

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**HAPTİK TELE-OPERASYONA YÖNELİK İŞ UZAYI EŞLEME
YÖNTEMLERİNİN İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Ahmet AKSEL

Makine Mühendisliği Anabilim Dalı

Tez Danışmanı: Yrd. Doç. Dr. Yiğit TAŞCIOĞLU

AĞUSTOS 2017

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

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

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 141511021 numaralı Yüksek Lisans Öğrencisi **Ahmet AKSEL**'in ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "**HAPTİK TELE-OPERASYONA YÖNELİK İŞ UZAYI EŞLEME YÖNTEMLERİNİN İNCELENMESİ**" başlıklı tezi **18.08.2017** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Yrd. Doç. Dr. Yiğit TAŞCIOĞLU**
TOBB Ekonomi ve Teknoloji Üniversitesi

Eş Danışman : **Prof. Dr. Yücel ERCAN**
TOBB Ekonomi ve Teknoloji Üniversitesi

Jüri Üyeleri : **Doç. Dr. Coşku KASNAKOĞLU (Başkan)**
TOBB Ekonomik ve Teknoloji Üniversitesi

Yrd. Doç. Dr. Ali Emre TURGUT
Orta Doğu Teknik 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.

Ahmet AKSEL

ÖZET

Yüksek Lisans Tezi

HAPTİK TELE-OPERASYONA YÖNELİK İŞ UZAYI EŞLEME

YÖNTEMLERİNİN İNCELENMESİ

Ahmet AKSEL

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Makine Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Yiğit TAŞCIOĞLU

Tarih: Ağustos 2017

Merkez/uydu tele-operasyon sistemleri on yıllardır çeşitli araştırma alanlarında kullanılmış ve bilgisayar teknolojilerindeki ilerleme daha küçük boyuttaki merkez cihazların kullanımına olanak sağlamıştır. Ancak, bu gelişmeler, merkez ve uydu robotların iş uzayları arasındaki eşlemeyi zorlaştırmıştır. Sistemin kinematik ve dinamik modeli ile iş uzayı boyutları bu eşleme yöntemlerinin uygulanmasında göz önüne alınması gereken temel unsurlardır. hız, konum, balistik ve sınır kaydırma kontrolü gibi birbirinden farklı birçok kontrol yöntemi literatürde önerilmiş ve incelenmiştir. Bu çalışmada, uygulanmaları diğer yöntemlere göre daha anlaşılır ve basit olması sebebiyle konum, hız, geçişli konum-hız kontrolü, balistik kontrol ve son olarak değişken ölçekli konum kontrolü incelenmiştir. İnsan deneylerinde kullanıcılardan, eşleme yöntemlerinin karşılaştırılmasına uygun bir teleoperasyon görevinin yapılması istenmiş ve ergonomi, kullanılabilirlik gibi niteliksel geri bildirimler alınıp deney verilerinden elde edilen sayısal sonuçlarla birlikte yorumlanmıştır. Phantom Omni haptik cihaz merkez ve Stäubli RX160 robot kol da uydu olarak kullanılmıştır. Bilgisayar üzerinde çalışan yazılım, hafif ve neredeyse bütün sistemlerde çalışabilen evrensel bir programlama dili olan C/C++ ile yazılmış

ve açık kaynak kodlu kütüphane olan OpenHaptics kullanılmıştır. Bu şekilde haberleşme gecikmelerinin de en aza indirilmesi amaçlanmıştır.

Anahtar Kelimeler: Teleoperasyon, Haptik, Robotik, Kuvvet geri besleme



ABSTRACT

Master of Science

EXAMINATION OF WORKSPACE MAPPING METHODS FOR HAPTIC

TELEOPERATION

Ahmet AKSEL

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

Supervisor: Associate Prof. Dr. Yiğit TAŞCIOĞLU

Date: August 2017

Master-slave teleoperation systems is being used in many research areas for decades and the advances in computer technologies allow smaller master devices to be used. However, these advances made it harder to map between master and slave robots. The kinematic and dynamic models of the system and workspace sizes are important factors to apply mapping methods. Rate, position, ballistic control and boundary drift control are the methods which are studied in literature. Rate control, position control, modified rate and position control, ballistic control and variable scaling control are used in this study because of their easy adaptation. For human experiments, users are given a mission appropriate to all mapping methods and qualitative information in regard to ergonomics and usability as well as data from distance sensors are collected and interpreted. Phantom Omni haptic device is used as master and Stäubli RX160 robot arm used as slave robot. The applications running on main computer is written and compiled by C/C++ programming language, which is a lightweight programming language and compatible for almost all operating systems. OpenHaptics Toolkit which is an open source library is used for communication of the computer with the haptic device. This way communication delays aimed to be lowered.

Keywords: Teleoperation, Haptic, Robotic, Force-feedback.



TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren hocalarım Yięit TAŐCIOęLU ve Yücel ERCAN'a, kıymetli tecrübelerinden faydalandıęım TOBB Ekonomi ve Teknoloji Üniversitesi Makine Mühendislięi Bölümü öğretim üyelerine ve destekleriyle her zaman yanımda olan aileme ve arkadaşlarıma çok teőekkür ederim.

Ayrıca yüksek lisans süresince bana burs saęlayan TOBB Ekonomi ve Teknoloji Üniversitesi'ne teőekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vii
ABSTRACT	ix
TEŞEKKÜR	xi
İÇİNDEKİLER	xiii
ŞEKİL LİSTESİ	xv
ÇİZELGE LİSTESİ	xvii
KISALTMALAR	xix
SEMBOL LİSTESİ	xxi
1. GİRİŞ	1
1.1 Tezin Amacı ve Yenilikçiliği	11
1.2 Tezin Kapsamı.....	11
2. SİSTEM YAPISI VE BİLEŞENLERİ	13
2.1 Model	13
2.2 İş Uzayı Eşleme ve Kontrol Yöntemleri	14
2.2.1 Konum kontrolü	14
2.2.2 Hız kontrolü	16
2.2.3 Geçişli konum-hız kontrolü	18
2.2.4 Balistik kontrol.....	19
2.2.5 Değişken ölçekli konum kontrolü	19
3. TEST DÜZENEGİ TASARIMI	21
3.1 Test Düzeneği Bileşenleri	21
3.1.1 Phantom Omni haptik cihaz	22
3.1.2 Stäubli RX160 robot kolu	25
3.1.3 Sensör altsistemi	27
3.2 Yörünge Tasarımı.....	28
3.3 Sistemin Çalıştırılması	29
4. DENEYLER	37
5. SONUÇLAR VE TARTIŞMA	43
KAYNAKLAR	49
ÖZGEÇMİŞ	53

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Sistem modeli.	13
Şekil 2.2 : Konum kontrolü ile haptik cihaz ve robot kol arasındaki iş uzayı eşlemesi.	15
Şekil 2.3 : Hız kontrolündeki kontrol bölgeleri.	16
Şekil 2.4 : Geçişli konum-hız kontrolündeki kontrol bölgeleri.....	18
Şekil 3.1 : Deney düzeneği.	21
Şekil 3.2 : Phantom Omni haptik cihaz.....	22
Şekil 3.3 : Phantom Omni haptik cihazın eklem açıları ve uzunlukları.....	23
Şekil 3.4 : Phantom Omni ters kinematığı.	24
Şekil 3.5 : Stäubli RX160 robot kolun eklemleri.....	25
Şekil 3.6 : Stäubli RX160 robot kolun bağ çerçeveleri.....	26
Şekil 3.7 : CS8C Robot kontrolcü (a) ve el kumandası (b).....	26
Şekil 3.8 : Raspberry Pi ve uzaklık algılayıcıların robot üzerine yerleştirilmiş hali..	27
Şekil 3.9 : VL6180X uzaklık sensörünün tipik menzil performansı [34].....	28
Şekil 3.10 : Robot kol için belirlenen yörünge.	29
Şekil 3.11 : Raspberry Pi, TCA9548A çoklayıcı ve sensörler arasındaki bağlantılar.	30
Şekil 3.12 : Algoritmanın genel çalışması.	31
Şekil 3.13 : Teleoperasyon sisteminin konfigürasyonu ve çalıştırılması.....	33
Şekil 3.14 : Robot kontrolcü soket konfigürasyonu.....	34
Şekil 3.15 : Robot kontrolcü ağ konfigürasyonu.	34
Şekil 3.16 : Bilgisayarda çalışan programın C++ kodunun bir parçası.	35
Şekil 3.17 : Bilgisayarın ağ ayarları.	35
Şekil 3.18 : Raspberry Pi üzerinde çalışan python kod parçası.	36
Şekil 4.1 : Kuvvet geri beslemeli yörünge ve labirent ölçüleri.....	40
Şekil 5.1 : Kullanıcıların kuvvet geri besleme miktarlarına 5 üzerinden verdikleri puanlar.	43
Şekil 5.2 : Kontrol yöntemleri ve kuvvet miktarlarının ayrı ayrı puan değerleri.....	44
Şekil 5.3 : Kontrol yöntemlerinin hata-bitirme zamanına göre kendi içinde analizi. 45	
Şekil 5.4 : Kuvvet miktarlarının hata miktarı – bitirme zamanına göre performans grafığı.	46
Şekil 5.5 : Kontrol yöntemlerinin karşılaştırmalı performans değerleri.	47

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1 : Stäubli RX-160 robot kolun DH parametreleri.....	27
Çizelge 4.1 : Rastgele 1-15 arası değerleri içeren matris üretici algoritmanın sahte kodu.	38
Çizelge 4.2 : Deney sıralamaları.	38
Çizelge 4.3 : Algılayıcı datasını filtreleyen ve yörüngeden kaçıklığı hesaplayan algoritmanın sahte kodu.....	42

KISALTMALAR

DH : Denavit-Hartenberg
GKH : Geçişli Konum-Hız



SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklama

mm	milimetre
N	Newton
ms	milisaniye
\hat{R}	Dönme matrisi
\hat{k}_g	Genel katsayı matrisi
k_p	Konum kontrolü eşleme katsayısı
k_r	Hız kontrolü eşleme katsayısı
k_b	Balistik kontrol eşleme katsayısı
k_v	Konum kontrolü için değişken eşleme katsayısı
r_s	Ölü bölgenin yarıçapı
$\bar{p}_t^{(r)}$	Robot kolun konumu
$\bar{p}_o^{(r)}$	Robot kolun başlangıç konumu
$\Delta\bar{p}_t^{(r)}$	Robot kolun hızı
$\bar{p}_t^{(h)}$	Haptik cihaz uç noktasının konumu
$\bar{p}_n^{(h)}$	Haptik cihaz uç noktasının anlık konumu (1ms hassasiyet)
$\bar{p}_{n-1}^{(h)}$	Haptik cihaz uç noktasının bir önceki andaki konumu

1. GİRİŞ

Günümüzde robotlar endüstri, medikal, rehabilitasyon gibi birçok alanda kullanılmaktadır. Yıllar içinde robotların çeşitli alanlarda kullanımına yönelik yapılan çalışmalar literatürde de yerini almıştır. Endüstriyel robotlar üzerinde araştırma ve geliştirme çalışmaları 1950 lerdeki nümerik kontrollü takım tezgâhlarına kadar uzanmaktadır. İlk endüstriyel robotlar 1960 larda kullanılmaya öncelikle malzeme taşıma ve otomasyonlu makine desteği ve sonrasında da punta kaynağı için kullanılmıştır. 1970 lerde robotlar çoğunlukla hidrolik tahrikli idi, fakat ark kaynağı gibi uygulamaların gereksinimlerinden dolayı elektrik tahrikli robotların geliştirilmesine öncelik verildi. 1973 'te ASEA (Şimdiki ABB Robotics) tarafından tamamen elektrik tahrikli ark kaynağı ve punta kaynağı robotları yapıldı. Bunların hareket kontrolü de halen basit noktadan noktaya kontrole benzer yapıdaydı. 70 lerin ortalarından itibaren çapak alma, cilalama, yapıştırma, döküm temizleme ve buna benzer diğer robot uygulamaları üzerinde testler ve değerlendirmeler yapıldı. Çevresel ekipmanlar geliştirilip bunların robotik istasyonlar üzerine yerleştirilmesi ve robot kontrolcülere entegrasyonu da bu çalışmalar dahilindedir. Daha sonra 1980 lerde yeni kontrolcülerle beraber Sürekli Yörünge kontrolü kullanılmaya başlandı. Bunun en büyük avantajlarından biri de artan hesaplama gücü sayesinde robot kinematiğinin kontrolcüye çözdürülebilmesi sonucunda farklı koordinat sistemlerinde robot hareketinin tanımlanabilmesine olanak tanınmasıydı. 1980 lerin sonuna doğru simülasyon ve çevrimdışı programlama kullanılmaya başlandı. 1990 larda, simülasyon ve çevrimdışı programlamanın, nesne tabanlı programlama, manipülasyon ve robot görevinin detaylı analizi ile beraber kullanılmasıyla da robot teknolojisinde ileriye gidilmeye devam edildi [1].

Günümüzde pratik robotik uygulamalarda kullanılan birçok programlama yöntemi mevcuttur. Bunlar temelde çevrimiçi ve çevrim-dışı programlama (ÇDP) olmak üzere iki ana kategoriye ayrılabilir. Çevrimiçi programlama genellikle kalifiye bir operatör tarafından robotun istenilen bir yörüngede el kumandası yardımı ile hareket ettirilip yörünge üzerinde belli noktaların kaydedilmesi ile yapılır. Bu yöntemin çeşitli

dezavantajları mevcuttur. Öncelikle robotun üzerinde çok fazla sayıda koordinat sistemi olduğu için operatör sürekli olarak hangi çerçevede hareket ettiğinin bilincinde olmak durumundadır. Bunun yanında robotu istenilen hassasiyette hareket ettirmek ve aynı zamanda çarpışmalardan kaçınmak özellikle iş parçası veya işlem karmaşıkla oldukça zor ve zaman alıcı bir süreçtir. Ayrıca program oluşturulduktan sonra güvenlik ve emniyet sebeplerinden dolayı birçok kez test edilmesi gerekmektedir. Bütün bu dezavantajlarına rağmen çoğu küçük ve orta ölçekli işletmede çevrimiçi programlama kullanılır. Daha sezgisel algılama yöntemleri kullanan destekli çevrimiçi programlama yöntemi de operatör destekli çevrimiçi programlama ve sensör yönlendirmeli çevrimiçi programlama olmak üzere ikiye ayrılabilir. Operatör destekli çevrimiçi programlama ile ilgili daha sezgisel olması için birkaç destekleyici öğretim aygıtı geliştirilmiştir. Sugita [2] çapak alma ve ince işlem için öğretim destekli aygıtlar geliştirerek öğretim metodu uygulamıştır. Uç işlem aygıtının konum ve yönelim vektörünü ölçmek üzere iki adet öğretici destek aygıtı kullanılarak robot programı kendi koordinat sisteminde oluşturulmuştur. Choi [3] ise COSMO isminde kuvvet/moment vektör yönünü ölçen bir sensör geliştirerek el kumandası ile öğretmeyi bir kademe öteye taşımıştır. Bu sensörün testlerinde 6 serbestlik dereceli bir robot kola yerleştirilen sensör, operatör tarafından tutulur ve istenilen hareket yönünde sensörü itme, çekme ve burkma ile yönlendirir. Bu sezgisel robot öğretim yöntemi oldukça düşük maliyetlidir. Schraft [4] daha hızlı ve etkili bir öğretim için sezgisel bir öğretim metodu geliştirmiştir. Kullanıcı üzerine kuvvet/tork sensörü yerleştirilmiş bir tutacak ile robotu yönlendirir ve sesli diyalog sistemi ile de emir verir. Destekli öğretim cihazları kullanmak genellikle hâlihazırda oldukça karışık olan robotik sistemine fazladan sensörler ve kalibrasyon süreçleri eklemek anlamına gelmektedir. Pan [5], ABB 'nin isteğe bağlı kuvvet kontrolü özelliği olan IRC5 kontrolcüsünü temel alıp rehberlik ile programlama yöntemi geliştirdi. Kuvvet kontrolcüsünün sağladığı iki temel fonksiyon vardır. Birincisi, kuvvet kontrolü yönünde uyumlu olup harekete izin vermesi fakat diğer yönlerde (konum kontrol yönleri) katı olup hareketi engellemesidir. Operatör tek elle robotu itmek veya sürüklemek suretiyle robotu istenilen konum ve yönetime getirebilir. İkincisi ise yörünge öğrenme olarak bilinen robotun yörüngeye dik doğrultularda hareketinin serbest olması sayesinde uç işlem aygıtının iş parçası ile sürekli temas halinde olmasını sağlamasıdır. Bu iki temel fonksiyon sayesinde bütün programlama süreci çarpışmasız ve otomatiktir. Pan ve Zhang [6]'ın yöntemi programlamanın ilk aşamasında robotun el ile yönlendirilmesini

gerektirmekteyken diğ er bazı arařtırmacılar farklı sensör teknolojilerini de olaya dâhil ederek ilk bařtaki el ile kumanda sürecini ortadan kaldırmıřlardır. Zhang [6] aynı platformun üzerine görsel kontrol ekleyerek konsepti daha da genişletmiřtir. Ç eřitli sensörlerden gelen geribesleme ile robotun farklı yönlerdeki hareketlerinin kontrolü için melez bir konum/kuvvet/görsel kontrol platformu geliřtirilmiřtir. Keç eli kalem ile çizilen bir yörüngeyi takip ederek otomatik program oluřturabilen bu sistemde konum kontrolü uç iřlem aygıtının konum ve yönelimini muhafaza etmek için, görsel algılama eğriyi takip etmek için, kuvvet kontrolü de uç iřlem aygıtı ile iř parçası arasındaki temasın kopmaması için kullanılmaktadır. Solvang [7] da iř parçası üzerine çizilen yolu tanımaya dayalı görsel tabanlı bir programlama yöntemi geliřtirmiřtir. Ç izilen rotanın x ve y eksenindeki koordinatları bir kamera ile kaydedilerek belirlenir. Rota üzerinde belli noktalarda “dokunup geri çekil” taktiğ i kullanılarak da yörünge üzerindeki noktaların z eksenindeki koordinatları belirlenir. Böylece 3 boyutlu uzayda yörünge belirlenmiř olur. Nicholson [8] kaynak tamiri iřleri için görüntü verileri kullanan bir seri robot programlama yöntemi geliřtirdi. Bu yöntemde kullanıcı iř parçasının üzerine çizmek yerine resim üzerinden 2 boyutlu rotaya karar verir. Kaynaklama sisteminin üzerinde bulunan dokunma algılayıcı ile de z koordinatı belirlenir. Bu yöntem, ölçümlene sonuçlarına güvenen ve ıřıklandırma durumuna hassas olan diğ er görsel tabanlı sistemlerden farklı olarak, basitliğ inden dolayı gürbüz bir yapıdadır. Çevrimdışı programlama yöntemleri iř parçasının üç boyutlu modeli kullanılarak genellikle yüksek ürün hacmindeki otomasyon sistemlerinde kullanılır. Çevrimiçi programlamaya göre çevrimdışı programlamanın birçoğ u avantajı vardır. Birincisi, programlama sürecinde esas robota ihtiyaç olmadığından robotun çalışmadığı süre minimuma indirilmiř olur. İkincisi, çevrimdışı olarak üretilen programlar kumanda ile öğ retme yöntemine göre daha esnektir. Üçüncüsü, simülasyon genelde çevrimdışı programlamaya dahil edilir. Bunun sonucu olarak program önceden kontrol edilip hata ihtimali en aza indirilip verimlilik ve güvenlik geliřtirilmiř olur. Çevrimdışı programlamanın bu gibi avantajları olmasına rağmen belli dezavantajlarından ötürü küçük ve orta ölçekli iřletmeler tarafından tercih edilmez. Çevrimdışı programlama ile özelleřtirilmiř yazılım geliřtirmek zaman alıcı ve yüksek seviye programlama becerisi gerektiren bir süreçtir. Çevrimdışı programlama yazılım sađlayıcıları çevrimdışı programlama paketlerini özel uygulamalar için yapılacak ikincil geliřtirmeleri azaltmak için daha güçlü, modüler ve esnek hale getirmek için

çalışırken, akademik araştırmacılar da iyileştirilmiş süreç planlama algoritmaları üzerine yoğunlaştılar ve açık kaynak kodlu birkaç paket geliştirdiler [9].

