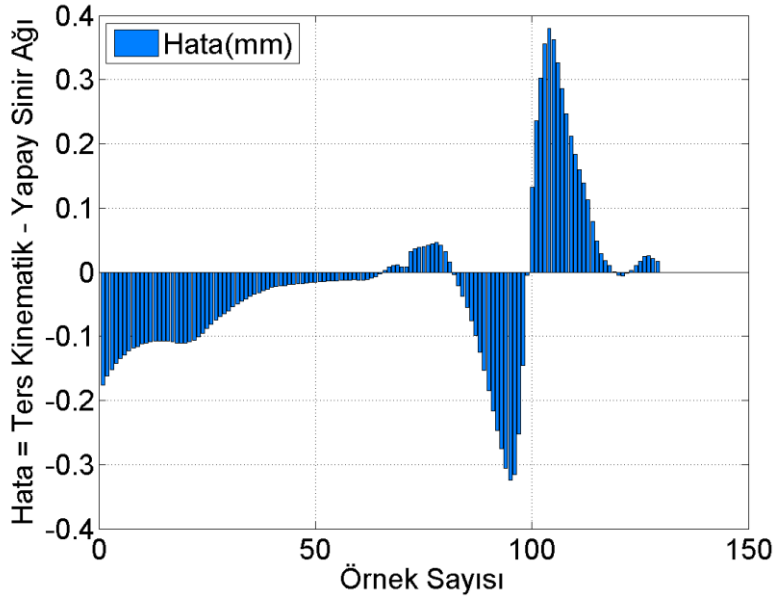


Şekil 5.16- Doğruluk karşılaştırması figür 3- yapay sinir ađı 2.

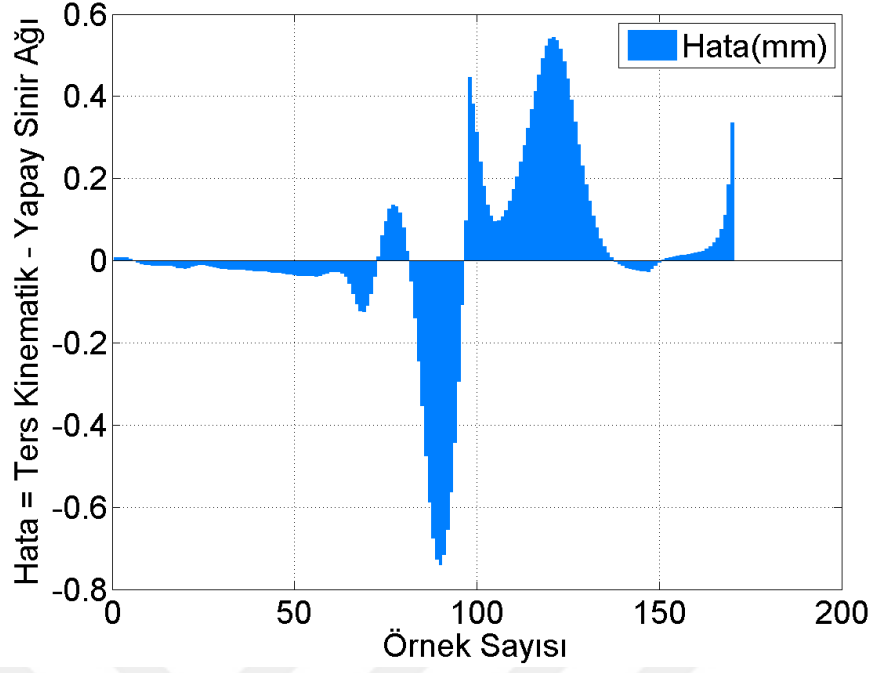


Şekil 5.17- Doğruluk karşılaştırması figür 4- yapay sinir ağı 2.