Dokunma bilimi, dokunarak bilgi toplama ve nesne manipülasyonu ile ilgilenir. Haptik arayüzler kullanıcıların bir sanal ortamda veya uzaktan kontrol sisteminde dokunup hissetmesine ve nesnelere manipüle etmesine olanak sağlar. Örneğin klavye ve fare kullanıcısının el hareketlerini algılayan bilinen pasif haptik arayüzlerdendir. Bunlar temas sırasında kullanıcının eline bir kuvvet uygulayıp dokunsal bir his verse de bu kuvvet bir program ile kontrol edilemez. Masaüstü robotları ve kuvvet geribeslemeli dış iskeletsel eldivenler gibi aktif haptik arayüzler ise hem algılayıcıları hem de eyleticileri bulunan daha gelişmiş cihazlardır. Bu cihazlar kullanıcılardan konum ve hareket bilgilerini almanın yanı sıra sanal nesnelere algılamalarına, hissetmelerine ve hatta kontrol etmelerine olanak sağlayacak şekilde kullanıcıya kontrol edilmiş bir kuvvet de verebilirler. Aktif haptik arayüzler tıp, eğlence, eğitim, endüstri ve güzel sanatlar gibi birçok alanda kullanılmaktadır [10].

Günümüzde haptik cihazlar empedans ve admitans olmak üzere iki temel kontrol etkileşimini kullanırlar. Empedans kontrolünde kullanıcı cihazı hareket ettirir ve cihaz aldığı verileri bilgisayara geri gönderir; dolayısıyla uygulama geri bildirim kontrol etmekten sorumludur. Geomagic firmasının in ürettiği Phantom ve Novint firmasının ürettiği Falcon bu tip cihazlara örnek olarak verilebilir. Tam tersi, admitans kontrolünde kullanıcı cihaza bir kuvvet uygular, cihaz da bu kuvvete oranla yer değiştirir. Bu tip kontrolü sağlayan cihazlar genellikle karmaşık ve büyük olduğundan ve insanlarla güvenli bir şekilde etkileşimde bulunması için dikkatli bir şekilde tasarlanması gerektiğinden dolayı eğitim alanında çok kullanılmaz. Bu tip cihazlara örnek olarak da MOOG' un ürettiği HapticMaster[11] verilebilir.

Tıp alanındaki prosedürler genellikle özel aletlerle organ veya dokuları dokunma hissinin de yardımıyla manipüle etmeyi kapsar. Ameliyat prosedürleri yüksek zorluk ve karışıklık seviyesine sahiptir. Öğrencilere veya pratisyenlere ameliyatları düzgün bir şekilde yapabilmeleri için doğru ve yoğun bir eğitim verilmesi gerekmektedir. Bu yüzden bilgisayarın icadından itibaren araştırmacılar öğretim süreçlerindeki maliyetleri azaltmak için yeni yollar aramışlardır. Haptik cihazlar bu bağlamda tıbbi eğitimlerde kullanılmak üzere çok iyi birer alternatif olmuştur. Dikiş atma prosedürü simülasyonları haptik teknolojisinin uygulandığı alanlardan biridir. Jia ve Pan [12]

nesnelerin ve dikiş ipliğinin deformasyonunu ve etkileşimini benzetim yapan bir dikiş simülatörü geliştirmişlerdir [13].

Bazı durumlarda merkez ve uydu robot iş uzayları ve kinematik yapıları birbirinden farklı olabilir. Bu da bazı eşleme senaryolarında teklik durumlarıyla karşılaşılmasına sebep olabilir. Bu gibi durumlardan kaçınmak ve aynı zamanda eşlenen uzayın olabilecek en büyük hacimde seçilebilmesi için eşleme değişkenlerinin iyi bir şekilde ayarlanması gerekmektedir. Bu değişkenlerin en iyi şekilde seçilebilmesi için [14]'deki gibi ortogonal deney tasarım yöntemi kullanılabilir.

Bazı durumlarda teleoperasyon için kullanılan merkez robotun serbestlik derecesi uydu robotun serbestlik derecesinden düşük olabilir. Bu gibi durumlarda da bu eksikliği kapatmak için çeşitli senaryolar geliştirilebilir. Örneğin uzaktaki robotun 6 serbestlik derecesi varsa ve bir joystick ile kontrol ediliyorsa öteleme ve dönel hareket arasında geçiş ekstra bir tuş, pedal gibi bir girdi ile yönetilebilir. Ayrıca yazılımsal olarak robotun bazı hareketleri de kısıtlanabilir. Bu sayede hem konum eşleme hem de hız eşleme olası bir teklik durumundan kaçınılabilir. Kullanılan iletişim protokolünün de kullanıcının gereksinimlerine bağlı olarak seçilmesi daha uygundur. [15]'te yapılan çalışma uzaktan birleştirme işlemi için yapılmıştır.

Bazı çalışmalarda uzaktan kontrol için Kinect gibi jest algılama aygıtları kullanılmıştır. Sınırlandırılmış iş uzayı sorununu ortadan kaldırmak ve kullanım kolaylığı açısından sürüklenme eşlemesi kullanılmıştır [16]. Bu yöntemde operatörün eli durduğunda, robot en son verilen hız komutuna göre hareket etmeye devam eder. Fakat bu yöntemin zorluğu da durmadan önceki son hız komutunun küçük değerler olmasıdır. Bu yüzden sürüklenme hızı oldukça düşük olur. Bundan kaçınmak için maksimum hız eşleme gibi yöntemler kullanılabilir.

[17]'de yapılan çalışma genel olarak kullanıcı yönlendirmeli teleoperasyon üzerine yapılmıştır. Önerilen yardımcı eşleme yönteminin kullanıcı üzerindeki yorucu etkisini azaltmaya yöneliktir. Tekrar senkronizasyon , diğer bir deyişle indeksleme metodu, bilgisayar faresinin atlamasına benzer. Burada konum kontrol modu kullanılır. Belli bir katsayı ile çarpılan haptik cihaz yer değiştirme bilgisi robota aktarılır. Eğer haptik cihaz işuzayı limitlerine ulaşırsa cihaz üzerinden bir tuşa basılarak senkronizasyon koparılıp tekrar konumlandırma yapılır. Tuş tekrar bırakılarak konum kontrolü modu ile devam edilir. Balistik takip metodunda konum

kontrolünden farklı olarak ölçekleme katsayısı haptik cihazın hızı ile orantılıdır. Bu sayede hız kontrol edilerek daha hassas hareketler ve daha kaba hareketler karşı tarafta elde edilebilir. Hız kontrolünde ise haptik cihazın konumu slave robotun hızına eşlenmektedir. Her bir yöntemin avantajları ve dezavantajları mevcuttur. Örneğin tekrar senkronize etme yöntemini kullanmak için ekstra bir butona ihtiyaç vardır. Ayrıca sürekli hareket gerektiren bir görevde hareket kesintisi olacağı için sorun oluşturabilir. Balistik takip yönteminin avantajı ise otomatik olarak ölçekleme katsayısını ayarlamasıdır fakat kullanıcı bir yönde farklı karşı yönde farklı hızlarla hareket ettiği zaman bir ofset oluşur. Bu da keşif bölgesindeki kontrolü azaltır. Hız kontrolü kaba hareketler için sezgisel ve kullanışlı bir yöntemdir. Fakat, hassas hareketler için uygun değildir. Bu çalışmada haptik yardımcı sistem üzerinde daha çok durulmuştur. Kullanıcının işini kolaylaştıracak şekilde bir kuvvet alanı oluşturularak yapılan göreve destek olunmaktadır. Yapılan deneyler haptik yardımcı sistemin yüzde 20 – 35 arasında verim artışı sağladığını göstermiştir. Değerlendirme farklı eşleme yöntemlerini karşılaştırmak için yapılmamıştır.

Sürüklenme eşlemesi dışında başka eşleme yöntemleri de yapılan çalışmalarda kullanılmıştır. [17]' de yapılan çalışmada, teleoperasyon işlemini kolaylaştırmak için tekrarlanan senkronizasyon yöntemi geliştirilmiştir. Bu yöntemin çalışma mantığı basitçe bilgisayar faresinin hareketine benzer [18]. Hareketin eşlenmesi için konum eşlemesi kullanılır. Haptik iş uzayının sınırlarına gelindiğindeyse senkronizasyon serbest bırakılıp (herhangi bir buton veya pedal vasıtasıyla olabilir) haptik cihaz yeniden konumlandırılır (Bu sırada robot hareketsizdir) ve uygun bir konumda tekrar senkronizasyon başlatılır. Bu yöntemin zorluklarından biri de birbirinden ayırmak için buton veya pedal gibi fazladan bir girdiye ihtiyaç duyulmasıdır.

Bu kadar çeşitli iş uzayı eşleme yöntemi geliştirildikten sonra bunların karşılaştırılmasının da yapılması gerekmektedir. Karşılaştırma yapılabilmesi için öncelikle bir test düzeneği geliştirilmesi ve bir görev tanımlanması gerekmektedir. [19]' da yapılan çalışmada görev tanımları şu şekilde verilmiştir:

1. 5 adet somun bir mile geçirilmiştir. Operatör bunları tek tek alıp başka bir mile geçirir.
2. Operatör zeminden aldığı 5 adet silindirik nesneyi üst üste yerleştirir.
3. Operatör içi sıvı dolu küçük bir şişeyi alıp bir kaba boşaltır.

Bu görevler hareket karmaşıklığı ve iş zorluğu bakımından farklılık göstermektedir. Seçilen bir operatör grubuna bu görevler yaptırılır. Daha sonra toplanan verilere varyasyon analizi (ANOVA) uygulanır.

Robot manipulatörlerle ilgili çalışmalara bakılacak olursa öncelikle robotik alanındaki gelişmelerden bahsetmek gerekir. Bunların arasında Denavit-Hartenberg [20]' in geliştirdiği parametrelendirme yöntemi en çok kullanılan yöntemlerden biridir. DH parametreleri bir robot manipulatördeki ardışık uzuvların aralarındaki ilişkiyi ifade eden dört adet parametreden oluşur. Daha sonra Richard Paul [21] DH parametrelerini kendine göre tekrar tanımlamaya başlamıştır. O zamanlar noktadan noktaya kontrol kullanan bu sade robotların parametrelerinin robot kinematiği çözümündeki önemini ortaya koymuştur.

Haptik teknolojisinin sanal gerçeklik için kullanımı ürün tasarımı ve geliştirme aşamalarında yardımcı olma potansiyeline sahiptir. Sanal gerçeklik – haptik sistemi sanal objeleri yüzey pürüzlülüğü, büyüklük, şekil ve ağırlık açısından karşılaştırmada kullanılabilir. [22]' de yapılan çalışmada Phantom Omni'nin kullanılabilirliğini (kolaylık, hassasiyet, güvenilirlik, doğruluk vb.) ve haptik cihazla 3-Boyutlu nesnelerin yüzey kalitesi, sertliği ve ağırlığı gibi özelliklerini anlamak için bir deney tasarlanmıştır. Data toplama kısmında öncelikle anketler oluşturulmuştur. Bu yüzden, 5 puanlı Likert ölçeği ile haptik geri beslemeli sistemin kullanılabilirliğini anlamak, sanal nesnelerin özelliklerini değerlendirmek ve katılımcılardan genel bilgi (demografik bilgiler, akademik arkaplan vb.) almak için katılımcılara açık uçlu sorular sorulmuştur. Daha sonra 15 öğrenci üzerinde anketin güçlülüğünü ölçmek için pilot data toplanmıştır. Buradan kuvvet geri beslemesi ile ilgili olumlu geri dönüş alınmıştır. Deneyler ve anket soruları son data toplamadan önce tekrar modifiye edilmiştir. Son datalar toplanırken 84 öğrenciden faydalanılmıştır.

[23]' te yapılan çalışmada modifiye edilmiş bir master-slave işuzayı eşleme yöntemi geliştirilmiştir. Bu yöntem, hız kontrolü ve değişken ölçekli konum eşleme yöntemlerini biraraya getirmektedir. Sistemin kararlılığını güvence altına almak için master tarafında ölü bir bölge tanımlanmıştır. Bu yöntem daha sonra hız kontrolü ve değişken ölçekli konum eşleme yöntemleri ile karşılaştırılmıştır.

Hız kontrolü arabanın gaz pedalından esinlenilerek oluşturulmuştur. Merkez etrafında oluşturulan ölü bölge el ile yanlış operasyon yapılmasını önlemek içindir. Bu

yöntemde r_{min} ile r_{max} arasındaki bölgede haptik cihazın konumu belli bir katsayı ile çarpılıp slave robota hız komutu olarak gönderilir. Bu yöntemdeki kuvvet geri beslemesi ise hızı daha rahat kontrol altına almak için yay şeklinde modellenip merkeze doğru verilmiştir ve merkezden uzaklıkla doğru orantılıdır. Yani hız arttıkça kullanıcıya daha fazla kuvvet uygulanarak hızlanmak zorlaştırılmıştır. Böylece güvenli bir işlem sağlanmış olur.

Değişken ölçekli konum eşlemede merkez robot konumu ile uydu robotun konumu kontrol ediliyor. Buradaki yöntemde aşağıdaki formül baz alınarak eşleme yapılmaktadır.

$$P_s = k_{scale} \cdot P_m \quad (1.1)$$

k_{scale} 0~1 aralığında olup haptik cihaz üzerindeki iki buton yardımıyla artırılıp azaltılabilmektedir.

[24]' te yapılan çalışmada konum kontrolü ve düzenlenmiş hız kontrolü bir butonla birbirine geçiş yapmaktadır. Robotun anlık konumu ile haptik cihazdan gönderilen referans konum arasında sanal bir yay oluşturulmuştur. Sonuç olarak konum kontrolü ile hız kontrolü karıştırılıp yeni bir yöntem bulunmuştur.

[25]' teki çalışmada sınır kaydırma kontrolü olarak ifade ettikleri ve kendi geliştirdikleri işuzayı eşleme yöntemini birtakım testler vasıtasıyla verimlilik ve doğruluk açısından değerlendirmişlerdir. Daha sonra sonuçların istatistiksel önemi varyans analizi (ANOVA) ile saptanmıştır. Yeni geliştirilen yöntemin hem hız hem de doğruluk açısından uygun bir seçim olduğu görülmüştür. Master tarafında Phantom Omni haptik cihaz ve slave tarafında da 6 serbestlik dereceli Kuka KR-6 endüstriyel robot kullanılmıştır. Bilgisayardaki konsol programı her 10 ms de bir haptik cihazdan alınan konum ve oryantasyon bilgilerine ve o anda kullanılan eşleme yöntemine göre kuka robota gönderilecek konum komutlarını hesaplar.

Sabit ölçekli eşleme, değişken ölçekli eşleme, joystick kontrol, sınır kaydırma yöntemi bu çalışmada karşılaştırılan yöntemlerdir.

DeneySEL değerlendirme kısmı için 3 adet görev oluşturulmuştur. Birincisinde kullanıcıdan bir mile geçirilmiş 5 adet somunu tek tek alıp diğer mile geçirmesi istenmekte. İkincisinde, 5 adet yakıt saçmasını üst üste koyması bekleniyor. Üçüncüsünde ise sıvı dolu ufak bir şişe tutulup içindeki sıvı toplayıcı kaba boş şişe de

çöp kutusuna atılacaktır. Her bir görev için uygun görülmeyen olaylar hata olarak tanımlanmıştır.

Deney sonuçları değerlendirilirken de varyans analizi her bir görev için kullanılmıştır. Ayrıca kullanıcılar da bir haftaya yayılı 5 oturumda bu eşleme yöntemlerinde ustalaşmışlardır. Her oturumda kullanıcı kendisine verilen eşleme yöntemi ile 3 görevi de 10'ar kere tekrarlamıştır.

Sonuç olarak hız bazında Joystick kontrol en iyisidir ve sabit ölçekli eşleme de neredeyse joystick kontrol kadar hızlıdır. Ancak, görevler hata kabul etmeyecek şekilde olursa değişken ölçekli eşleme veya sınır kaydırma seçilebilir. Sınır kaydırma hız ve doğruluk açısından çoğu durumda iyi sonuçlar verebilmektedir, fakat öğrenmesi biraz zor bir yöntemdir. Sabit ölçekli eşleme yöntemi hataya en yatkın yöntemdir. 1 haftalık pratik sayesinde kullanıcılar daha yüksek bir doğrulukta ve hızda görevleri tamamlamışlar, ayrıca hassas görevlerdeki hataları da azalmıştır. Fakat pratik yüzünden farklı eşleme yöntemleri arasındaki farklar da azaldığından göreceli performansların karşılaştırılması zorlaşmıştır. Kullanıcılar açısından sabit ölçekli eşleme en hızlı öğrenilen yöntem olmasının yanında sınır kaydırma yöntemi öğrenmesi zor bir yöntem olsa da en rahat idare edilen yöntem olmuştur.

[26]' da yapılan çalışma haptik kontrolü bütünüyle ele almıştır. Operatörün kontrolcü tasarımında nasıl modellenebileceği, haptik performansın nasıl inceleneceği detaylı bir şekilde açıklanmıştır.

[27]' de yapılan çalışmada haptik sistemler ve bunların sanal gerçeklik ile kullanımı ve kapsamı ele alınmaktadır.

[28]' de yapılan çalışmada bütün farklı manipülatör modellerinin kendine has DH tablosu var ve sistem hangi manipülatör kontrol ediliyorsa ona göre hesaplama yapıyor. Böylece farklı bir manipülatöre geçince işuzayında ve eşlemede uyumsuzluk olması gibi bir problem de ortadan kalkıyor. Konum ve hız kontrolü kullanılmıştır. DH tablolarına göre Java programlama dili kullanılarak konum ve hız çözücü yapılmıştır. Simülatör sayesinde farklı senaryo ve kontrol yöntemleri denenerek deniz vinçlerinin kullanımındaki riskler azaltılmaktadır.

[29]' da yapılan çalışmada PHANTOM cihazın tanıtımını yapmışlardır. Kişisel haptik arayüz mekanizması (PHANTOM) düşük maliyetli ve güzel haptik sonuçlar veren bir sistemdir.

[30]' da yapılan çalışmadaki iş uzayı eşleme yöntemi, ölçekleme kontrolü, sürüklenme kontrolü ve kenar hareket kontrolünü içinde barındırmaktadır. Bunlar aşağıda tek tek açıklanmıştır.

Ölçekleme kontrolü: Buradaki ölçekleme kontrolü diğer çalışmalardan biraz farklıdır.

$$X_a = k_s X_h + X_w \quad (1.2)$$

X_a genel koordinat sisteminde haptik imlecin konumunu ifade eder. X_h haptik cihazın kendi koordinat sistemindeki konum vektörüdür. X_w ise haptik koordinat sisteminin genel koordinat sistemine göre konumunu gösterir.

$$k_s = k_r t + k_{s0} \quad (1.3)$$

k_{s0} katsayısı k_s nin başlangıç değeridir ve t de istenilen imleç hareketi için gereken zaman aralığıdır. Haptik işuzayını genişletirken k_r değeri sıfırdan büyük olmak zorundadır. Bu yüzden k_r balistik fonksiyon olarak tanımlanmıştır ve değeri haptik imleç hızı ile orantılıdır. Diğer bir taraftan imleç hızı düşerken k_r değeri de sıfıra yakın veya daha küçük olmalıdır. Bu durumda k_r değeri sabit oranda tanımlanabilir. k_{sp} pozitif ölçekleme katsayısı, k_{sn} negatif ölçekleme katsayısı ve v_{ht} haptik imleç hızının eşik değeridir.

$$k_r = \begin{cases} k_{sp} \|\dot{X}_h\|, & \text{if } \|\dot{X}_h\| > v_{ht} \\ k_{sn} & , \text{if } \|\dot{X}_h\| \leq v_{ht} \end{cases} \quad (1.4)$$

Sürüklenme kontrolünde ise haptik uzayın orijini de kaymaktadır. Kayma oranı katsayısı aşağıdaki formüle göre hesaplanır.

$$X_w = \begin{cases} k_d \left(\dot{X}_h + \left(\frac{X_h}{\|X_h\|} \right) \right) t + X_{w0}, & \text{if } \|X_h\| > 0 \\ k_d \dot{X}_h t + X_{w0} & , \text{if } \|X_h\| = 0 \end{cases} \quad (1.5)$$

$$k_d = \begin{cases} k_{dp} & , \text{if } \|\dot{X}_h\| > v_{ht} \\ -k_{sn} & , \text{if } \|\dot{X}_h\| \leq v_{ht} \end{cases} \quad (1.6)$$

Sınır hareket kontrolünde, fiziksel haptik cihaz uzayının sınırlarına gelindiğinde haptik imleç hızı fiziksel iş uzayı limitinden dolayı sıfıra iner ancak bu haptik iş uzayındaki haptik imleç hızını etkilememelidir ($k_s X_h$). Bununla beraber sürüklenme hızı, robot kol uç noktasının hareket hızından daha yüksek olmalıdır ki sürüklenme kuvveti daha az hissedilsin.

[31]' de yapılan çalışmada konum ve hız kontrolün güçlü yanları biraraya getirilerek melez bir kontrol yöntemi önerilmiştir. Ancak bu bir sorunu da beraberinde getirmektedir. İki kontrol yöntemi birarada kullanıldığı için konum ve hız kontrolü için kullanılan işzayı boyutları küçülmüştür. Bu yüzden tasarım parametreleri de dikkatli seçilmesi gerekmektedir.

1.1 Tezin Amacı ve Yenilikçiliği

Bu çalışmada kuvvet geri beslemeli robotik teleoperasyonun kullanılabilirliği incelenmiştir. Tezin amacı, uzaklık sensörleri kullanılarak yapılan bir haptik teleoperasyon için en uygun kontrol yöntemini bulmaktır. Diğer çalışmalarda kuvvet geri beslemesi için genel olarak uzak robot tarafında kuvvet-tork sensörleri kullanılmasına rağmen bu çalışmada uçuş zamanlı uzaklık sensörleri kullanılmıştır. Uçuş zamanlı uzaklık algılayıcılar ile etraftaki cisimlere olan uzaklık ölçülür ve bu uzaklık azaldıkça kullanıcıya daha büyük bir kuvvet uygulanır. Bu sayede kullanıcının robotu kontrol ederken robotun etrafındaki nesnelere ile çarpışma riski azaltılmış olur. Ayrıca diğer çalışmalardan alınan farklı kontrol yöntemleri bu teleoperasyon sistemi üzerinde uygulanmış olup yeni bir kontrol yöntemi de geliştirilmiştir. Bu geliştirilen kontrol yöntemi de diğer yöntemler ile beraber incelenmiştir.

1.2 Tezin Kapsamı

Tez genel olarak 5 anabashiya ayrılmaktadır. İlk olarak giriş kısmında literatür taraması yapılmış ve tezin amacı ve yenilikçiliğinden bahsedilmiştir. Daha sonra ikinci bölümde sistem yapısından bahsedilmiş ve teleoperasyon sisteminde kullanılan kontrol yöntemleri detaylı bir şekilde açıklanmıştır. Üçüncü bölümde test düzeneği üzerindeki bileşenlerden bahsedilmiştir. Ayrıca yörünge nasıl tasarlandığı ve sistemin nasıl çalıştırılacağı da anlatılmıştır. Dördüncü kısımda deneylerin nasıl yapıldığı ve datanın nasıl toplandığı anlatılmıştır. Beşinci bölümde ise sonuçlar ortaya konulmuştur.

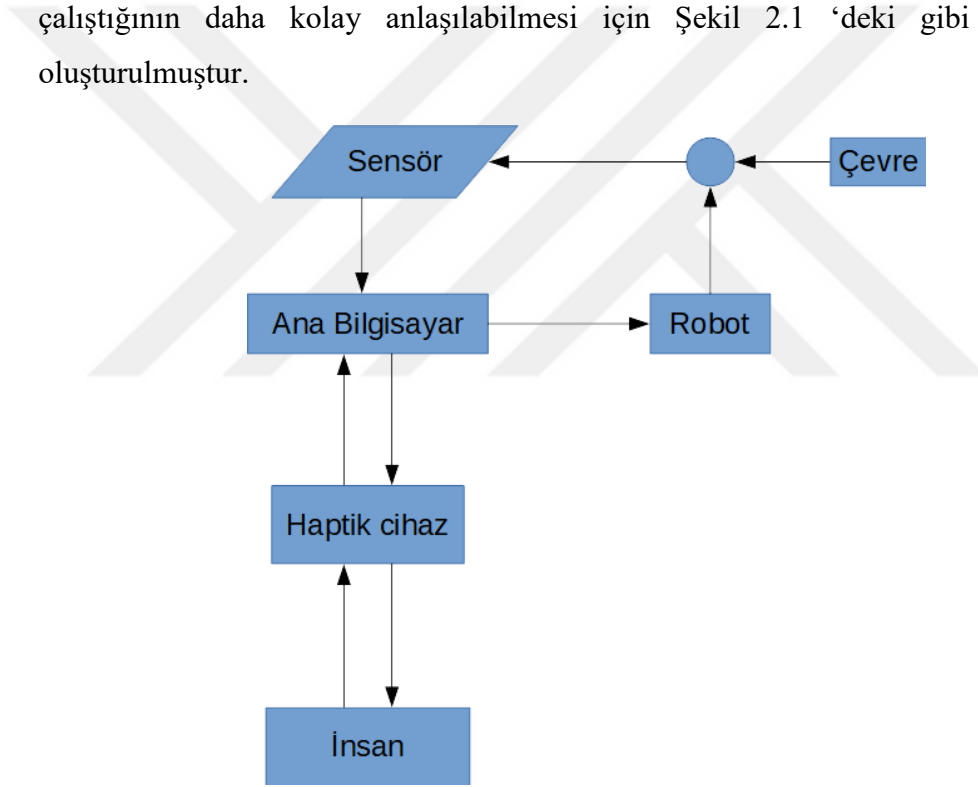


2. SİSTEM YAPISI VE BİLEŞENLERİ

Bu bölümde sistem yapısından ve kontrol yöntemlerinden bahsedilmiştir. Bölüm 2.1’de sistem modeli açıklanmakta olup Bölüm 2.2 ‘de de kontrol yöntemleri detaylı bir şekilde açıklanmıştır.

2.1 Model

Bu kısımda sistemin genel bir modeli verilmiştir. Teleoperasyon sisteminin nasıl çalıştığının daha kolay anlaşılabilmesi için Şekil 2.1 ‘deki gibi bir model oluşturulmuştur.



Şekil 2.1 : Sistem modeli.

Şekil 2.1 ‘de teleoperasyon sisteminin bir modeli görülmektedir. Bir insan haptik cihaz ile uzaktan bir robot kolu hareket ettirmektedir. Bunun için de veri iletişimi ana bilgisayar üzerinden sağlanmaktadır. Robot kol üzerindeki algılayıcılar sayesinde de etrafındaki nesnelere olan uzaklıklar algılanıp ana bilgisayara gönderilmektedir. Bu uzaklıklar kullanıcıya anlık olarak kuvvet geri beslemesi sağlanması için bir değişkendir. Çevredeki nesnelere yaklaşıldıkça kullanıcıya daha büyük bir kuvvet geribeslemesi verilir.

2.2 İş Uzayı Eşleme ve Kontrol Yöntemleri

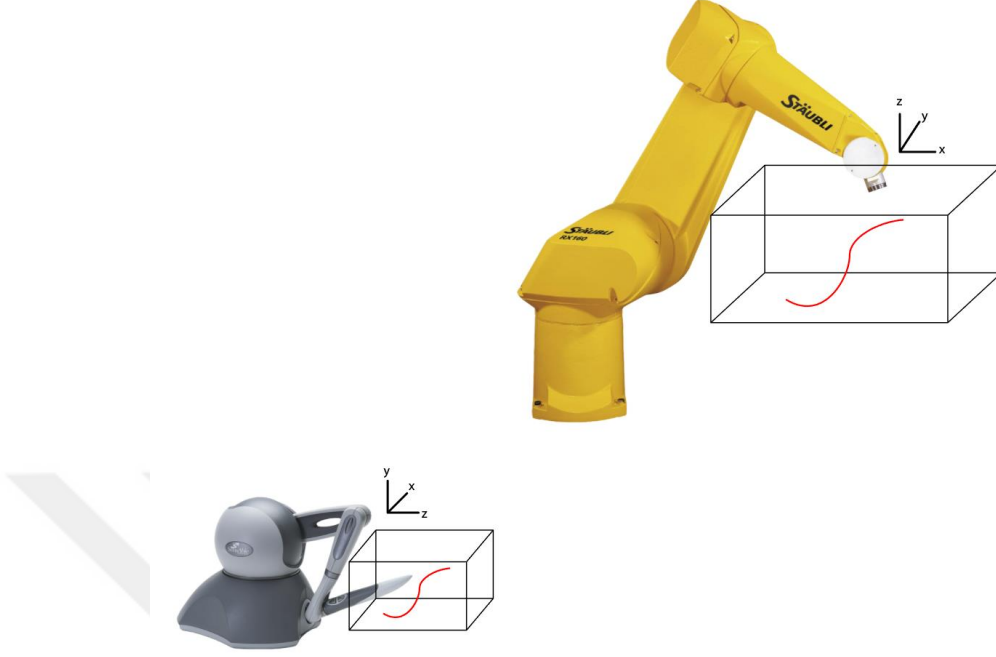
Teleoperasyon sistemlerinde temel olarak üç eşleme yöntemi bulunmaktadır. Bunlar eklemekten eklemeye eşleme, poz eşleme ve noktadan noktaya eşlemedir. Eklemekten eklemeye eşleme, birbiri ile bağlantılı iki robotun eklem açılarını ilişkilendirir. Bu eşleme yönteminde merkez robotun her eklem açısı, uydu robotun eklem açılarıyla birebir eşlenir. İki robot arasında kinematik ve yapısal farklılıklar olduğu zaman eklemekten eklemeye eşleme sınırlıdır. Poz eşlemede poz eşleme beklenmeyen hareketler oluşmasına sebep olabilir. Bu gibi durumlarda en iyi eşleme yöntemi noktadan noktadır. Bu yöntem yapıdan bağımsız olarak iki robotun uç noktalarını birbiriyle eşler. Bu çalışmada noktadan noktaya eşleme yapılmış ve aşağıdaki kontrol yöntemleri uygulanmıştır.

2.2.1 Konum kontrolü

Bu kontrol yönteminde merkez robot ve uzaktan kontrol edilen robot yaklaşık olarak aynı hareketleri yaparlar. Merkez robot bir yönde çekildiği zaman uzaktaki robot da aynı yönde harekete geçer, durduğu zaman o da durur. Bu yöntem anlaşılması ve kullanımı kolay bir kontrol yöntemidir.

Konum kontrolü Şekil 2.2’de de görüldüğü gibi haptik cihaz hareketinin robot kola doğrudan aktarımı ile gerçekleşmektedir. Haptik cihaz tarafında yapılan bir hareket robot kol tarafına anlık olarak aktarılır ve robot kol da aynı hareketi yapar. Konum kontrolünde uzay eşlemesi Denklem (2.1)’deki gibi yapılmaktadır. Denklemdeki $\bar{P}_t^{(r)}$ uzaktaki robotun uç noktasının robot eksen takımına göre konumunu, $\bar{P}_o^{(r)}$ robot uç noktasının başlangıç konumunu (robot eksen takımına göre), \hat{R} dönme matrisini, \hat{k}_g el katsayı matrisini, k_p konum kontrolü eşleme katsayısını, $\bar{P}_t^{(h)}$ haptik cihaz uç noktasının haptik eksen takımına göre konumunu ifade etmektedir. Eğer k_p değeri daha küçük seçilirse daha hassas kontrol yapılabilir, fakat bu şekilde etkin kullanılan iş uzayı boyutu küçülmüş olur. Katsayı daha büyük seçilirse de daha kaba bir eşleme yapılmış olur. \hat{k}_g katsayı matrisi merkez robot ile uydu robot arasındaki iş uzayı ilişkisini belirtmektedir. Bu matrisin katsayıları değiştirilerek kullanılmak istenen iş uzayı boyutları belirlenebilir. Burada her doğrultuda aynı katsayı kullanılmak zorunda değildir. Farklı doğrultular için farklı katsayılar da belirlenebilir. Örneğin k_1 ,

k_2 ve k_3 birbirinden farklı seçilirse merkez robot tarafında küp şeklindeki bir işuzayı, uzak robot tarafında dikdörtgenler prizması şeklini alabilir.



Şekil 2.2 : Konum kontrolü ile haptik cihaz ve robot kol arasındaki iş uzayı eşlemesi.

$$\hat{R} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\hat{k}_g = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix}$$

$$\bar{P}_t^{(h)} = \begin{bmatrix} x_t^{(h)} \\ y_t^{(h)} \\ z_t^{(h)} \end{bmatrix}$$

$$\bar{P}_t^{(r)} = \begin{bmatrix} x_t^{(r)} \\ y_t^{(r)} \\ z_t^{(r)} \end{bmatrix}$$

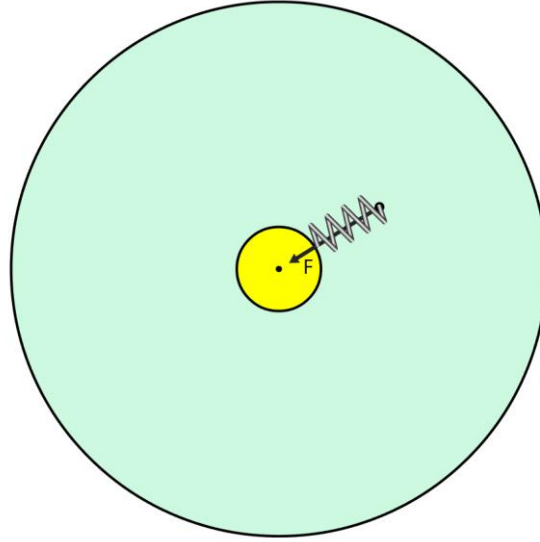
$$\bar{P}_o^{(r)} = \begin{bmatrix} x_0^{(r)} \\ y_0^{(r)} \\ z_0^{(r)} \end{bmatrix}$$

$$\bar{P}_t^{(r)} = \bar{P}_o^{(r)} + \hat{R} \hat{k}_g k_p \bar{P}_t^{(h)} \quad (2.1)$$

2.2.2 Hız kontrolü

Hız kontrolünde merkez robot ile uydu arasında doğrudan bir konum eşleme söz konusu değildir. Onun yerine merkez robotun kendi eksen takımındaki konumu, uydu robota hız komutu olarak verilir.

Hız kontrolündeki kontrol bölgeleri Şekil 2.3'te görülmektedir. Bu kontrol bölgeleri merkez robotun orijini etrafındaki hayali bölgeler olmakla birlikte dışta ve içte olmak üzere merkezleri çakışık iki küreden oluşmaktadır. İki küre arasındaki bölgede konum vektörü uzaktaki robota hız vektörüne dönüştürülerek gönderilir. Yani merkezden uzaklaştıkça daha hızlı, merkeze yaklaştıkça da daha yavaş bir hareket uzak robot tarafında sağlanabilir. Fakat bu şekilde yapılan bir kontrolde robotun sürekli olarak hareket etmesi kontrolü zorlaştırabileceğinden dolayı orijin etrafına küçük bir ölü bölge eklenmiştir. Bu bölgenin içinde iken uzaktaki robota hız komutu gönderilmez. Merkez robotun uç noktası ölü bölgenin içerisine alınarak uzaktaki robotun herhangi bir noktada sabit tutulması sağlanabilir. Ayrıca merkezden uzaklaştıkça hayali bir yay kuvveti haptik cihaz tarafından kullanıcıya uygulanır. Bu sayede robot kolun hızı kuvvet geri beslemesi ile takip edilebilir.



Şekil 2.3 : Hız kontrolündeki kontrol bölgeleri.