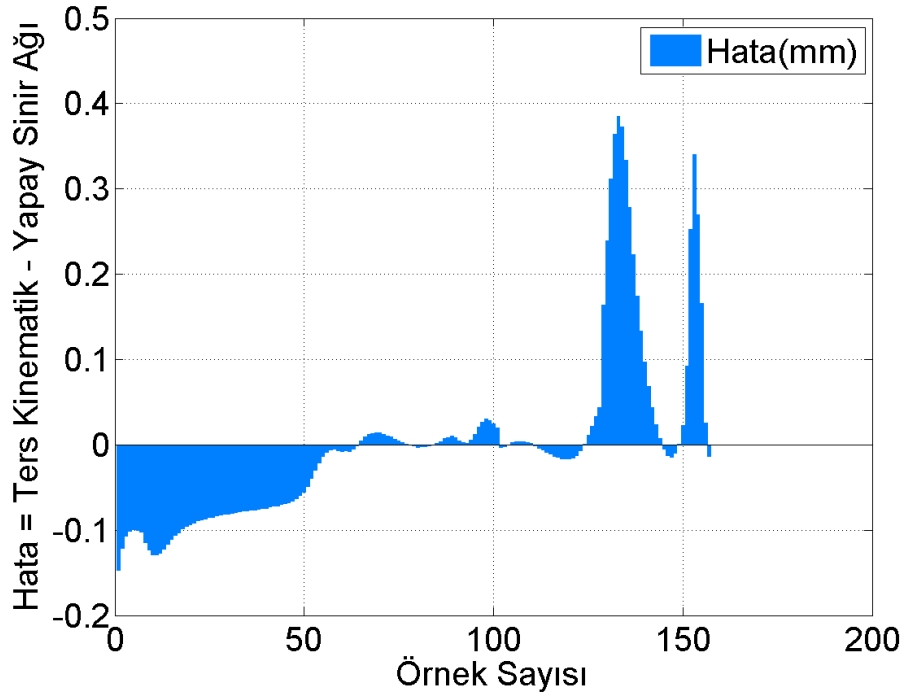
Daha fazla çözüm örneği içeren veri seti ile eğitilmiş yapay sinir ağının, figürleri takip etme ölçümü daha az çözüm örneği içeren veri seti ile eğitilmiş olan yapay sinir ağına göre daha başarılıdır. Figürlerin tamamında şekilsel bir bozukluk olmadan uç işlemci pozisyonu referans pozisyonu takip edebilmiştir. Bu ölçümlere ait hata histogramları Şekil 5.18 – 5.21 arasında gösterilmiştir.



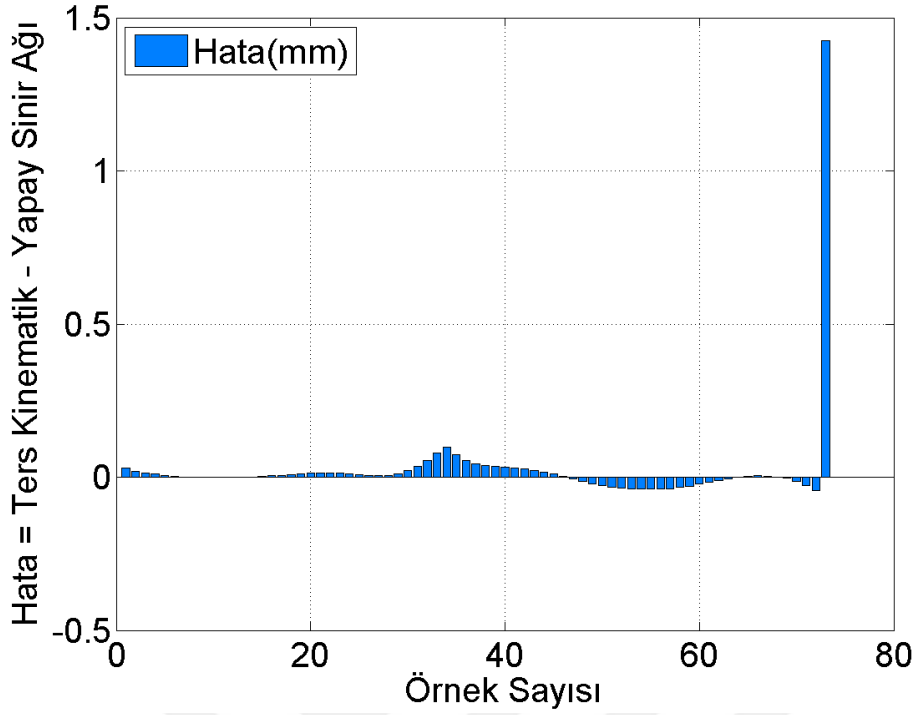
Şekil 5.18- Figür 1 hata grafiği- yapay sinir ağı 2.



Şekil 5.19- Figür 2 hata grafiği- yapay sinir ağı 2.



Şekil 5.20- Figür 3 hata grafiği- yapay sinir ağı 2.



Şekil 5.21- Figür 4 hata grafiği- yapay sinir ağı 2.

Yukarıdaki hata histogramlarından da görüleceği gibi daha çok çözüm örneği içeren veri seti ile eğitilmiş yapay sinir ağının doğruluk performansı daha önceki yapay sinir ağına göre gelişmiştir. Hata histogramları incelendiğinde bileşke uzaklık cinsinden gösterilen uç işlemci pozisyonları arasındaki farklar operasyon için kabul edilebilir düzeyde bulunmuştur. En yüksek hata değeri Şekil 5.17’de gösterilen figür 4’ün takip edilmesi sırasında oluşmuştur. Burada uç işlemciler arasındaki bileşke uzaklığın hata değeri yaklaşık 1.5 mm olarak hesaplanmıştır. Bir önceki yapay sinir ağında 3.5 mm hata değeri elde edilen figür 2’de ise elde edilen hata değerleri azalarak 0.5 mm seviyesine gelmiştir. Eğitim setleri için seçilen örnekler, yukarıda deneyleri yapılan figürlere ait hareketlerin operatör tarafından sürekli tekrar edilmesi sırasında hesaplanan ters kinematik çözümlerden çıkarılmıştır. Bu nedenle veri setlerinde diğer figürlere oranla daha az çözüm örneği bulunduran figürlerin hata değerleri diğer figürlere göre yüksek çıkmıştır. Veri setlerinde bulunan çözüm örneği sayısının artırılması ile elde edilen hataların azalması, doğruluk performansının veri sayısı ile doğru orantılı olduğunu göstermiştir. Bu testlere bağlı olarak kullanılan veri setinin çözüm örneği sayısının artırılmasının doğruluk performansı ile doğru orantılı

bir sonuç verdiđi gözlemlenmiştir. Bu testlerle robotun gerçekleştirilmesi istenilen hareketler operatör tarafından Kinect karşısında tekrar edildikçe artacak çözüm örneđi sayısı ile bu hareketin yapay sinir ađı yöntemi kullanılarak gerçekleştirilmesi sırasında elde edilecek doğruluk performansının artacađı gösterilmiştir.





6. SONUÇLAR

Bu çalışmada Stäubli Rx160 robot kolu Microsoft Kinect kullanılarak eş zamanlı olarak uzaktan kontrol edilmiştir. Merkez birim olarak kabul edilen operatör hareketleri Kinect'ten alınan derinlik haritası analiz edilerek hesaplanmış, uydu birim olan robota aktarılmıştır. Teleoperasyon için üç farklı yöntem denenerek sonuçları karşılaştırılmıştır. Bu yöntemlerden ikisi merkez birim ile uydu birimin uç işlemci pozisyonlarını eşleme üzerine kurulu iken diğer yöntem birebir eklem açılarını eşleme esasına dayanmaktadır.

Kinect'ten iskelet haritasındaki eklem pozisyonları işlenerek operatörün eklem açıları hesaplanmıştır. Aynı zamanda operatörün çalışma uzayı robotun çalışma uzayıyla eşleşecek biçimde genişletilmiştir. Çalışma uzayı eşlemenin operatörü çalışma uzayını genişleterek uygulanması seçilen senaryodan kaynaklıdır. Bu çalışmada gerçekleştirilmek istenen senaryolar operatörün limitlerinin genişletilerek normalde erişemeyeceği yerlere erişip, bu noktalara kendi gücü ile taşıyamayacağı ağırlıkları taşıması veya kendi uzuv boyutları ile erişemediği noktalarda robotu kullanarak gerekli işleri yapması gibi senaryolardır. Bu senaryolara uygun olarak yapılan çalışma uzayı eşlemesinde operatörün iskelet haritasında bulunan sağ el bilek eklemi pozisyonu robotun parametreleri kullanılarak genişletilmiş ve kontrol edilmek istenen Stäubli Rx160 robotun uç işlemci pozisyonu birbirine eşlenmiştir. Bu eşleme ile ters kinematik denklemleri kullanılarak uç işlemci pozisyonundan robotun eklem açıları bulunmuş ve eş zamanlı olarak robota gönderilmiştir.