Merkez robotun iş uzayında orijin etrafında r_s yarıçaplı hayali bir ölü bölge bulunmaktadır. Merkez robot uç noktası bu bölgenin içerisinde iken, uydu robot hareketsizdir. Ölü bölgenin dışına çıkıldığında ise bu küreden uzaklıkla orantılı olarak bir hız komutu uzaktaki robota gönderilir. Denklem (2.2) 'de bu ilişki matematiksel olarak gösterilmiştir. Denklemdeki $\Delta \bar{P}_t^{(r)}$ terimi uydu robot uç noktasının hızını ifade

etmektedir. k_r katsayısı ile merkez robotun belli bir konumuna karşılık gelen uydu robot hızı belirlenebilir. Örneğin bu değer daha büyük seçilirse merkez robot aynı konumda olmasına rağmen uzaktaki robot daha hızlı hareket edecektir. r_s de merkez robot uzayındaki hayali kürenin yarıçapıdır. Buna göre merkezdeki kürenin dışına çıkıp uzaklaştıkça hız artmaktadır. Belli bir konumu hassas olarak bulma konusunda iyi bir yöntem değildir, fakat sabit bir hızda belli bir yönde gitmek için en iyi yöntemlerden biridir.

$$\Delta \bar{P}_t^{(r)} = \begin{cases} \hat{R} \hat{k}_g \bar{P}_t^{(h)} \left(\frac{|\bar{P}_t^{(h)}| - r_s}{|\bar{P}_t^{(h)}|} \right) k_r & \left| \bar{P}_t^{(h)} \right| > r_s \\ 0 & \left| \bar{P}_t^{(h)} \right| \leq r_s \end{cases} \quad (2.2)$$

Kuvvet geri beslemesi olarak da ortadaki ölü bölgeye çeken bir yay kuvveti ve duvarlara yaklaştıkça artan bir tepki kuvveti vardır. Bu iki kuvvetin bileşkesi kullanıcıya verilir. Aynı anda hem yan duvarlar hem de tabandan gelen kuvvet merkez robotun kontrolünü zorlaştırdığından dolayı tabandan gelen kuvvet azaltılmıştır. d_d , içerisinde sensörlerden gelen uzaklık datasını bulundurur. d_1 ön, d_2 sağ, d_3 arka, d_4 sol ve d_5 alt taraftaki nesneye olan uzaklığı ifade eder. \bar{d}_s sıra matrisi sanal ortamda yayların sıkıştırılma yönünü ve miktarını ifade eder. d_{max} değeri kuvvetin uygulanmaya başlanacağı mesafeyi ifade etmektedir. Bu mesafeden daha çok duvara yaklaşırsa duvar tarafından hayali bir kuvvet uygulanmaya başlar. Duvarın dibinde en yüksek değeri alan bu kuvvet Denklem (2.3)' deki gibi hesaplanır.

$$d_d = [d_1 \quad d_2 \quad d_3 \quad d_4 \quad d_5]$$

$$\bar{d}_s = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

$$d_x = \begin{cases} d_{max} - d_2, & d_2 \leq d_{max} \\ d_4 - d_{max}, & d_4 \leq d_{max} \end{cases}$$

$$d_y = \{(d_{max} - d_5)/10, \quad d_5 \leq d_{max}\}$$

$$d_z = \begin{cases} d_1 - d_{max}, & d_1 \leq d_{max} \\ d_{max} - d_3, & d_3 \leq d_{max} \end{cases}$$

$$\hat{k}_y = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & k \end{bmatrix}$$

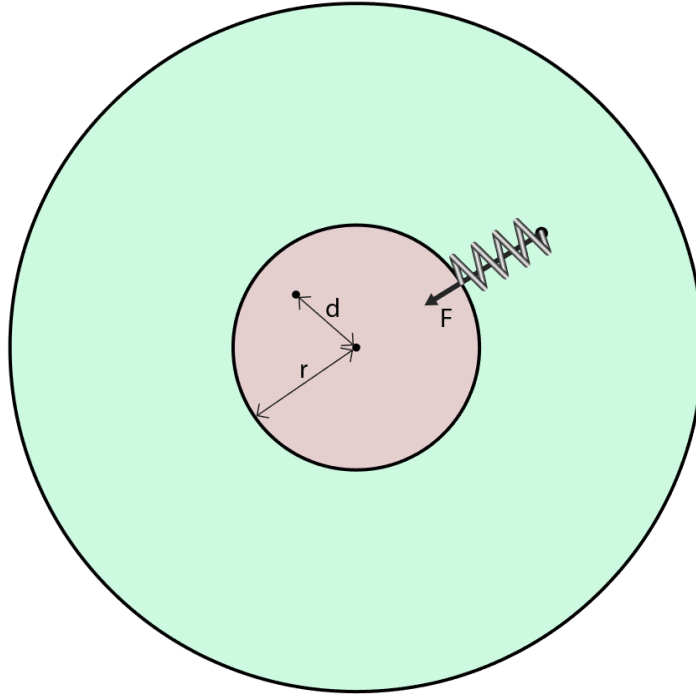
$$F = \hat{k}_y \bar{d}_s \quad (2.3)$$

2.2.3 Geçişli konum-hız kontrolü

Bu kontrol yönteminde konum ve hız kontrolü yöntemlerinin ikisinden de faydalanılmıştır. Merkez etrafında bir küre belirlenmiş, kürenin iç kısmında konum kontrolü, dışında ise hız kontrolü kullanılmıştır. Bu sayede iki yöntemin avantajlarından da faydalanılmak istenmiştir. Fakat kontrol geçişleri sırasında hız ve konumdaki atlamalar sebebiyle kontrol zorlaşmaktadır. Denklem (2.4) 'te bu yöntem matematiksel olarak ifade edilmiştir. Burada r_s merkez robot uzayındaki hayali kürenin yarıçapıdır. Merkez robotun orijinine konumlandırılmış $2 * r_s$ yarıçaplı bir küre etrafında hız kontrolü, içinde de konum kontrolü yapılır ve yöntemler arası geçişler otomatik olarak gerçekleşmektedir.

$$\Delta \bar{P}_t^{(r)} = \begin{cases} \hat{R} \hat{k}_g \bar{P}_t^{(h)} \left(\frac{|\bar{P}_t^{(h)}| - r_s}{|\bar{P}_t^{(h)}|} \right) k_r & \left| \bar{P}_t^{(h)} \right| \geq 2 * r_s \\ \hat{R} \hat{k}_g \Delta \bar{P}_t^{(h)} & \left| \bar{P}_t^{(h)} \right| < 2 * r_s \end{cases} \quad (2.4)$$

Geçişli konum-hız kontrolündeki kontrol bölgeleri Şekil 2.4'te görülmektedir. Merkezleri çakışık iç içe geçmiş iki adet küresel bölge vardır. İçteki bölgede konum kontrolü kullanılırken dış bölgede hız kontrolü kullanılır. Ayrıca hız kontrolü kullanılırken merkezden uzaklığa bağlı olarak da kullanıcıya kuvvet geri beslemesi verilmektedir. Bu da hayali bir yay kuvveti ile sağlanmaktadır.



Şekil 2.4 : Geçişli konum-hız kontrolündeki kontrol bölgeleri.

2.2.4 Balistik kontrol

Balistik kontrol, konum kontrolünü andırmaktadır, fakat katsayı çarpanı sabit değil, onun yerine hız ile orantılı olarak artıp azalmaktadır. Hızlı kullanımlarda daha büyük bir bölgede hareket edilirken, yavaş kullanımda daha hassas bir kontrol elde edilir. Bu sayede istendiğinde hassas, istendiğinde kaba hareketler aynı hareket içinde yapılabilmektedir. Denklem (2.5)' teki $\bar{P}_n^{(h)}$ merkez robotun anlık konumunu $\bar{P}_{n-1}^{(h)}$ de bir önceki andaki (1 ms öncesi) konumunu vermektedir. k_b balistik kontrole özel katsayıdır. Bu katsayı artırılıp azaltılarak uzaktaki robotun hızı istenen bir değere ayarlanabilir. Daha yumuşak hareketler elde edebilmek için zaman aralığı olabildiğince küçük tutulup integral operatörüne benzer şekilde toplama işlemi yapılmıştır. Her 60ms aralıklarla uzaktaki robot kol konumuna bu fark eklenmektedir.

$$\Delta\bar{P}_t^{(r)} = \hat{R}\hat{k}_g \left(\sum_{n=1}^{60} k_b \left| \bar{P}_n^{(h)} - \bar{P}_{n-1}^{(h)} \right| \left(\bar{P}_n^{(h)} - \bar{P}_{n-1}^{(h)} \right) \right) \quad (2.5)$$

2.2.5 Değişken ölçekli konum kontrolü

Değişken ölçekli konum kontrolü normal konum kontrolün iki farklı eşleme katsayılı olan halidir. Merkez robot belli bir hızın altında iken k_1 katsayısı, üstüneyken de k_2 katsayısı kullanılır. Bu yavaş kullanımlarda daha hassas kontrol elde etmek, hızlı kullanımlarda da daha büyük bir işuzayı kullanarak kaba bir kontrol elde etmek içindir. Denklem (2.6)' daki $\bar{P}_g^{(h)}$ katsayı değişimlerinde robot kolun konumundaki sıçramaları önlemek için eklenmiştir. Her katsayı değişiminde bu değer de değişmektedir. Denklem (2.7)' de iki değişik katsayı arasında nasıl geçiş yapıldığı görülmektedir. $\bar{P}_n^{(h)}$ şu andaki merkez robotun konumu $\bar{P}_{n-1}^{(h)}$ de bir önceki andaki merkez robot konumudur (1 ms öncesi). Bu fark c sabitinden küçükse k_1 katsayısı, büyükse k_2 katsayısı geçerlidir.

$$\bar{P}_t^{(r)} = \bar{P}_o^{(r)} + \hat{R}\hat{k}_g(\bar{P}_g^{(h)} + \bar{P}_t^{(h)}k_v) \quad (2.6)$$

$$k_v = \begin{cases} k_1 & \left| \bar{P}_n^{(h)} - \bar{P}_{n-1}^{(h)} \right| \leq c \\ k_2 & \left| \bar{P}_n^{(h)} - \bar{P}_{n-1}^{(h)} \right| > c \end{cases} \quad (2.7)$$

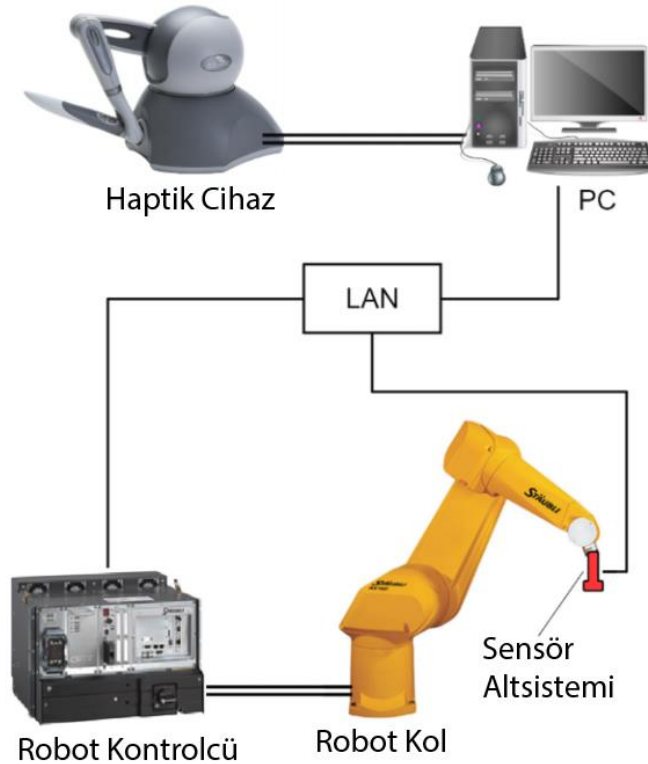


3. TEST DÜZENEĞİ TASARIMI

Bu bölümde test düzeneğinin tasarımında nelere dikkat edildiğinden, sistemdeki bileşenlerden bunların boyutlarından ve hassasiyetlerinden, izlenecek rotanın tasarımından ve sistemin nasıl çalıştırıldığından bahsedilmiştir.

3.1 Test Düzeneği Bileşenleri

Şekil 3.1’de teleoperasyon sistemindeki elemanlar görülmektedir. Sistem üzerinde Phantom Omni haptik cihaz, Stäubli RX160 robot kolu, ana bilgisayar, robot kontrolcü, Raspberry Pi 3 ve uzaklık algılama için kullanılan VL6180X uçuş zamanlı uzaklık sensörleri bulunmaktadır. Bu bölüm altında Phantom Omni haptik cihazın ve Stäubli RX160 robot kolun özelliklerinden ve kinematığından, sensör altsistemindeki VL6180X in çalışmasından bahsedilecektir.



Şekil 3.1 : Deney düzeneği.

3.1.1 Phantom Omni haptik cihaz

Phantom Omni haptik cihaz 6 serbestlik derecesine sahiptir. Seri olarak 6 ekleme sahip olan bu cihazın ilk 3 eklemi üzerindeki motorlar sayesinde eyletilebilir ve bu sayede kullanıcıya kuvvet geri beslemesi verilebilir. Son 3 bilek eklemi ise serbest olarak hareket ettirilebilir. Normal çözünürlüğü 0.055mm olan cihazın toplam ağırlığı da 1.79kg 'dır. Nominal kuvvet sınırları olarak 3.3N yazılımsal olarak belirlenmiştir. Yer koordinat sisteminin x, y ve z eksenlerinde anlık konum ve hız ölçümü için optik enkoderler bulunmaktadır. Haptik kalem ucunda 45g kütlesi bulunan haptik cihazın haberleşmesi RJ45 ethernet soketinden USB' ye çevirici kullanılarak yapılmakta ve gerçek zamanlı programlaması C++ programlama dili üzerinden OpenHaptics Toolkit ile yapılabilmektedir.

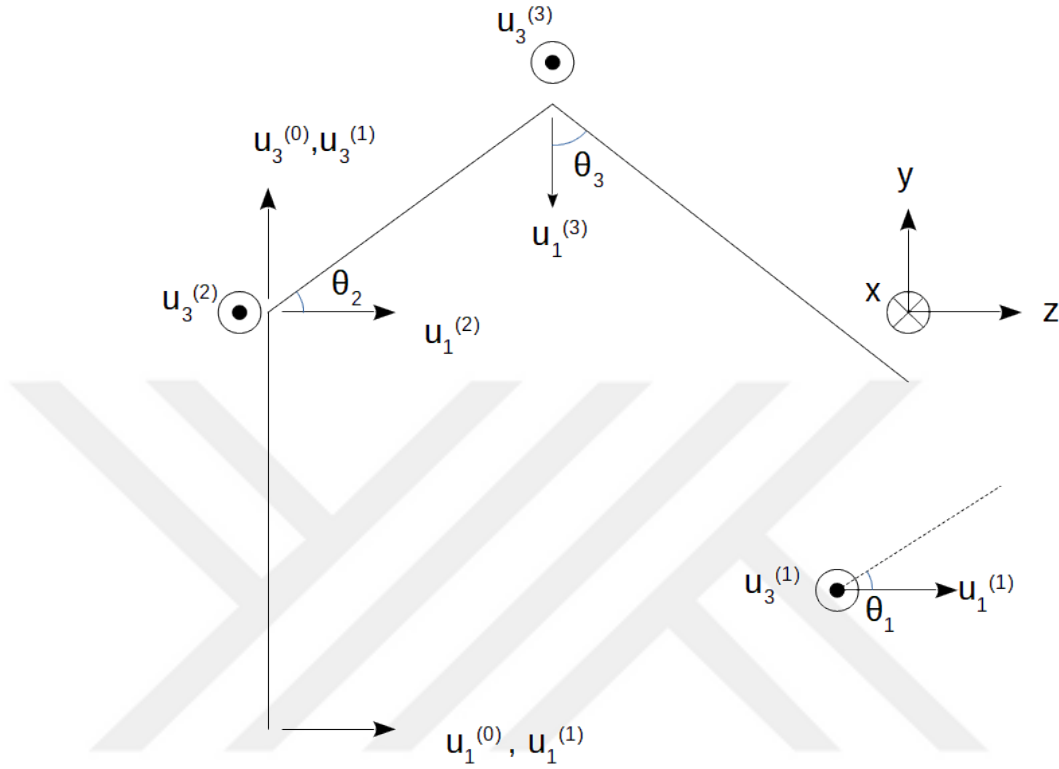
Şekil 3.2' de Phantom Omni haptik cihaz görülmektedir. Cihazın 6 serbestlik derecesi vardır. 6 adet eklemin her birinde enkoder bulunmasına rağmen sadece ilk üç eklem motorlar vasıtası ile kontrol edilebilmektedir. Motorlarla kontrol edilebilen eklemler şekil üzerinde gösterilmiştir. Sadece ilk üç eklem kontrol edilebildiği için ileri kinematik hesaplamalar ilk üç eklem için yapılmıştır.



Şekil 3.2 : Phantom Omni haptik cihaz.

Şekil 3.3' te Phantom Omni haptik cihazın ilk üç eklemi için eklem açıları ve uzunlukları gösterilmiştir. Burada d_1 ve d_2 sırasıyla birinci ve ikinci uzuv uzunluğudur. Hesaplanması istenen ikinci uzvun uç noktası ikinci ekleminin dönme noktasına göre bir miktar kaymış durumdadır. Bu kayma z ekseninde 0.170m, y

ekseninde ise -0.025m dir. Bu değerler düz kinematik hesaplama yapılırken orijin noktasını kaydırmak için kullanılmıştır. Haptik cihazın kontrol edilebilen üçüncü eklem uç noktasının kendi uzayındaki konumu aşağıdaki gibi hesaplanabilir [32].



Şekil 3.3 : Phantom Omni haptik cihazın eklem açıları ve uzunlukları.

Denklem (3.1), (3.2) ve (3.3) 'te haptik cihazın ileri kinematik hesaplamaları görülmektedir.

$$x = \sin\theta_1(d_1\cos\theta_2 + d_2\sin\theta_3) \quad (3.1)$$

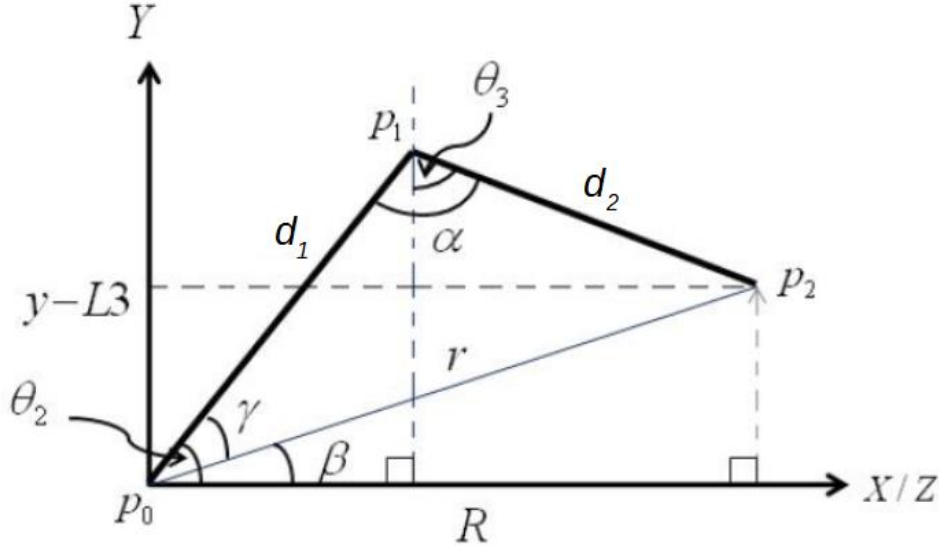
$$y = L_3 - d_2\cos\theta_3 + d_1\sin\theta_2 \quad (3.2)$$

$$z = -L_4 + \cos\theta_1(d_1\cos\theta_2 + d_2\sin\theta_3) \quad (3.3)$$

Haptik cihazın ters kinematik hesaplamaları da yapılmıştır. Yani x , y ve z koordinatları bilinen uç eklem noktaya bakılarak eklem açıları hesaplanabilir. Çözümüne önce θ_1 açısından başlanırsa bu açı Denklem (3.4) 'teki gibi bulunabilir.

$$\theta_1 = \text{atan2}(x, z + L_4) \quad (3.4)$$

Şekil 3.4'te Phantom Omni'nin ters kinematiği için hesaplanması gereken uzunluklar ve açılar verilmiştir. Bunlardan R , r ve β sırasıyla Denklem (3.5), (3.6) ve (3.7)'deki gibi ifade edilebilir.