Ters kinematik çözümlerde birden fazla açı değerinin aynı uç işlemci pozisyonuna ulaşabilmektedir. Bu nedenle robotun dirsek-yukarı pozisyonda çalışabilmesi için çözümler filtrelenmiştir. Yapay sinir ağı yöntemi için veri seti hazırlanırken filtrelenen veriler kullanılmıştır. Operatör tarafından gerçekleştirilen hareketlerin tekrarından oluşan farklı büyüklükteki veri setleriyle eğitilen yapay sinir ağı karşılaştırılmış, çözüm örnekleri sayısının doğruluk performansı ile doğru orantılı

olduğu gözlenmiştir. Bu yöntem ile elde edilen teleoperasyon yönteminin doğruluk değerinin uç işlemci pozisyonu doğruluğu bazında milimetre hassasiyetinde olduğu, ayrıca ters kinematik çözümü yöntemine göre yaklaşık 70 kat daha hızlı olduğu gözlemlenmiştir. Özellikle daha düşük işlem gücüne sahip platformlar üzerinde eş zamanlı operasyon gerektiği durumlarda yapay sinir ağ yönteminin ters kinematik çözümü yöntemine göre daha yüksek hesaplama zamanı performansı ile çalışacağı gözlemlenmiştir. Hassasiyet ile zaman performansı göz önüne alınarak iki yöntem karşılaştırıldığında yapay sinir ağı yöntemi öne çıkmıştır.

Direkt eşleme yöntemi referans olması açısından denemiş ve etkili olmadığı gösterilmiştir. Bunun nedeni operatör ile robotun farklı kinematik yapıda olmasıdır. Merkez birim olan operatörün sağ kolunda takip edilen eklem açıları ile robotun eklem açıları birbirinden farklıdır. Operatörün omuz-dirsek ekleminde Stäubli Rx160'da bulunmayan ekstra bir serbestlik derecesi daha bulunmaktadır. Bu nedenle direkt açı eşleme yöntemi uç işlemci pozisyonlarında farklılık yaratabilmektedir.

Gelecek çalışmalarda robotun uç işlemcisine eklenecek olan tutucu birimin kontrolünün sağlanması hedeflenmiştir. Microsoft Kinect yazılım geliştirme setinin şu anda geliştiricilere sunduğu iskelet haritasında parmak eklemlerini dışarıda tutmuştur. Ancak Bölüm 2'de özetlenen çalışmaların bir kısmında derinlik haritası üzerinde görüntü işleme yöntemi ile parmakların işaretlenebildiği gözlemlenmiştir. Bu sayede operatörün sağ elinde işaret ve baş parmak bulunarak aralarındaki açıklık değeri ile robotun uç işlemcisinde bulunan tutucu birimin kontrolünün sağlanabileceği öngörülmüştür. Aynı zamanda ataletsel ölçüm birimi kullanılarak bilek yönelimlerini hesaplamak ve yönelimlerle beraber robotun 6 eklemi de kontrol edebilecek bir teleoperasyon yöntemi de yapılabilecek çalışmalardan biridir. Bu çalışmalar sonlandığında operatörler Stäubli Rx160 robot kolunu kullanarak çalışma uzayı içerisinde bulunacak objeleri bir pozisyondan başka bir pozisyona rahatlıkla taşıyabilecektir. Bununla beraber yapay sinir ağı eğitimi için hazırlanacak daha detaylı bir veri seti ile robotun dirsek-aşağı, dirsek-yukarı gibi birden fazla konfigürasyonda da çalışması planlanan çalışmalar arasında bulunmaktadır. Yapılan çalışmada gürültülü derinlik görüntüsü karelerinden hesaplanan açılar kinematik doğrulama modelinden geçemediği için elenmektedir.

Bu nedenle bazı görüntü karelerinden açı hesaplanmadığı için teleoperasyon sırasında veri kaybı yaşanmaktadır. Kinematik modele uymadığı için elenen açılardan çözüm setinden çıkarılmasının yerine, zamansal ve kinematik kısıtlar çerçevesinde gürültülü kareler için eklem açılarının tahmin edilmesi yürütülmesi gelecek çalışmalar içerisinde yer alacaktır. Bu sayede sürekliliği artırılmış bir teleoperasyon yöntemi sağlanacaktır. Kullanıcının hareketlerine robotun eş zamanlı olarak verdiği tepkilerin limitlerini belirlemek için maket kol modeli üzerinde performans denemeleri yapılması da gelecek çalışmalar içerisinde bulunacaktır. Bu çalışma ile operatörün hareketi gerçekleştirme sırasındaki eklem hızlarının değiştirilerek kullanılan teleoperasyon yöntemlerinin fiziksel limitleri bulunabilecektir.

Bu çalışmada kullanılan çalışma uzayı genişletilmesi senaryosu dışındaki senaryolar gelecek çalışmalar için zemin oluşturmaktadır. Bu senaryolar için robotun operatörün hareketlerine göre artımlı tepki vermesi veya daha hassas çalışma gereken işlerde sadece uç işlemcinin çalışma uzayı eşlenerek daha küçültülmüş bir ortamda çalışma yapılması örnekleri verilebilir. Bu durumlarda mevcut durumda yapılan robot parametreleri kullanılarak uzuv genişletilmesi yerine artımlı çalışmada operatörün hareketlerinin zamansal analizi ile robotun kontrol edilmesi, hassas çalışma modu için ise robotun çalışma uzayının daraltılması gibi çalışmalar yürütülebilir.



KAYNAKLAR

- [1] S. Foix, G. Alenya ve C. Torras, Lock-in Time-of-Flight (ToF) Cameras: A Survey, *IEEE Sensors Journal*, cilt 11, no. 9, pp. 1917-1926, 2011.
- [2] Microsoft, Microsoft Kinect Xbox 360, Microsoft, 2015. [Çevrimiçi], Erişim adresi : <http://www.xbox.com/en-US/xbox-360/accessories/kinect>, Erişim tarihi: 17 11 2015.
- [3] Y. Kim, L. Simon, S. Azad, K. Axel ve P. Kim, Kinect technology for hand tracking control of surgical robots: technical and surgical skill comparison to current robotic masters, *Surgical Endoscopy*, cilt 28, no. 6, pp. 1993-2000, 2014.
- [4] Staubli Robotics, VAL3 Language, Stäubli, 2016. [Çevrimiçi], Erişim adresi : <http://www.staubli.com/en/robotics/robot-software/val3-robot-programming/val3-language/>, Erişim tarihi: 10 8 2015.
- [5] T. Moelund ve E. Granum, A Survey of Computer Vision-Based Human, *Computer Vision and Image Understanding*, cilt 81, pp. 231-268, 2001.
- [6] T. Weaver, Hacking The Microsoft Kinect, 2016. [Çevrimiçi], Erişim adresi : <http://treyweaver.blogspot.com.tr/2011/03/hacking-microsoft-kinect.html>, Erişim tarihi: 14 02 2016.
- [7] D. Hogg, Model-based vision: a program to see a walking person, *Image and Vision Computing*, cilt 1, no. 1, pp. 5-20, 1983.
- [8] K. Rohr, Human Movement Analysis Based on Explicit Motion Models, içinde *Motion-Based Recognition*, Boston, Kluwer Academic Publishers, 1997, pp. 171-198.
- [9] D. Gavrila ve L. Davis, 3-D model-based tracking of humans in action: a multi-view approach, içinde *Conference on Computer Vision and Pattern Recognition, IEEE*, San Francisco, 1996.
- [10] C. Hu, Q. Tu, Y. Li ve S. Ma, Extraction of parametric human model for posture recognition using genetic, içinde *The fourth International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 2000.
- [11] H. Lee ve Z. Chen, Determination of 3D human body posture from a single

- view, *Comuter Vision, Graphics and Image Processing*, cilt 30, pp. 148-168, 1985.
- [12] H. Lee ve Z. Chen, Knowledge-guided visual perception of 3-D human gait from a single image sequence, *IEEE Transaction on Systems, Man and Cybernatics*, cilt 22, no. 2, pp. 336-342, 1992.
- [13] C. Atwood, G. Sullivan ve K. Baker, Model-based Recognition of Human Posture Using Single Synthetic Images, içinde *Fifth Alvey Vision Conference*, United Kingdom, 1989.
- [14] O. Munkelt, C. Ridder, D. Hansel ve W. Hafner, A model driven 3D image interpretation system applied to person detection in video images, içinde *International Conferance on Pattern Recognition*, 1998.
- [15] W. Hafner ve O. Munkelt, Using Color for Detecting Per-sons in Image Sequences, *Pattern Recognition and Image Analysis*, cilt 7, no. 1, pp. 47-52, 1996.
- [16] Y. Guo, G. Xu ve S. Tsuji, Tracking human body motion based on a stick figure model, *Journal of Visual Communication and Image Representation*, cilt 5, no. 1, pp. 1-9, 1994.
- [17] R. Plänkers, P. Fua ve N. D'Apuzzo, Automated body modeling from video sequences, içinde *International Workshop on Modeling People at ICCV'99*, Greece, 1999.
- [18] N. Jovic, B. Brumitt, B. Meyers, S. Harris ve T. Huang, Detection and estimation of pointing gestures in dense disparity maps, içinde *The Fourth International Conference on Automatic Face and Gesture Recognition*, France, 2000.
- [19] Point Grey, Triclops SDK Home Page, Point Grey, 2015. [Çevrimiçi], Erişim adresi : <https://www.ptgrey.com/triclops>, Erişim tarihi: 16 11 2015.
- [20] Point Grey, Triclops SDK, Point Grey Research, Inc., 2015. [Çevrimiçi], Erişim adresi : <https://www.ptgrey.com/triclops>, Erişim tarihi: 14 02 2016.
- [21] F. Lerasle, G. Rives ve M. Dhome, Human body limbs tracking by multi-ocular vision, içinde *Scandinavian Conference on Image Analysis*, Finland, 1997.
- [22] F. Lerasle, G. Rives ve M. Dhome, Tracking of human limbs by multiocular vision, *Computer Vision and Image Understanding*, cilt 75, no. 3, pp. 229-246, 1999.
- [23] H. Sidenbladh, F. De la Torre ve M. Black, A Framework for Modeling the Appearance of 3D Articulated Figures, içinde *The Fourth International*