Şekil 3.4 : Phantom Omni ters kinematiği.

$$R = \sqrt{x^2 + (z + L_4)^2} \quad (3.5)$$

$$r = \sqrt{x^2 + (z + L_4)^2 + (y - L_3)^2} \quad (3.6)$$

$$\beta = \text{atan2}(y - L_3, R) \quad (3.7)$$

$\Delta P_0 P_1 P_2$ üçgeninde kosinüs teoremi uygulanırsa aşağıdaki denklemler elde edilir.

$$d_2^2 = d_1^2 + r^2 - 2d_1 r \cos \gamma \quad (3.8)$$

$$\gamma = \cos^{-1} \left(\frac{d_1^2 + r^2 - d_2^2}{2d_1 r} \right) \quad (3.9)$$

Haptik cihazın sınırlandırmaları sebebiyle $\gamma > 0$ olmak zorundadır. Buna göre θ_2 aşağıdaki gibi elde edilir.

$$\theta_2 = \gamma + \beta \quad (3.10)$$

$\Delta P_0 P_1 P_2$ üçgeninde tekrar kosinüs teoremi uygulanıp bu sefer α açısı hesaplanır.

$$r^2 = d_1^2 + d_2^2 - 2d_1 d_2 \cos \alpha \quad (3.11)$$

$$\alpha = \cos^{-1} \left(\frac{d_1^2 + d_2^2 - r^2}{2d_1 d_2} \right) \quad (3.12)$$

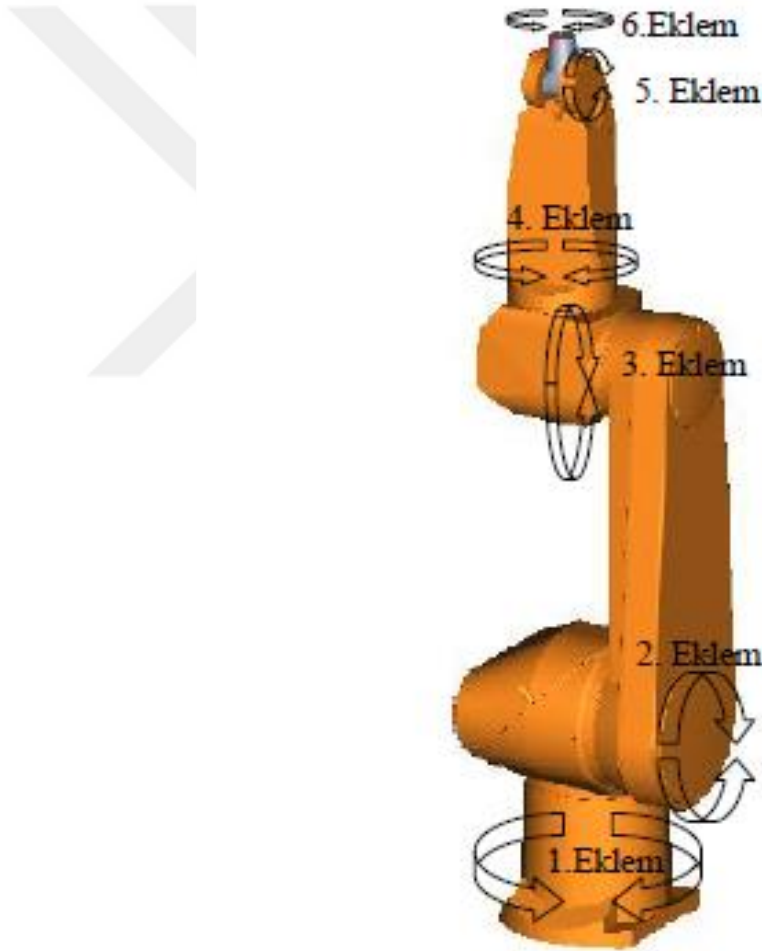
$$\theta_3 = \theta_2 - \frac{\pi}{2} + \alpha \quad (3.13)$$

Bu şekilde tüm açılar hesaplanmış olur.

3.1.2 Stäubli RX160 robot kolu

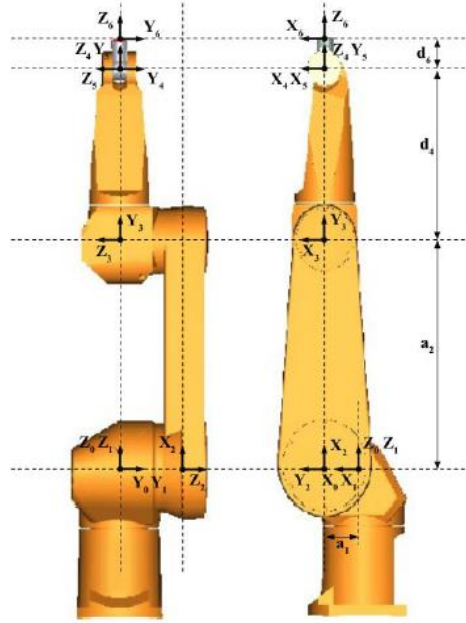
Stäubli RX160 robot kolu 6 serbestlik derecesine sahiptir ve 20kg bir nominal yük taşıma kapasitesi vardır. Yüksek hassasiyette konum kontrolü yapılabilen bu robot kol sayesinde bir çok hassas iş programlanarak otomatik olarak yapılabilmektedir. Ayrıca bu konum hassasiyeti robot kolun arařtırmalarda ve testlerde de kullanılmasına olanak tanımaktadır.

Şekil 3.5'te Stäubli RX160 robot kolun eklemleri görülmektedir. Eklemlerin hepsi de dönel eklemlerdir. Robot yere veya tavana sabitlenebilmektedir. Bu çalışmada yere sabitlenmiş olarak kullanılmıştır.



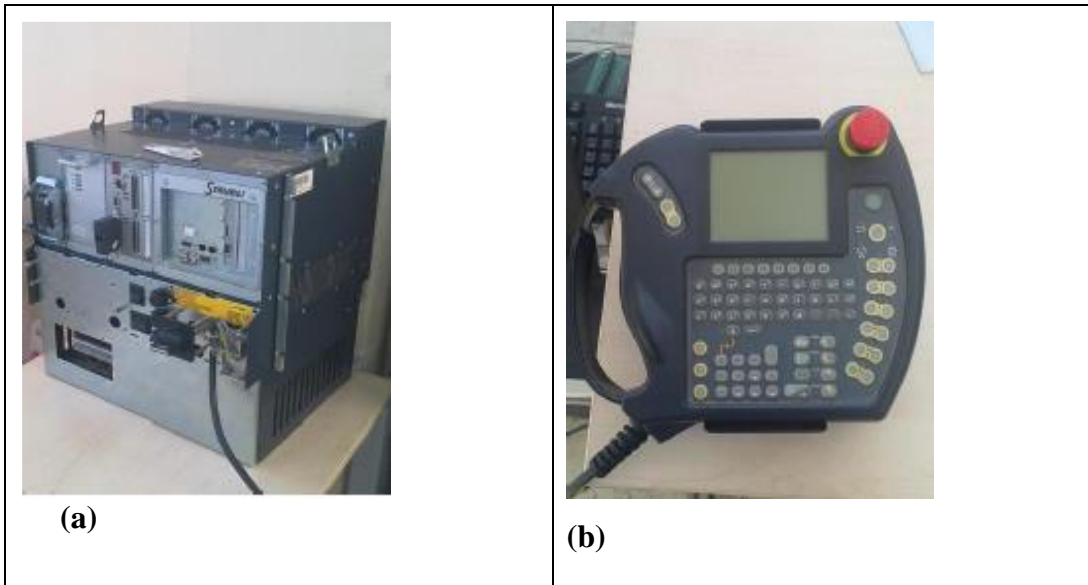
Şekil 3.5 : Stäubli RX160 robot kolun eklemleri.

Şekil 3.6' da robot kolun DH parametrelerinin hesaplanması için çizilen bağ çerçeveleri görülmektedir. Eksenlerin bu kısımda düzgün seçilmesi, daha sonra DH parametrelerinin bulunması için önemlidir.



Şekil 3.6 : Stäubli RX160 robot kolun bağ çerçevesi.

Şekil 3.7' de robot kolu kontrol eden CS8C kontrolcü (solda) ve el kumandası (sağda) görülmektedir. Robot kolun düz ve ters kinematik hesapları ile beraber bütün kontrolü CS8C robot kontrolcüsü üzerinden yürütülür. Robota giden güç yalnızca el kumandası ile aktif edilecek şekilde tasarlanmıştır. Bu sayede kullanıcının isteği dışında robota güç aktarımı engellenerek ek bir güvenlik katmanı sağlanır. Kumanda vasıtasıyla robota giden gücü aktif hale getirmek için robot acil durdurma modundan çıkarıldıktan sonra kumanda üzerinden yeşil butona basılır.



Şekil 3.7 : CS8C Robot kontrolcü (a) ve el kumandası (b).

Çizelge 3.1’de Stäubli RX160 robot kolun DH parametreleri görülmektedir. Bu parametreler daha önce [33]’ te aynı robot üzerinde yapılan başka bir çalışmada hesaplanmıştır.

Çizelge 3.1 : Stäubli RX-160 robot kolun DH parametreleri.

i	γ_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90	$a1 = 150 \text{ mm}$	0	θ_2
3	0	$a2 = 825 \text{ mm}$	0	θ_3
4	-90	0	$d4 = 625 \text{ mm}$	θ_4
5	90	0	0	θ_5
6	-90	0	$d6 = 110 \text{ mm}$	θ_6

3.1.3 Sensör altsistemi

Sensör altsistemi robot kol uç noktasının etrafındaki nesnelere uzaklığını belirlemek için kullanılmaktadır. Burada uçuş zamanlı uzaklık sensörü sayesinde hassasiyeti mm düzeyinde olacak şekilde uzaklık ölçümü yapılabilmektedir. Raspberry Pi 3 de bu sistem içerisindedir ve üzerindeki Linux işletim sistemi sayesinde kolaylıkla sensör verileri anlık olarak izlenebilmekte ve yerel ağ bağlantısı üzerinden ana bilgisayara gönderilebilmektedir. Sistem üzerinde ayrıca çoklayıcı (yolseçici) da bulunmaktadır. Bu sayede sensörlerden sıra ile data toplanabilmektedir. Sensör altsisteminin robot üzerine yerleştirilmiş hali Şekil 3.8’ de görülmektedir.

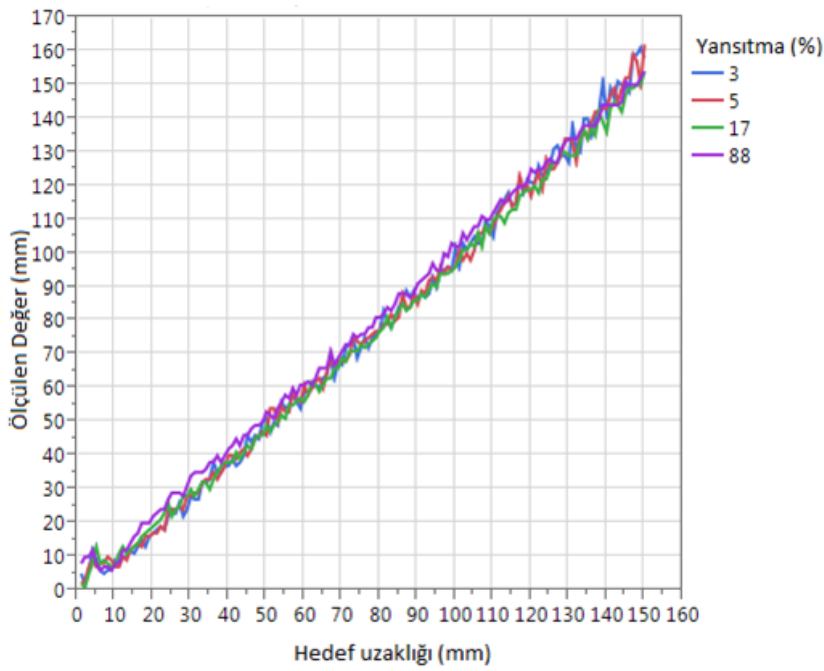


Şekil 3.8 : Raspberry Pi ve uzaklık algılayıcıların robot üzerine yerleştirilmiş hali.

Sistem üzerinde uzaklığı algılamak için VL6180X uzaklık ve ışık sensörü kullanılmıştır. Bu sensörler bir ışık fotonunun hedef nesneden yansıyor geri geldiği süreyi baz alarak o nesneye olan uzaklığı ölçer. Fotonun yolculuk süresi ile ışık hızı çarpıldığında nesne ile olan uzaklık belirlenir. Sensörden okunan değerler uzaklık

miktarlarının mm cinsinden değerleridir. Uzaklık ölçümleri 1mm hassasiyette ölçülebilir.

Şekil 3.9’ da VL6180X uçuş zamanlı uzaklık algılayıcıların yansıtma oranına bağlı tipik ölçüm performansı grafiği verilmiştir. Grafiğe bakıldığında en iyi ölçüm sonuçlarına yüksek yansıtma oranlarında ulaşılabildiği görülse de düşük yansıtma oranlarında da doğrusala çok yakın ölçüm sonuçları elde edilebildiği görülebilir. En iyi şekilde 0-100mm arasında performans gösteren bu sensörler normalde 200mm’ye kadar ölçüm yapabilirken katsayıları değiştirilip hassasiyetten ödün verilerek 400mm’ ye kadar uzaklık ölçümü yapılması sağlanabilir.

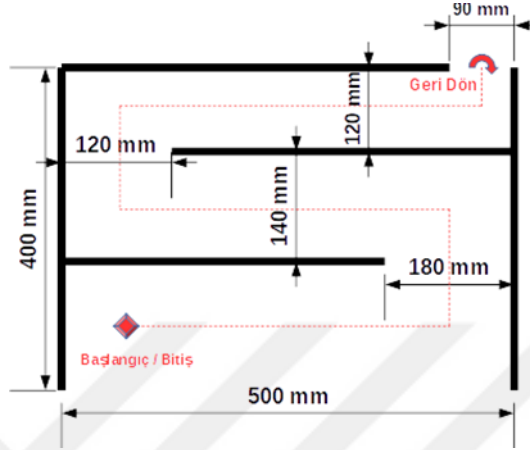


Şekil 3.9 : VL6180X uzaklık sensörünün tipik menzil performansı [34].

3.2 Yörünge Tasarımı

Bu bölümde robot kolun izlediği yörünge nasıl tasarlandığından bahsedilmiştir. Yörünge tasarlanırken öncelikle öngörülen kontrol yöntemlerinin kullanılabileceği ve sayısal data toplanabilecek şekilde bir yörünge tasarımı yapılması gerekmektedir. Bunun için çeşitli şekillerde yörüngeler belirlenebilir. Örneğin, sıvı boşaltma, çubuğa somun geçirme, üst üste nesne dizme gibi sadece iki noktada dikkat gerektiren yörüngeler belirlenebileceği gibi, bir düzlemde belli bir şekil çizme, labirent içinde belli bir rota takip etme şeklinde tüm işlem boyunca dikkat gerektiren yörüngeler de belirlenebilir.

Bu çalışmada Şekil 3.10' daki gibi bir yörünge belirlenmiştir. Yörünge birden fazla manevra ve dikkatli kontrol gerektiren zorlayıcı bir özelliğe sahip olduğundan dolayı bu şekilde belirlenmiştir. Bu sayede kontrol yöntemleri arasında süre ve hata bakımından daha belirgin farklar oluşacağından dolayı daha kolay karşılaştırılabileceklerdir.



Şekil 3.10 : Robot kol için belirlenen yörünge.

3.3 Sistemin Çalıştırılması

Sistemin teleoperasyon için uygun hale getirilmesi için öncelikle gerekli programların yüklenmesi, gerekli ayarların da yapılması gerekmektedir. Yapılması gereken ayarlar aşağıda gösterilmiştir:

1. Sistemde kullanılan bileşenler aşağıda listelenmiştir. Sistem bileşenlerinin eksiksiz olduğundan ve bağlantıların düzgün yapıldığından emin olunur. Bileşenler ve yapılan bağlantılar aşağıda gösterilmiştir.

Sistem bileşenleri;

CS8C Robot Kontrolcü (VxWorks gerçek zamanlı işletim sistemi)

Stäubli RX-160 Robot Kolu

Phantom Omni Haptik Cihaz

Bilgisayar (Windows 10 işletim sistemi (opsiyonel))

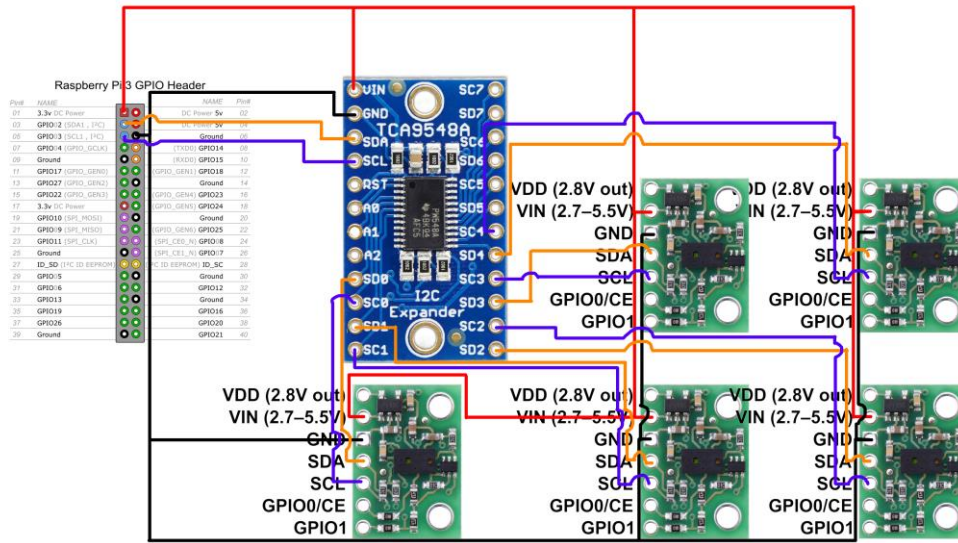
Raspberry Pi 3

Uçuş zamanlı uzaklık ve ışık sensörü (5 adet, her yön için birer tane)

Çoklayıcı (Adafruit TCA9548A)

Web kamerası

2. İlk maddede verilen sistem bileşenleri ve Şekil 3.11’de gösterilen bağlantılar hazırlandıktan sonra kontrolcü, bilgisayar ve Raspberry Pi üzerindeki yazılım versiyonları kontrol edilir. CS8C robot kontrolcü üzerinde standart olarak VxWorks işletim sistemi yüklüdür. Cihaz üzerinde ayrıca VAL3 versiyon s6.5.5 yüklüdür. Cihaz üzerindeki programlar VAL3 versiyonuna uygun şekilde yazılır.



Şekil 3.11 : Raspberry Pi, TCA9548A çoklayıcı ve sensörler arasındaki bağlantılar.

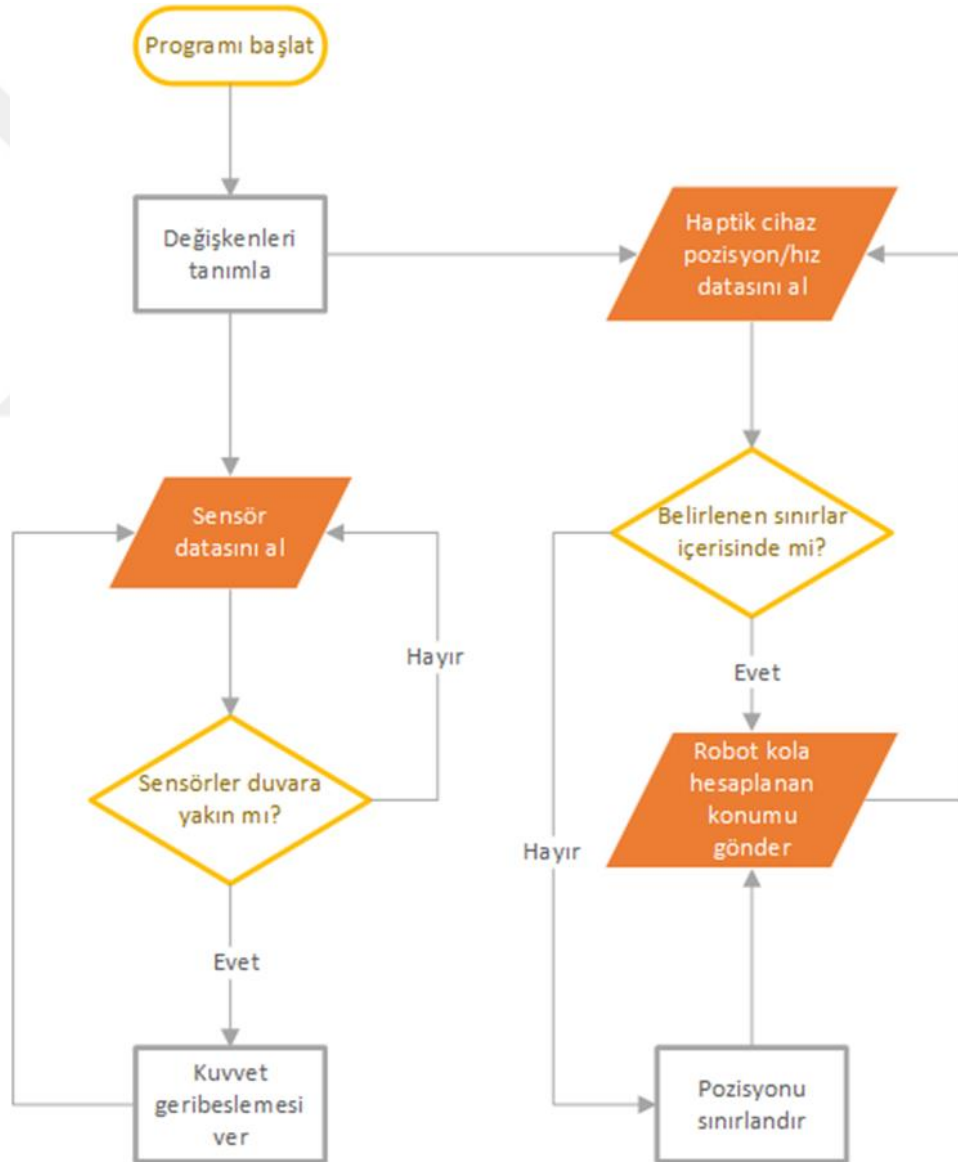
3. Bilgisayar üzerine Windows 10 işletim sistemi yüklenir. Ardından kontrol yöntemlerini çalıştırmak için yapılan programları, düzenlemek ve derlemek için Visual Studio 2015 Community Edition yüklenir. Daha sonra Phantom Omni haptik cihaz ile iletişim kurmak ve cihazı control etmek için Geomagic Touch Device Driver 2016.1.1 ve OpenHaptics Developer Edition v3.4.0 yüklenir.

4. CS8C robot kontrolcü ile bağlantı kurup üzerine program yüklemek veya kontrolcü üzerinden mevcut programları bilgisayara indirmek için Stäubli Robotic Suite 2013.4.7 programı yüklenir. Bu program robotun web sitesindeki Stäubli Technical Database kısmından kullanıcı adı ve şifre girişi yapılarak indirilebilir.

5. Raspberry Pi üzerine Raspbian işletim sistemi kurulur. Daha sonra Python programlama dili vasıtasıyla sensörlerden veri okuyup bilgisayara internet üzerinden gönderen bir program yazılır.

6. Bilgisayar ile Raspberry pi arasında ssh bağlantısı kurmak için bilgisayar üzerine PuTTY programı yüklenir.

Bu çalışmada, içerisinde çeşitli mekanik ve elektronik bileşenlerinin olduğu bir teleoperasyon sistemi ele alınmaktadır. Şekil 3.12' ye bakıldığında algoritmanın genel bir çalışma mantığı görülmektedir. Sistem basit olarak kullanıcıdan konum bilgisi alıp bu bilgi ile robotu hareket ettirmekte ve robotun uç noktasının serbest uzayda çevreye olan uzaklığına bağlı olarak da kullanıcıya kuvvet geri beslemesi vermektedir. Bütün bunlar yapılırken de robot kola kendi uzayında olmayan bir konum bilgisi gönderilmemesi için her döngüde haptik cihazdan alınıp robota gönderilen konum bilgisi de kontrol edilir.



Şekil 3.12 : Algoritmanın genel çalışması.

Robot kolun uç noktasına, etrafı algılamak için, ön,arka,sağ,sol ve alt tarafa bakacak şekilde uzaklık sensörleri yerleştirilmiştir. Bunlar labirent şeklindeki test düzeneğinde robot kol uç noktasının duvarlara olan uzaklığını anlık olarak takip etmekte görev alırlar. Sensörlerin kendine has fiziksel adresleri vardır. Sensörlerde adres çakışmasını önlemek için sensörler ile Raspberry Pi arasına çoklayıcı bağlanmıştır. Adres değişimleri multiplexer üzerinden sağlanmakta olup sistemi daha güvenli hale getirmektedir. Raspberry Pi üzerindeki Raspbian işletim sisteminde çalışacak şekilde, sensörlerden uzaklık verilerini okuması için python programlama dili kullanılarak ufak bir program yazılmıştır. Program basitçe, sensörlerden sırayla okuduğu veriyi internet üzerinden TCP protokolü ile ana bilgisayara göndermektedir. Ana bilgisayar üzerinde çalışan program Phantom Omni haptik cihaz ile robot kontrolcü arasında veri iletişimi sağlamaktadır. Ayrıca sensörlerden okuduğu dataya göre de kullanıcıya kuvvet geri beslemesi vermektedir.

Ana bilgisayar üzerinde çalıştırılan program üzerinde, her biri farklı kontrol yöntemi entegre edilmiş program parçacıkları bulunmaktadır. Bütün konum hesaplamaları bilgisayar tarafında yapıldığı için robot kontrolcü üzerinde tek bir program çalıştırılması yeterli olmaktadır.

Şekil 3.13'te robot kontrolcü üzerindeki programın nasıl çalıştırıldığı gösterilmiştir. Bu adımlardan önce robot kontrolcünün açık konuma getirilmesi gerekir. 1-2 dakika arasında açılması beklenir. Acil durdurma butonu kontrol edilip basılı olmadığından emin olunduktan sonra el kumandasından güç aktif hale getirilir. Daha sonra "Application manager" > "Val3 applications" > "Disk" > "Ethernet_J" programı sağ ok yardımı ile yüklenir. Daha sonra yüklenen program "Run">"F8" butonlarına basılarak çalıştırılır. Ekranın üst tarafında S sembolü varsa robot hareket etmeyecektir. Bu durumda "Move/Hold" butonuna basılması gerekir. Hemen ardından bilgisayar üzerinde programların olduğu klasörde komut satırı açılır ve çalıştırılacak program "aaa.exe" yazılarak çalıştırılır. Burada "aaa" her bir kontrol yöntemi için çalıştırılan programın adı değişeceği için farklılıklar göstermektedir. En son olarak da Raspberry Pi üzerinde sensörlerden data okuyup ana bilgisayara gönderen program çalıştırılır. Bunun için de Raspberry Pi komut satırına "python VL6180X.py" yazılır. Bütün programların çalıştığından emin olunduktan sonra kullanıcı haptik cihaz ile robotu uzaktan kontrol etmeye başlayabilir. Teleoperasyon işlemi bittikten sonra, bilgisayar üzerinde çalışan program durdurulur. Daha sonra Raspberry Pi üzerinde çalışan

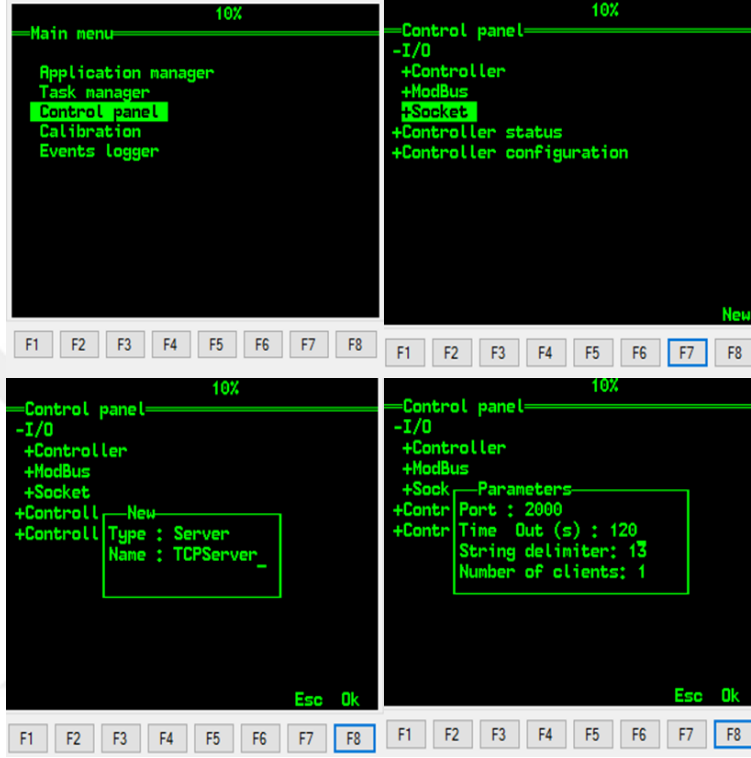
program durdurulur. En son olarak da robot kontrolcü üzerinde çalışan program durdurulur.



Şekil 3.13 : Teleoperasyon sisteminin konfigürasyonu ve çalıştırılması.

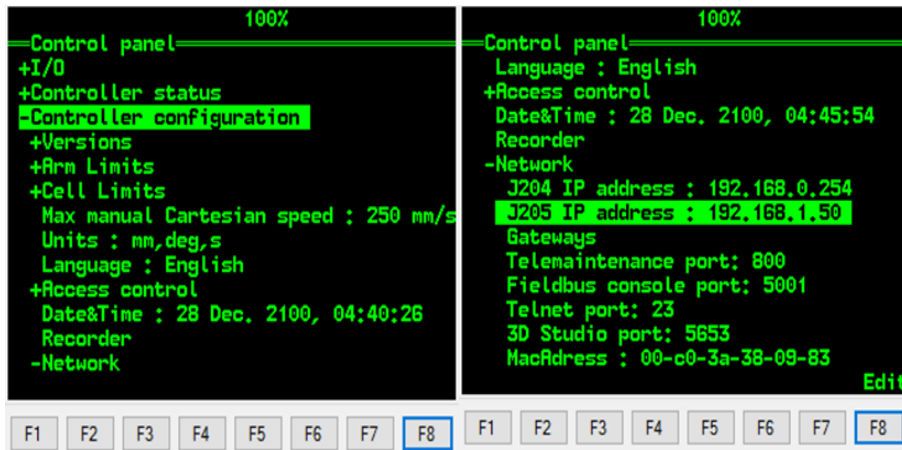
Bilgisayar, robot kontrolcü ve sensörler arasındaki veri iletişiminin sağlanması için arada düzgün bir haberleşme kanalı olması gerekmektedir. Güvenli ve hızlı olması sebebiyle TCP protokolü ile sistem bileşenleri arasındaki bağlantı sağlanmıştır. Bu da her bir alt sistemin yerel ağ bağlantısı ile birbirine bağlanıp, her birinin IP adresinde port açılarak sağlanabilmektedir. Robot kontrolcüde öncelikle Menüden “Control Panel” üzerine gelinir. Sağ ok tuşu ile alt seçenekler açılır. Daha sonra ”Socket” üzerine ok tuşları ile gelinip F8 e basılır. Bu sayede yeni bir soket oluşturmak için ekranın ortasında bir pencere açılır. Burada oluşturulacak soket tipi olarak Enter tuşuna basılıp Server seçilmelidir. “Name” kısmında da sokete isim verildikten sonra tekrar F8 e basılarak oluşturulan soketin parametreleri ayarlanır. Buradaki port numarası bilgisayar tarafından robot kontrolcüye bağlanmak için kullanılacaktır. Portun altındaki seçenek de zaman aşımını ifade etmektedir. Yani portun bağlantı kurulmadan daha ne kadar açık tutulacağını ifade etmektedir. O yüzden bunun yeterince büyük bir

değer olarak girilmesi önerilir. Bu ayarlar Şekil 3.14’te de görülmektedir. Daha sonra robot kontrolcü üzerinde çalıştırılan VAL3 programına "*sioLink(sioASCII,io:TCPServer)*" satırı eklenir. Bu sayede program çalışmaya başlarken TCPServer soketi aktif hale gelir ve robot kontrolcü tarafından sokete gelen veriler bir değişkene atanarak kullanılır.



Şekil 3.14 : Robot kontrolcü soket konfigürasyonu.

Robot kontrolcünün IP adresi de bilgisayar ile aynı adres grubunda olması gerekmektedir. Bu yüzden robot kontrolcünün J205 soketinin IP adresi Şekil 3.15’te görüldüğü gibi “192.168.1.50” olarak ayarlanmıştır.



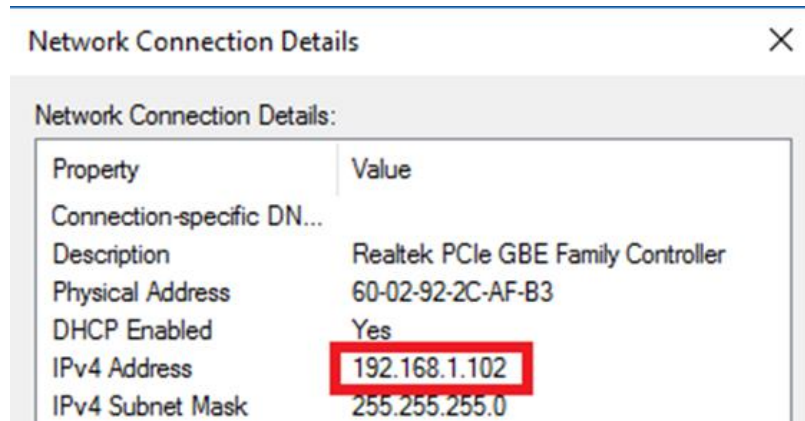
Şekil 3.15 : Robot kontrolcü ağ konfigürasyonu.

Şekil 3.16'da bilgisayar üzerinde çalıştırılan programın C++ kodlarının bir kısmı görülmektedir. Burada DEFAULT_IP değişkeni robot kontrolcü üzerinde bağlanılacak olan soketin IP adresini, DEFAULT_PORT değişkeni de bu IP üzerinde bağlanılacak olan açık portu ifade etmektedir. TCP_PORT değişkeni de daha sonra Raspberry Pi üzerinden bilgisayara data aktarımı için bilgisayar tarafında açılacak portu ifade eder.

```
#define DEFAULT_BUFLen 512
#define DEFAULT_PORT "2000"
#define DEFAULT_IP "192.168.1.50"
#define TCP_IP "192.168.1.102"
#define TCP_PORT "5005"
#define BUFFER_SIZE 30
```

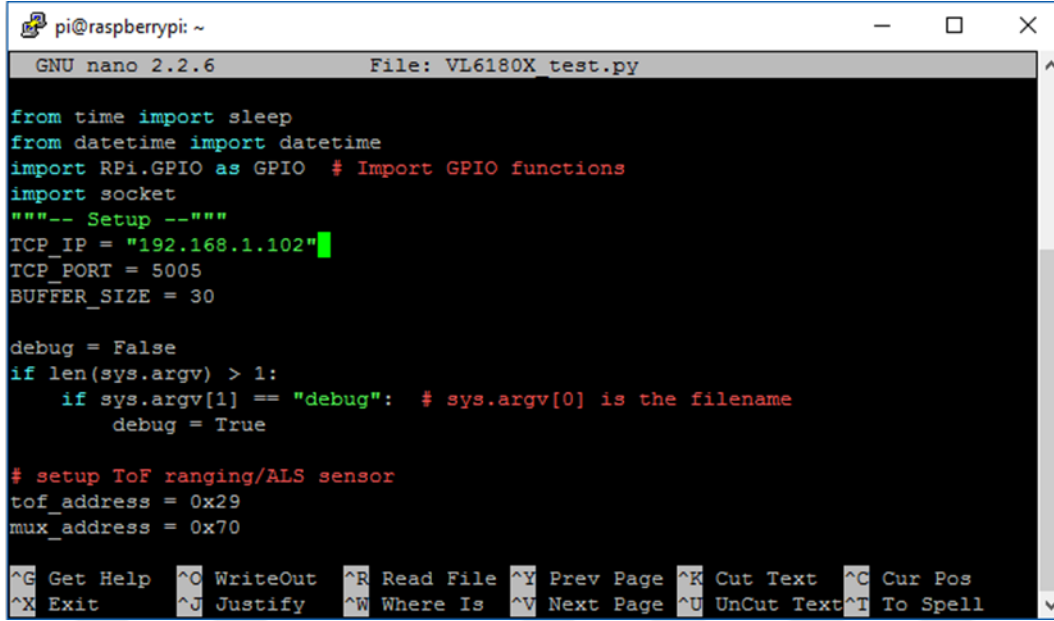
Şekil 3.16 : Bilgisayarda çalışan programın C++ kodunun bir parçası.

Raspberry Pi üzerinde çalıştırılan python kodunda, sensör datasının gönderileceği IP adresi ve port numarası tanımlanmalıdır. Şekil 3.18'de çalıştırılan python kodunun bir parçası görülmektedir. Burada TCP_IP değişkeni bağlanılan bilgisayarın IP adresini , TCP_PORT değişkeni de bağlanılacak port numarasını ifade etmektedir. Eğer bilgisayar herhangi bir nedenle başka bir IP adresine sahipse bu değer uygun bir şekilde değiştirilmelidir. Bunun için Şekil 3.17'de görüldüğü gibi ethernet özelliklerinden bilgisayarın IP adresi kontrol edilebilir.



Property	Value
Connection-specific DN...	
Description	Realtek PCIe GBE Family Controller
Physical Address	60-02-92-2C-AF-B3
DHCP Enabled	Yes
IPv4 Address	192.168.1.102
IPv4 Subnet Mask	255.255.255.0

Şekil 3.17 : Bilgisayarın ağ ayarları.



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: VL6180X_test.py

from time import sleep
from datetime import datetime
import RPi.GPIO as GPIO # Import GPIO functions
import socket
"""-- Setup --"""
TCP_IP = "192.168.1.102"
TCP_PORT = 5005
BUFFER_SIZE = 30

debug = False
if len(sys.argv) > 1:
    if sys.argv[1] == "debug": # sys.argv[0] is the filename
        debug = True

# setup ToF ranging/ALS sensor
tof_address = 0x29
mux_address = 0x70

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Şekil 3.18 : Raspberry Pi üzerinde çalışan python kod parçası.