Conference on Automatic Face and Gesture Recognition, France, 2000.

- [24] M. Yamamoto ve K. Koshikawa, Human motion analysis based on a robot arm model, içinde *Conference on Computer Vision and Pattern Recognition*, 1991.
- [25] S. Ju, M. Black ve Y. Yacoob, Cardboard People: A parameterized model of articulated image motion, içinde *International Conference on Automatic Face and Gesture Recognition*, Killington, 1996.
- [26] Microsoft, Microsoft Kinect SDK, Microsoft, 2015. [Çevrimiçi], Erişim adresi : <https://www.microsoft.com/en-us/download/details.aspx?id=44561>, Erişim tarihi: 17 11 2015.
- [27] J. Han, L. Shao, D. Xu ve J. Shotton, Enhanced Computer Vision with Microsoft Kinect Sensor: A Review, *IEEE Transactions on Cybernetics*, cilt 43, no. 5, pp. 1318-1334, 2013.
- [28] L. Xia, C. Chen ve J. Aggarwal, Human detection using depth information by Kinect, içinde *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011.
- [29] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte ve J. Meunier, Fall detection from depth map video sequences, içinde *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, 2011.
- [30] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman ve A. Blake, Real-time human pose recognition in parts from a single depth image, içinde *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [31] M. Ye, X. Wang, R. Yang, L. Ren ve M. Pollefeys, Accurate 3D Pose Estimation From a Single Depth Image, içinde *International Conference on Computer Vision*, 2011.
- [32] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo ve Z. Tu, Exemplar-Based Human Action Pose Correction and Tagging, içinde *Computer Vision and Pattern Recognition*, Providence, RI, 2012.
- [33] X. Yang ve Y. Tian, EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor, içinde *IEEE Computer Vision and Pattern Recognition Workshops(CVPRW)*, Providence, 2012.
- [34] X. Yang ve Y. Tian, Effective 3-D action recognition using eigenJoints, *Journal of Visual Communication and Image Representation*, cilt 25, no. 1, pp. 2-11, 2014.

- [35] R. Tara, P. Santosa ve T. Adji, Hand segmentation from depth image using anthropometric approach in natural interface development, *International Journal of Scientific & Engineering Research*, cilt 3, no. 5, pp. 1-4, 2012.
- [36] U. Lee ve J. Tanaka, Finger Controller: Natural User Interaction Using Finger Gestures, içinde *Human-Computer Interaction: Interaction Modalities and Techniques*, Springer, 2013, pp. 281-290.
- [37] H. Liang, J. Yuan ve D. Thalman, 3-D fingertip and palm tracking in depth image sequences, içinde *Proceedings of the 20th ACM international conference on Multimedia*, 2012, 2012.
- [38] G. Hanckenberg, R. McCall ve W. Broll, Lightweight palm and finger tracking for real-time 3-D gesture control, içinde *Virtual Reality Conference(VR), IEEE*, 2011, 2011.
- [39] C. Keskin, F. Kira, Y. Kara ve L. Akarun, Real time hand pose estimation using depth sensors, içinde *ICCV Workshops, IEEE*, 2011.
- [40] C. Keskin, F. Kira, Y. Kara ve L. Akarun, Hand pose estimation and hand shape classification using multilayered randomized decision forests, içinde *Computer Vision EECV 2012*, Springer, 2012, pp. 852-863.
- [41] M. Al-Faiz ve A. Shanta, Kinect- Based Humanoid Robotic Manipulator for Human Upper Limbs Movements Trackingg, *Intelligent Control And Automation*, cilt 6, pp. 29-37, 2015.
- [42] Aldebaran, NAO Documentation, Aldebaran, 2015. [Çevrimiçi], Erişim adresi : http://doc.aldebaran.com/2-1/home_ao.html, Erişim tarihi: 25 11 2015.
- [43] S. Mukherjee, D. Paramkusam ve S. Dwivedy, Inverse Kinematics of a NAO Humanoid Robot using Kinect to Track and Imitate Human Motion, içinde *International Conference on Robotics, Automation, Control and Embedded Systems*, Chennai, India, 2015.
- [44] R. Schilling, Fundamentals of Robotics: Analysis and Control, Prentice Hall, 1990.
- [45] Mathworks, Neuro-Adaptive Learning and ANFIS, Mathworks Inc., 24 11 2015. [Çevrimiçi], Erişim adresi : <http://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html>, Erişim tarihi: 24 11 2014.
- [46] O. Yongsheng, H. Jianbing, W. Zhiyang, F. Yiquin, W. Xinyu ve L. Xiaoyun, A Real-Time Human Imitation System Using Kinect, *International Journal of Social Robotics*, cilt 7, no. 5, pp. 587-600, 2015.