4. DENEYLER

Bu kısımda yapılan deneyler ayrıntılı olarak incelenmiştir. Deneyler sırasında ikinci bölümde bahsedilen kontrol yöntemleri 5 farklı kullanıcı tarafından üçer farklı kuvvet seviyesinde denenmiştir. Bu kullanıcılar haptik cihaz teknolojisine yabancı olmayan kişilerden seçilmiştir. Buna göre her deney seti 15 deneyden oluşmakta olup kullanılan kontrol yöntemi ve kuvvet miktarının sırası rastgele olarak belirlenmiştir. Bunun için MATLAB programının randperm fonksiyonundan yararlanılmıştır. Aşağıdaki MATLAB kodu her çalıştırıldığında 1' den 15' e kadar sayılar 5x3' lük bir matrise rastgele olarak yerleştirilir. Bu sayılar deney sırasını ifade eder. Yani 1 ilk yapılacak deneyi ifade ediyorsa 15 de son yapılacak deneydir. Bu işlem iki defa yapılır ve iki data seti için deney sıralaması belirlenmiş olur. Böylece deneye hangi kontrol yönteminden başlanacağı ve nasıl bir sıralama ile yapılacağı bilinmediğinden kullanıcılar her deneyde tekrar öğrenme sürecine girer. Her kullanıcı deneyleri farklı sıralamada yaptığı için de herkesin öğrenme süreci birbirinden farklı olmuştur. Deney sırasının rastgele olmasındaki amaç öğrenme sürecinin sonuçlara etkisinin en aza indirilmesidir. Örneğin, kontrol yöntemlerinden biri her seferinde en son kullanılmış olsun. Kullanıcıların son deneylerde haptik cihaza daha adapte olmuş olması sebebiyle sonuçların son deneylerde daha iyi çıkması kaçınılmazdır. Eğer sıralama rastgele olursa bu adaptasyon farkları da en aza indirilmiş olur. Kullanıcılar kontrol yöntemlerini hangi sırada kullanacaklarını bilmemektedirler. Onlara verilen rastgele dizilmiş ve önceden belirlenmiş sırasıyla labirent görevini tek tek her bir kontrol yöntemi için yaparlar. Dolayısı ile kullandıkları kontrol yöntemini de görev sırasında keşfetmeleri beklenmektedir. MATLAB' da kullanılan rastgele matris üretici algoritmanın sahte kodu Çizelge 4.1' de verilmiştir.

Çizelge 4.1 : Rastgele 1-15 arası değerleri içeren matris üretici algoritmanın sahte kodu.

n değerini 15 olarak belirle

A 5x3 matrisi oluşturup 1 den 15 e kadar olan değerleri içine sırayla yerleştir

ix değişkenine randperm fonksiyonunu n değeri için uygula

Rastgele sıralanan bu sıralamayı 5x3 A1 matrisine ata

Aynı işlemleri tekrar yap bu sefer yeni sıralamayı 5x3 lük A2 matrisine ata

Çizelge 4.2' de rastgele dizilmiş deney sıraları görülmektedir. Deneyler bu sıralama ile yapılmış ve karışıklığın önüne geçmek için uzaklık algılayıcı datası ayrı ayrı kaydedilmiştir.

Çizelge 4.2 : Deney sıralamaları.

Denek 1	1. set			2. set		
	0%	50%	100%	0%	50%	100%
Konum	7	9	4	1	14	8
Hız	12	2	13	3	9	7
GKH	6	8	11	5	4	2
Balistik	1	10	3	6	11	13
Değişken Ölçekli	14	15	5	15	12	10

Denek 2	3. set			4. set		
	0%	50%	100%	0%	50%	100%
Konum	15	7	12	4	12	1
Hız	1	4	2	6	3	13
GKH	6	14	8	7	10	15
Balistik	5	13	11	5	9	2
Değişken Ölçekli	3	10	9	8	11	14

Çizelge 4.2 : Deney sıralamaları (devamı).

Denek 3	5. set			6. set		
	0%	50%	100%	0%	50%	100%
Konum	6	11	12	8	6	3
Hız	3	8	10	11	4	10
GKH	15	7	5	14	12	5
Balistik	1	14	13	13	1	2
Değişken Ölçekli	2	9	4	7	9	15

Denek 4	7. set			8. set		
	0%	50%	100%	0%	50%	100%
Konum	6	4	9	13	15	8
Hız	15	3	11	3	9	12
GKH	7	10	1	14	6	4
Balistik	13	8	12	5	11	7
Değişken Ölçekli	2	14	5	2	10	1

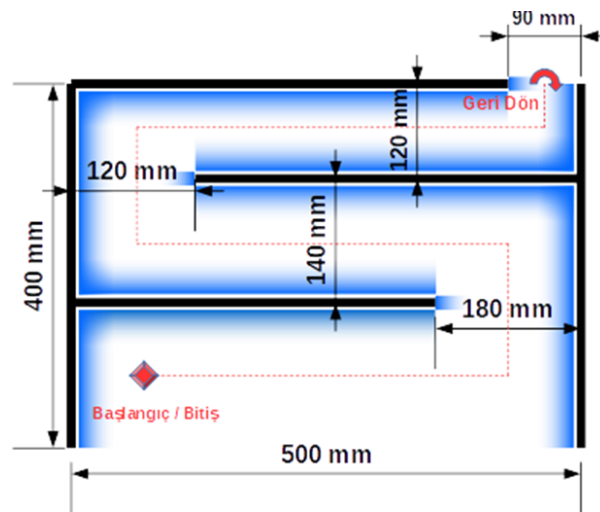
Denek 5	9. set			10. set		
	0%	50%	100%	0%	50%	100%
Konum	8	9	6	14	12	10
Hız	4	5	11	13	2	5
DKO	13	2	7	4	1	11
Balistik	12	1	15	8	6	9
Değişken Ölçekli	10	14	3	15	3	7

Deneyler sırasında kullanıcılara aşağıdaki gibi bir görev profili verilmiştir:

- Labirent duvarlarına çarpmadan robot kolu labirentin içinde belirlenmiş yörüngede hareket ettirmek, (bkz. Şekil 4.1)
- Yörüngeyi en kısa sürede bitirmeye çalışmak,
- Bu süreçte yörüngeye en yakın rotayı takip edecek şekilde robotu kontrol etmek.

Kullanıcılar bu görevleri yaparken robot kolun uç noktasındaki uzaklık algılayıcılar sayesinde de duvarlara olan yakınlıklar ölçülmektedir. Herhangi bir duvara olan uzaklık 32mm den küçük olduğunda, kullanıcıya haptik cihaz tarafından duvarın tersi yönünde tepki kuvveti uygulanmaya başlar.

Şekil 4.1' de deneyde kullanılan test düzeneği ve labirent ölçüleri görülmektedir. Labirentin duvarları 75mm yüksekliğindedir. Şekil üzerinde takip edilmesi istenen yörünge kırmızı kesikli çizgilerle gösterilmiştir. Duvar kenarlarındaki mavi renkli bölgeler kuvvet geri beslemesinin olduğu bölgelerdir. Şekilde görüldüğü gibi kuvvet geri beslemesiz bölge labirentin sonuna gittikçe daralmaktadır. İki duvar arasının en dar olduğu kısım 120mm genişliğindedir. Burada kuvvet geri beslemesi olmayan bölge 56mm genişliğinde bir koridor olarak düşünülebilir. Bu sayede yörünge zorluğu son koridorda önceki koridorlara göre bir miktar artırılmış olup, hata yapılmaya daha müsait bir ortam oluşturulmuştur. Ayrıca labirentin üzerine bir de kamera yerleştirilmiş ve görüntüler anlık olarak bilgisayara aktarılmıştır. Bu sayede kullanıcılar robot kolun uç noktasının anlık konumunu bilgisayar ekranından takip edebilmektedirler.



Şekil 4.1 : Kuvvet geri beslemeli yörünge ve labirent ölçüleri.

Uçuş zamanlı uzaklık algılayıcıların ölçümlerini düzgün şekilde yapabilmeleri için bir zaman aralığı gereklidir. Ölçülecek uzaklık aralığı ve labirentin duvarlarının yansıtma oranları bilindiğinden dolayı bu zaman aralığı da 10ms olarak belirlenmiştir. Aynı anda bütün uzaklık algılayıcılardan data alınabilecek bir devre tasarımı yapılamadığından ve bunun için gerekli algoritma geliştirilemediğinden dolayı ölçümler ve data toplama sıra ile yapılmaktadır. Bu da ölçümler için 50ms ve döngü için 10ms olmak üzere toplamda her bir data döngüsü için 60ms zaman gerektirmektedir.

Göreve başlanırken öncelikle Bölüm 3.3' de anlatılan prosedür takip edilerek program başlatılır. Ama her seferinde log dosyası ismi ve kuvvet geri besleme katsayısı uygun şekilde değiştirildikten sonra program tekrar derlenir. Daha sonra program çalıştırılıp göreve başlanır. Görev uygun şekilde bitirildikten sonra program sonlandırılıp log dosyası kaydedilir.

İçerisinde sensörlerin duvara anlık uzaklıkları ve zaman bilgisini barındıran log dosyaları daha sonra kontrol yöntemlerinin ve kuvvet miktarlarının karşılaştırmalı analizinde kullanılmak üzere kaydedilmektedir. Log dosyaları xx_logfilezz.txt şeklinde oluşturulur. Dosya isimlendirmesindeki logfile dan önce gelen xx metod isminin kısaltmasıdır. Konum kontrolü için “pc”, hız kontrolü için “rc”, GKH kontrolü için “modprc”, balistik kontrol için “bc” ve değişken ölçekli konum kontrolü için “vs” kısaltmaları kullanılmıştır. Logfile dan sonraki y denek numarasını ifade eder. Son 2 rakam olan zz de farklı kuvvet kademelerini ifade eder. Son iki rakam 00 ise kuvvet 0% , 10 ise kuvvet 50%, 11 ise kuvvet 100% anlamına gelmektedir.

Kuvvet geri beslemesi için haptik cihaz sınırlarını zorlamayacak şekilde 3 ana kuvvet seviyesi belirlenmiştir. Bunlar kullanıcıların farklı kuvvet seviyelerine göre tepkilerini anlamak içindir ve ana program içinde k sabitini değiştirerek ayarlanabilmektedir. k sabiti 0, 0.05 ve 0.10 olmak üzere 3 farklı değer almaktadır. Bu sayede duvarın en yakınında iken ulaşılan en büyük kuvvet geri besleme değeri 3.2N olmaktadır. Bu değer haptik cihazın nominal kuvvet sınırı olan 3.3N değerini geçmeyecek şekilde ayarlanmıştır. Bu sayede haptik cihaz kuvvet değerleri güvenli bölgede tutulup cihaza zarar verme ihtimali en aza indirilmiş olur. Ayrıca hız kontrolü ve GKH kontrolü için yerden uygulanan kuvvet azaltılmıştır. Bu kuvvetin azaltılmasındaki amaç yerden sürekli uygulanan bir kuvvet ile kullanıcıyı meşgul etmemektir. Böylece kullanıcı görev sırasında yörünge üzerinden gidip gelmeye odaklanabilir.

Uçuş zamanlı uzaklık sensörünün optimum ölçüm aralığı 0-100mm olduğu için sensör datası genellikle bu aralıkta kalacak şekilde filtrelenmiştir. Sadece yatay koridor boyunca olan veriler incelenirken diğer veriler ihmal edilmiştir. Filtreleme için MATLAB programı kullanılmıştır. Bu filtreleme ve kontrol yönteminde merkez yörüngeden kaçıklığı hesaplamak için kullanılan algoritmanın sahte kodu Çizelge 4.3’te görülmektedir.

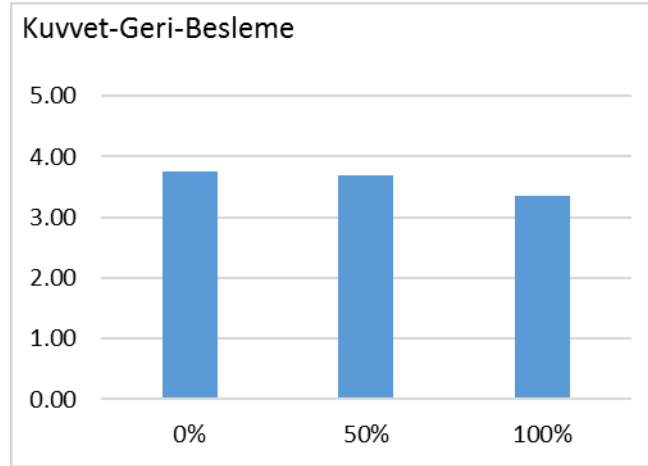
Çizelge 4.3 : Algılayıcı datasını filtreleyen ve yörüngeden kaçıklığı hesaplayan algoritmanın sahte kodu.

```
Algılayıcı datasını A matrisine ata
1 ve 3 numaralı uzaklık algılayıcı datası haricindeki datayı temizle
Toplam değişkenini ilklendir sum=0
1 ve 3 numaralı uzaklık algılayıcı datası toplamının 170 den büyük olduğu durumları
filtreleyen bir koşul yaz. Daha sonra bu koşula göre filtrele.
for (i=1 den A matrisinin uzunluğuna kadar)
    Algılayıcı verilerinin satır satır farkını alıp 2’ ye böl ve mutlak değerini alıp
    toplam üzerine ekle
end
Toplamı A matrisinin uzunluğuna böl ve ortalamayı hesapla
```


5. SONUÇLAR VE TARTIŞMA

İkinci bölümde bahsedilen kontrol yöntemleri üçüncü bölümde belirlenen test düzeneği ve yörünge baz alınarak kullanıcılar tarafından test edilmiştir. Deneylein nasıl yapıldığı dördüncü bölümde de detaylı şekilde anlatılmıştır. Bu deneyler sonucunda kullanıcılardan sayısal deney verileri elde edilmiştir. Bu veriler kullanıcıların hangi kontrol yönteminde daha kararlı ve doğru bir rota takip ettiklerini ve hangi kontrol yöntemi ile görevi daha erken bitirdiklerini karşılaştırmak için önemli rol oynamaktadır. Ayrıca kullanıcılardan kullandıkları kontrol yöntemine hemen kullandıktan sonra, kullanım kolaylığına, tercih edilebilirliğine, rahatlığına göre 1’den 5’ e kadar bir puan vermeleri istenmiştir. 1 en rahatsız ve tercih edilmez anlamına gelirken 5 de çok rahat ve tercih edilir anlamına gelmektedir.

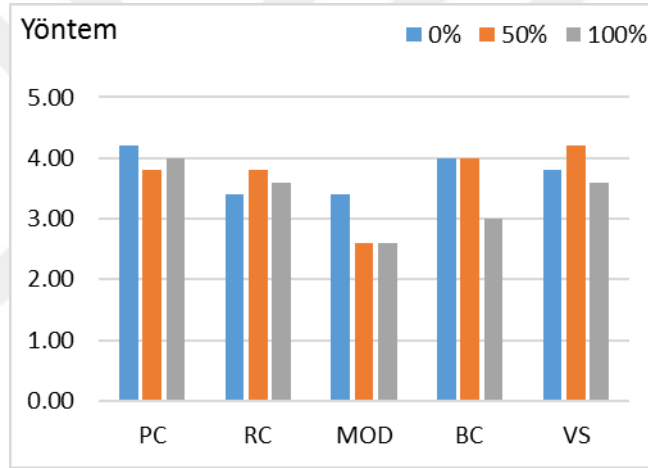
Şekil 5.1’ de kullanıcıların kuvvet geri besleme miktarlarına 5 üzerinden verdikleri puanlar görülmektedir. Buna göre teleoperasyon sistemi üzerinde hiç kuvvet geri beslemesi yokken kullanıcıların görevi daha rahat yaptıkları söylenebilir. Kuvvet geri beslemesi artırıldıkça kullanım rahatlığı ve kolaylığı azalmaktadır.



Şekil 5.1 : Kullanıcıların kuvvet geri besleme miktarlarına 5 üzerinden verdikleri puanlar.

Kullanıcıların ayrı kontrol yöntemlerine ve kuvvet miktarlarına verdikleri puanlar Şekil 5.2’ de görülmektedir. Burada “PC” konum kontrolü, “RC” hız kontrolü, “MOD” GKH kontrolü, “BC” balistik kontrol ve son olarak da “VS” değişken ölçekli konum kontrolünü ifade eder. Yapılan puanlamaya göre 4-5 arası puana sahip konum

kontrolün kuvvet geri beslemesiz ve tam kuvvet geri beslemeli hali, balistik kontrolün kuvvet geri beslemesiz ve yarı kuvvet geri beslemeli hali ve değişken ölçekli konum kontrolünün yarı kuvvet geri beslemeli hali en tercih edilen yöntemlerdendir. Daha sonra konum kontrol, balistik kontrol ve değişken ölçekli konum kontrolünün daha az tercih edilen diğer kuvvet geri beslemeli halleriyle birlikte hız kontrolü ve GKH kontrolünün kuvvet geri beslemesiz hali gelmektedir. En son olarak da GKH kontrolünün kuvvet geri beslemeli durumda tercih edilmediği görülmüştür. Kontrol yöntemleri genel olarak karşılaştırılmak istenirse ortalama puanlar konum kontrolü için 4.00, değişken ölçekli konum kontrolü için 3.87, balistik kontrol için 3.67, hız kontrolü için 3.60 ve GKH kontrolü için 2.87' dir. Burada konum kontrolünün en tercih edilen kontrol yöntemi olmasının sebebinin en geleneksel yöntem olmasından kaynaklandığı düşünülmektedir.

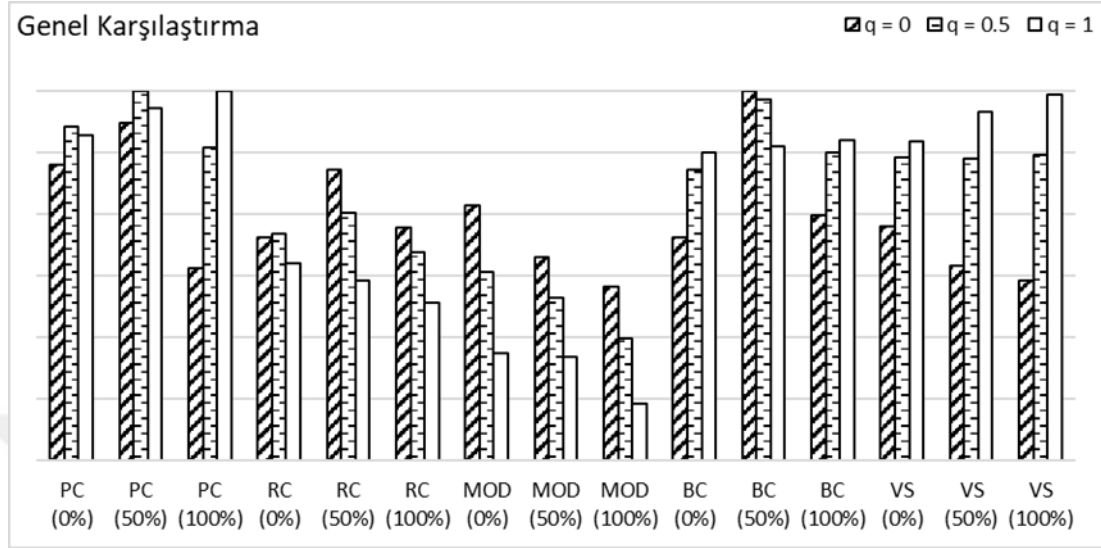


Şekil 5.2 : Kontrol yöntemleri ve kuvvet miktarlarının ayrı ayrı puan değerleri.

Tüm deneylerden toplanan uzaklık algılayıcı verileri incelendiğinde daha tarafsız sonuçlar elde edilmiştir. Şekil 5.3' te kullanılan tüm kontrol yöntemlerinin kuvvet miktarları ile birlikte genel bir karşılaştırılması yapılmıştır. Burada dikkate değer sonuçlar ortaya çıkmıştır. Her yöntem için bitirme zamanı (t) ve yörüngeden kaçıklık miktarları (e) değerleri tablo haline getirilmiştir. Bu değerlerin küçük olması daha iyi olduğu için en küçük değer 100 puan alırken diğerleri de ters orantılı şekilde daha az puan alacak şekilde normalleştirme işlemi gerçekleştirilmiştir. Bu sayede karşılaştırma daha mantıklı hale getirilmiştir. Grafik üzerinde q göreceli olarak performans puanı için bitirme zamanının hata miktarına göre ağırlık katsayısını ifade etmektedir. Yani bu değer 1 olduğunda sadece bitirme zamanına, 0 olduğunda da sadece hata miktarına bakılarak performans puanı hesaplanır. Ara değerlerden biri olan 0.5 değerinde ise hata

miktarı ve bitirme zamanının puan ağırlıkları aynıdır. Bu puan hesaplaması Denklem (5.1)' te görülmektedir.

$$x = qt + (1 - q)e \quad (5.1)$$

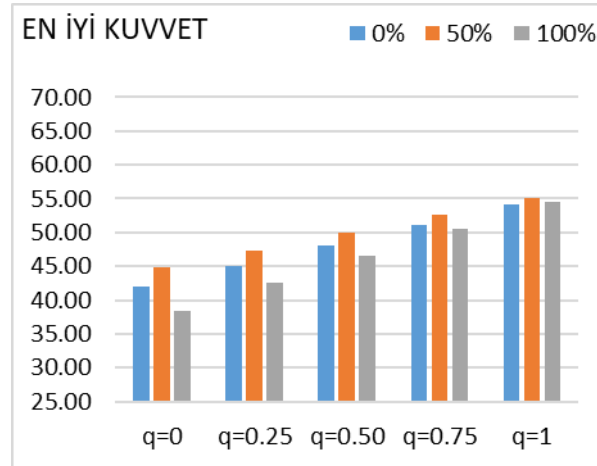


Şekil 5.3 : Kontrol yöntemlerinin hata-bitirme zamanına göre kendi içinde analizi.

Kontrol yöntemleri tek tek kendi içinde incelenirse değişken ölçekli konum kontrolü için kuvvet geri beslemesi artırıldıkça zaman ağırlıklı puan artarken hata ağırlıklı puan azalmaktadır. Eşit ağırlıklı puanlamada ise kuvvet değişiminin değişken ölçekli konum kontrolü yöntemi üzerinde pek bir etkisi görülmemiştir. Ayrıca değişken ölçekli konum kontrolünde en yüksek puanlar zaman ağırlıklı olanlardır. Balistik kontrole bakıldığında yarı kuvvet geri besleme değerinde hata ağırlıklı ve eşit ağırlıklı puanlarda en iyi puan değerine ulaşmıştır. Bitirme zamanı açısından bakıldığında ise kuvvet artışının balistik kontrol ile görevi bitirme süresine çok az bir etkisi olduğu görülmüştür. Konum kontrol incelenirse yarı kuvvet geri besleme değerinde hata ağırlıklı ve eşit ağırlıklı puanlamada en yüksek değerlere ulaştığı görülmektedir. Kuvvet artışı konum kontroldeki bitirme zamanını olumlu yönde ve önemli ölçüde etkilemiş olup en yüksek kuvvet geri besleme değerinde ve zaman ağırlıklı puanlamada en yüksek değere ulaşmasını sağlamıştır. Hız kontrolüne bakıldığında yarı kuvvet geri beslemesinde hata ağırlıklı ve eşit ağırlıklı puanların konum kontroldekine benzer şekilde en iyi değerlerine ulaştığı görülmektedir. Ancak konum kontroldekinden farklı olarak kuvvet geri besleme miktarı artışı hız kontrolünün zaman ağırlıklı puanının düşmesine sebep olmuştur. GKH kontrolünde ise kuvvet artışı bütün puanların düşmesine sebep olmuştur. Bu kontrol yönteminde en yüksek

puanlar hep hata ağırlıklı puanlardır. Kontrol yöntemlerini birbirleri ile karşılaştırmak gerekirse bunlar bu üç ayrı puanlamaya göre yapılabilir. Hata ağırlıklı puanlamaya bakılırsa yarı kuvvet geri beslemeli balistik kontrol en iyi sonucu vermiştir. Eşit ağırlıklı puanlamada bakılırsa konum kontrol ve balistik kontrolün yarı kuvvet geri beslemeli durumu en iyi sonuçlara sahiptir. Zaman ağırlıklı puanlamada ise konum kontrol ve değişken ölçekli konum kontrolünün tam kuvvet geri beslemeli durumunda en iyi sonuçlara ulaşılmıştır. Hız kontrolü ve GKH kontrolü hata bakımından kendi içlerinde en iyi sonuçlara sahip olmalarına rağmen diğer yöntemlerle karşılaştırıldığında o kadar da iyi sonuçlara ulaşamadıkları görülmüştür. Sonuç olarak görev tanımında hatanın mı yoksa bitirme zamanının mı daha önemli olduğuna bağlı olarak en uygun yöntem hangisi ise o seçilmelidir.

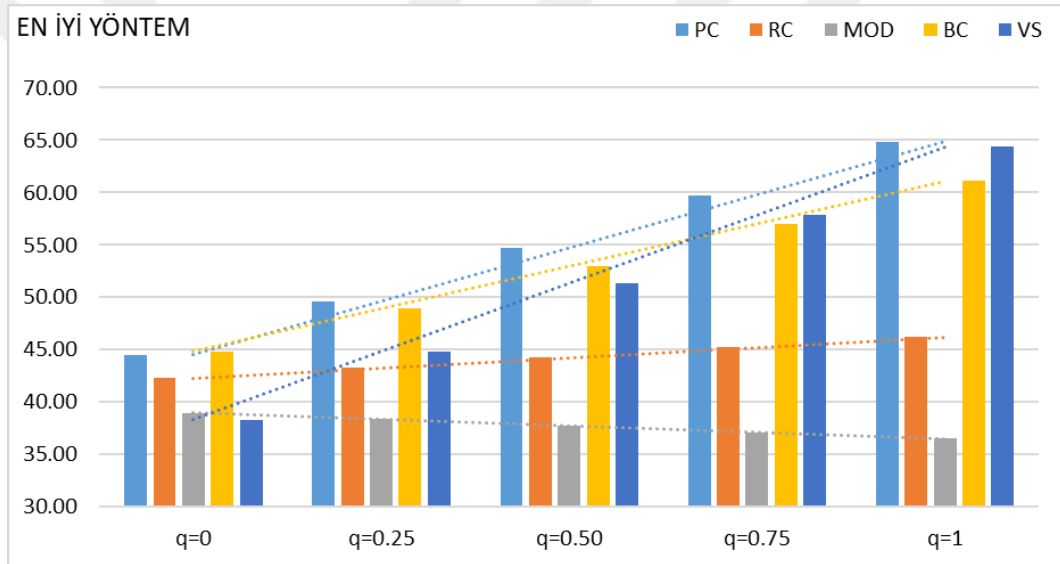
Şekil 5.4'te kuvvet miktarları kendi aralarında karşılaştırılmıştır. Bunun için her bir kontrol yönteminin belirli kuvvet geri besleme miktarına göre performans puanları alınıp bunların o kuvvet miktarındaki ortalamaları alınmıştır. Burada 0%, 50% ve 100% kuvvet miktarları olmak üzere üç seviye vardır. Burada yine q bitirme zamanının ağırlık katsayısıdır. 0' dan 1'e kadar olan ara değerler bitirme zamanının ağırlıklarıdır. Hem hata hem de bitirme zamanı göz önüne alındığında ikisinde de 50% kuvvet geri besleme miktarı en iyi performansı göstermiştir.



Şekil 5.4 : Kuvvet miktarlarının hata miktarı – bitirme zamanına göre performans grafiği.

Şekil 5.5' te deney sırasında kullanılan 5 kontrol yöntemi performans açısından karşılaştırılmıştır. Bitirme zamanı ve hata miktarları arasında yine q gibi bir ağırlık katsayısı kullanıldığında kontrol yöntemleri bu iki açıdan ve kendileri arasında karşılaştırılabilir duruma gelmiştir. Grafiğe bakıldığında tek bir yöntemin en iyi

performansı sergilediği söylenemez. Bitirme zamanının önemli olduğu durumlarda konum kontrolü iyi sonuçlar vermiştir. Hata oranı göz önüne alındığında ise balistik kontrol daha iyi sonuçlar vermiştir. Yani balistik kontrol ile hedeflenen yörüngeye daha çok yaklaşmıştır. Değişken ölçekli konum kontrolü ise bitirme zamanı açısından iyi sonuçlar verse de hata miktarlarına bakıldığında diğer yöntemlere göre çok kötü sonuçlar vermiştir. Bu 5 kontrol yöntemine genel olarak bakıldığında GKH kontrolü diğer kontrol yöntemleri arasında en kötü sonuçları vermiştir. Hız kontrolü hata miktarları açısından bakıldığında ortalama sonuçlar vermiş olmasına rağmen hız kontrolü olduğu için bitirme zamanı olarak diğer konum kontrollerinden geride kalması beklenen bir sonuçtur. Bu sonuçlar göz önüne alındığında hedeflenen görev tanımına göre seçilebilecek en iyi yöntem de değişiklik göstermektedir.



Şekil 5.5 : Kontrol yöntemlerinin karşılaştırmalı performans değerleri.

Bu çalışmanın hedeflerinden biri olan uzaklık sensörü yardımı ile endüstriyel bir robotun uzaktan kontrolünün mümkün olduğu görülmüştür. Sensörler herhangi bir çarpışmayı engellemektedir fakat kullanıcının robot uç noktasını etraftaki nesnelere çarpmasını engelleyecek şekilde kuvvet geri beslemesi sağlanması için etraftaki nesnelere aradaki mesafeyi anlık olarak bilgisayara geri bildirmektedir. Hassasiyeti mm düzeyinde olması istenen ve çok hassas olmayan işlerde bu sensörlerden yararlanılarak düşük maliyetli bir şekilde robot-çevre güvenliği sağlanabilir. Burada ayrıca daha önceden başka çalışmalarda uygulanmış kontrol yöntemlerinin tekrar karşılaştırması yapıp GKH kontrolü yöntemi önerilmiştir. Fakat parametrelerin düzgün şekilde ayarlanamamasından dolayı bu kontrol yöntemi diğerlerinin performans açısından gerisinde kalmış olduğu düşünülmektedir. Bu önerilen

yöntemde iki farklı kontrol yöntemi aynı iş uzayı içerisinde olduğundan birbirlerinin kontrol alanını daraltmaktadırlar. Bu da kontrolü zorlaştıran faktörlerden biri olarak düşünülebilir. Ayrıca kuvvet dağılımı ilk başta her eksen için (x,y ve z) eşit şekilde tasarlandığından dolayı yerden kullanıcıya uygulanan kuvvet kontrolde zorluklara sebep olmuştur. Kullanıcılardan alınan geri bildirimler baz alınarak düzenlenen son deneyde kuvvet geri beslemesi düşey ekseninde onda birine düşürülmüştür. Bunun dışında uzaklık algılayıcılardan sıra ile data alınmasından kaynaklanan gecikmeler sebebiyle de sistemin herhangi bir girdiye veya bozuntuya verdiği tepki değişeceğinden dolayı sonuçlar etkilenmiş olabilir. Teleoperasyon sisteminin gerçek zamanlı bir sisteme daha yakın hale getirilebilmesi için sistem üzerinde daha çok değişiklik yapılması gerekmektedir. Bundan sonraki çalışmalarda farklı kontrol yöntemleri bu teleoperasyon sistemine adapte edilip test edilebilir. Halihazırda test edilen kontrol yöntemlerinin katsayıları daha da iyileştirilip robot için birden fazla yörünge ve test düzeneği belirlendikten sonra tekrar test edilebilir.

KAYNAKLAR

- [1] **Bolmsjö, G., Olsson, M., & Cederberg, P.** (2002). Robotic arc welding—trends and developments for higher autonomy. *Industrial Robot: An International Journal*, 29(2), 98-104.
- [2] **Sugita, S., Itaya, T., & Takeuchi, Y.** (2004). Development of robot teaching support devices to automate deburring and finishing works in casting. *The International Journal of Advanced Manufacturing Technology*, 23(3-4), 183-189.
- [3] **Choi, M. H., & Lee, W. W.** (2001). A force/moment sensor for intuitive robot teaching application. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (Vol. 4, pp. 4011-4016). IEEE.
- [4] **Schraft, R. D., & Meyer, C.** (2006). The need for an intuitive teaching method for small and medium enterprises. *VDI BERICHTE, 1956*, 95.
- [5] **Pan, Z., & Zhang, H.** (2007). Robotic programming for manufacturing industry. In *Proceedings of ICMEM, international conference on mechanical engineering and mechanics. Wuxi, China*.
- [6] **Zhang, H., Chen, H., Xi, N., Zhang, G., & He, J.** (2006). On-line path generation for robotic deburring of cast aluminum wheels. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (pp. 2400-2405). IEEE.
- [7] **Solvang, B., Sziebig, G., & Korondi, P.** (2008). Robot programming in machining operations. In *Robot Manipulators*. InTech.
- [8] **Nicholson, A. D.** (2005). Rapid adaptive programming using image data.
- [9] **Pan, Z., Polden, J., Larkin, N., Van Duin, S., & Norrish, J.** (2012). Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 28(2), 87-94.
- [10] **Biggs, S. J., & Srinivasan, M. A.** (2002). Haptic interfaces. *Handbook of virtual Environments*, 93-116.
- [11] **Van der Linde, R. Q., Lammertse, P., Frederiksen, E., & Ruiters, B.** (2002). The HapticMaster, a new high-performance haptic interface. In *Proc. Eurohaptics* (pp. 1-5).
- [12] **Jia, S., & Pan, Z.** (2010). A preliminary study of suture simulation in virtual surgery. In *Audio Language and Image Processing (ICALIP), 2010 International Conference on* (pp. 1340-1345). IEEE.
- [13] **Escobar-Castillejos, D., Noguez, J., Neri, L., Magana, A., & Benes, B.** (2016). A review of simulators with haptic devices for medical training. *Journal of medical systems*, 40(4), 104.

- [14] **Chen, Y., Zhang, J., Yang, C., & Niu, B.** (2007). The workspace mapping with deficient-DOF space for the PUMA 560 robot and its exoskeleton arm by using orthogonal experiment design method. *Robotics and Computer-Integrated Manufacturing*, 23(4), 478-487.
- [15] **Reinhart, G., & Radi, M.** (2009, January). Some issues on integrating telepresence technology into industrial robotic assembly. In *Proceedings of the international conference on control and automation, Dubai*.
- [16] **Shirwalkar, S., Singh, A., Sharma, K., & Singh, N.** (2013, December). Telemanipulation of an industrial robotic arm using gesture recognition with Kinect. In *Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on* (pp. 1-6). IEEE.
- [17] **Pérez, A., & Rosell, J.** (2011). An assisted re-synchronization method for robotic teleoperated tasks. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (pp. 886-891). IEEE.
- [18] **Azorin, J. M., Sabater, J. M., Paya, L., & Garcia, N.** (2004). Kinematics correspondence & scaling issues in virtual telerobotics systems. In *Automation Congress, 2004. Proceedings. World* (Vol. 15, pp. 383-388). IEEE.
- [19] **Jaju, A., Das, A. P., & Pal, P. K.** (2013). Evaluation of motion mappings from a Haptic Device to an Industrial Robot for effective master-slave manipulation. *International Journal of Robotics and Automation*, 28(1).
- [20] **Denavit, S., Hartenberg, R. S.,** (1955). "A Kinematic Notation for Lower Fair Mechanisms Based on Matrices", *ASME Journal of Applied Mechanica*, pp. 215-221.
- [21] **Paul, R. P.,** Robot Manipulators: Mathematics, Programming, and Control, The MIT Press, 1982
- [22] **Teklemariam, H. G., & Das, A. K.** (2015). A case study of phantom omni force feedback device for virtual product design. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 1-12.
- [23] **Liu, L., Liu, G., Zhang, Y., & Wang, D.** (2014). A modified motion mapping method for haptic device based space teleoperation. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on* (pp. 449-453). IEEE.
- [24] **Mamdouh, M., & Ramadan, A. A.** (2012). Development of a teleoperation system with a new workspace spanning technique. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on* (pp. 1570-1575). IEEE.
- [25] **Jaju, A., Das, A. P., & Pal, P. K.** (2013). Evaluation of motion mappings from a Haptic Device to an Industrial Robot for effective master-slave manipulation. *International Journal of Robotics and Automation*, 28(1).
- [26] **Klomp, F. M.** (2006). Haptic Control for Dummies: An introduction and analysis. *Eindhoven, August*.

- [27] **Salisbury, K., Conti, F., & Barbagli, F.** (2004). Haptic rendering: introductory concepts. *IEEE computer graphics and applications*, 24(2), 24-32.
- [28] **Sanfilippo, F., Hatledal, L. I., Styve, A., Pettersen, K. Y., & Zhang, H.** (2016). Integrated flexible maritime crane architecture for the offshore simulation centre AS (OSC): A flexible framework for alternative maritime crane control algorithms. *IEEE Journal of Oceanic Engineering*, 41(2), 450-461.
- [29] **Massie, T. H., & Salisbury, J. K.** (1994). The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems* (Vol. 55, No. 1, pp. 295-300).
- [30] **Chotiprayanakul, P., & Liu, D. K.** (2009). Workspace mapping and force control for small haptic device based robot teleoperation. In *Information and Automation, 2009. ICIA'09. International Conference on* (pp. 1613-1618). IEEE.
- [31] **Voskuil, M. J.** Hybrid design combining position and rate control allows intuitive workspace extension for teleoperation.
- [32] **Silva, A. J., Ramirez, O. A. D., Vega, V. P., & Oliver, J. P. O.** (2009). Phantom omni haptic device: Kinematic and manipulability. In *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09.* (pp. 193-198). IEEE.
- [33] **Gültekin Y.**, Bir Endüstriyel Robotun İnsan Kolu Hareketlerinin Ataletsel Ölçümü İle Uzaktan Kontrolü, *Yüksek Lisans Tezi*, TOBB ETÜ Fen Bilimleri Enstitüsü, Ankara
- [34] “VL6180X Proximity and ambient light sensing (ALS) module” Erişim adresi: <http://www.st.com/content/ccc/resource/technical/document/datasheet/c4/11/28/86/e6/26/44/b3/DM00112632.pdf/files/DM00112632.pdf/jcr:content/translations/en.DM00112632.pdf> , Erişim tarihi: 02.08.2017.



ÖZGEÇMİŞ

Ad-Soyad : Ahmet Aksel
Uyruğu : T.C.
Doğum Tarihi ve Yeri : 07.06.1991 / Antalya
E-posta : ahmetaksel33@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2014, TOBB Ekonomi ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği (Anadal)
2014, TOBB Ekonomi ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Elektrik ve Elektronik Mühendisliği (Çift Anadal)
- **Yükseklisans** : 2017, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Makine Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2013	YİĞİT AKÜ	Stajyer
2012	OTOKAR	Stajyer
2014-2017	TOBB ETÜ	Özel B.B. YL ögr.

YABANCI DİL: İngilizce (İleri düzeyde), Almanca (Temel düzeyde)

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Aksel A.**, Taşcıoğlu Y., Ercan Y., 2017. Force-feedback robotic teleoperation using Time-of-Flight proximity sensors, *ICMSCE 2017 International Conference on Mechatronics Systems and Control Engineering*

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- **Aksel A.**, Gönül Ö., Çöllü Ö., Taşcıođlu Y., 2014. Mini Stewart Platform, 5. *Ulusal Mekatronik Mühendisliđi Öğrenci Kongresi (MeMÖK2014)*, Ankara, Turkey.