- [47] J. More, The Levenberg-Marquardt algorithm: Implementation and theory, içinde *Lecture Notes in Mathematics*, Berlin Heidelberg, Springer, 1978, pp. 105-116.
- [48] R. Aftoni, A. Rizal ve E. Susanto, Proportional Derivative Control Based Robot Arm System Using Microsoft Kinect, içinde *International Conference on Robotics, Biomimetics, Intelligent Computational Systems*, Yogyakarta, Indonesia, 2013.
- [49] V. Nguyen ve J. Lee, Full-Body Imitation of Human Motions with Kinect and Heterogeneous Kinematic Structure of Humanoid Robot, içinde *International Symposium on System Integration*, Fukuoka, Japan, 2012.
- [50] I. Almetwally ve M. Mallem, Real-time tele-operation and tele-walking of humanoid Robot Nao using Kinect Depth Camera, içinde *10th IEEE International Conference on Networking, Sensing and Control*, Evry, France, 2013.
- [51] Robokind, Zeno R50 Documentation, Robokind, 2015. [Çevrimiçi], Erişim adresi : <http://www.robokindrobots.com/support-documentation/r50/>, Erişim tarihi: 25 11 2015.
- [52] N. Torres, N. Clark ve I. Ranatunga, Implementation of interactive arm playback behaviors of social robot Zeno for autism spectrum disorder therapy, içinde *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*, Haraklion, Crete, Greece, 2012.
- [53] H. Reddivari, C. Yang, Z. Ju, P. Liang ve B. Xu, Teleoperation Control of Baxter Robot using Body Motion Tracking, içinde *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems*, Beijing, 2014.
- [54] C. Arango, M. J.R. ve V. Perez, Master-slave system using Kinect and an industrial robot for teleoperations, içinde *Pan American Health Care Exchanges Conference*, Medellin, Colombia, 2013.
- [55] Kuka Robotics, Kuka KR6, Kuka Robotics, 2015. [Çevrimiçi], Erişim adresi : http://www.kuka-robotics.com/en/products/industrial_robots/low/kr6_2/start.htm, Erişim tarihi: 25 11 2015.
- [56] D. Guanglong, Z. Ping, M. Jianhua ve L. Zeling, Markerless Kinect-Based Hand Tracking for Robot Teleoperation, *International Journal of Advanced Robotic Systems*, cilt 9, no. 10, 2012.
- [57] G. Tang, S. Asif ve P. Webb, The integration of contactless static pose

- recognition and dynamic hand motion, *Industrial Robot: An International Journal*, cilt 42, no. 5, pp. 416-428, 2015.
- [58] T. I. Cerlinca, S. Pentiu ve V. Vlad, Real-Time 3D Hand Gestures Recognition for Manipulation of Industrial Robots, *Elektronika IR Elektrotehnika*, cilt 19, no. 2, pp. 3-8, 2012.
- [59] P. Zhang, G. Du ve D. Li, A novel human-robot interface using hybrid sensors with Kalman filters, *Industrial Robot: An International Journal*, cilt 41, no. 6, pp. 585-595, 2014.
- [60] G. Bradsky, Computer Vision Face Tracking For Use in a Perceptual User, *Intel Technology Journal*, 1998.
- [61] L. Munoz ve A. Casals, Improving the human-robot interface through adaptive multispace transformation, *IEEE Transactions on Robotics*, cilt 25, no. 5, pp. 1208-1213, 2009.
- [62] D. Kruse, J. Wen ve R. Radke, A Sensor-Based Dual-Arm Tele-Robotic System, *Automation Science and Engineering, IEEE Transactions on*, cilt 12, no. 1, pp. 4-18, 2015.
- [63] M. Azimi, Skeletal Joint Smoothing White Paper, Microsoft, [Çevrimiçi], Erişim adresi : <https://msdn.microsoft.com/en-us/library/jj131429.aspx>, Erişim tarihi: 16 11 2015.
- [64] Microsoft, Joint Filtering, Microsoft, [Çevrimiçi], Erişim adresi : <https://msdn.microsoft.com/en-us/library/jj131024.aspx>, Erişim tarihi: 16 11 2015.
- [65] M. Mihelj, Human Arm Kinematics for Robot Based Rehabilitation, *Robotica*, cilt 24, no. 3, pp. 377-386, 2006.
- [66] N. Badler ve D. Tolani, Real-Time Inverse Kinematics of the Human Arm, *Center for Human Modeling and Simulation*, cilt 5, no. 4, pp. 393-401, 1996.
- [67] J. Rosen, J. Perry, N. Manning, S. Burns ve B. Hannaford, The Human Arm Kinematics and Dynamics During Daily Activities - Toward a 7 DOF Upper Limb Powered Exoskeleton, içinde *International Conference on Advanced Robotics*, Seattle, 2005.
- [68] E. Zhang, Kinect v1 SDK C++ -3. Kinect Point Clouds, Princeton University, [Çevrimiçi], Erişim adresi : <http://www.cs.princeton.edu/~edwardz/tutorials/kinect/kinect3.html>, Erişim tarihi: 17 11 2015.

- [69] Staubli Robotics, RX160 6-axis industrial robot, 2015. [Çevrimiçi], Erişim adresi : <http://www.staubli.com/en/robotics/6-axis-scara-industrial-robot/medium-payload-6-axis-robot/6-axis-industrial-robot-rx160/>, Erişim tarihi: 9 12 2015.
- [70] Y. Gültekin, Bir Endüstriyel Robotun İnsan Kolu Haerketlerinin Ataletsel Ölçümü ile Uzaktan Kontrolü, TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2012.
- [71] J. Craig, Introduction to Robotics: Mechanics and Control (3rd Edition), Addison-Wesley, 1989.
- [72] C. Reddy, Difference Between Denavit Hartenberg (D-H) Classical and Modified Conventions for Forward Kinematics of Robots with Case Study, içinde *International Conference on Advanced Materials and Manufacturing Technologies*, 2014.
- [73] MathWorks, Matlab Neural Network Fitting Toolbox, MathWorks, 2015. [Çevrimiçi], Erişim adresi : www.mathworks.com/help/nnet/ref/nftool.html, Erişim tarihi: 22 12 2015.
- [74] MathWorks, Train and Apply Multilayer Neural Networks , MathWorks, 2015. [Çevrimiçi], Erişim adresi : <http://www.mathworks.com/help/nnet/ug/train-and-apply-multilayer-neural-networks.html>, Erişim tarihi: 15 02 2016.
- [75] R. Battiti, First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method, içinde, *Neural Computation*, cilt 4, no. 2, pp. 141-166, 1992.

EKLER

Bu çalışmada kullanılan bütün kodlar CD içerisinde aşağıdaki adreslerde bulunmaktadır.

EK 1: CD\Kinect Kodları

EK 2: CD\Matlab Kodları

EK 3: CD\VAL3 Kodları



ÖZGEÇMİŞ

Ad-Soyad : Burak MERT
Uyruđu : T.C
Dođum Tarihi ve Yeri : Gaziantep, 05/01/1989
E-posta : bmert88@gmail.com

ÖĐRENİM DURUMU:

- **Lisans** : 2010, İhsan Doğramacı Bilkent Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliđi

YABANCI DİL:

İngilizce

DİĐER YAYINLAR, SUNUMLAR VE PATENTLER:

- B. Mert, U. Aradag, S. Uludag and H. O. Unver, "An architecture for a microgrid-based eco industrial park using a Multi-Agent System," *Power Engineering, Energy and Electrical Drives (POWERENG), 2013 Fourth International Conference on*, Istanbul, 2013, pp. 1658-1664.
- U. Aradag, B. Mert, G. Demirel, S. Uludag, H.O. Unver, S. Aradag, "A framework for a multiagent-based virtual enterprise with a microgrid energy market model", 11th Global Conference on Sustainable Manufacturing, 2013, Berlin