Emniyetli İnsan-Robot Etkileşimi İçin Çarpışmasız Yol Planlama Yöntemi Geliştirilmesi

Sezgin Seçil

DOKTORA TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Kasım 2021

A Collision Free Path Planning Method For Safe Human-Robot Interaction

Sezgin Seçil

DOCTORAL DISSERTATION

Department of Electrical-Electronics Engineering

November 2021

Emniyetli İnsan-Robot Etkileşimi İçin Çarpışmasız Yol Planlama Yöntemi Geliştirilmesi

Sezgin Seçil

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Kontrol ve Kumanda Sistemleri Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Metin ÖZKAN

Kasım 2021

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Metin ÖZKAN danışmanlığında hazırlamış olduğum "**Emniyetli İnsan-Robot Etkileşimi İçin Çarpışmasız Yol Planlama Yöntemi Geliştirilmesi**" başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 17/11/2021

Sezgin Seçil

ÖZET

Endüstriyel robot manipülatörlerinin özellikle verimlilik açısından sağladığı faydalar, otomotiv, elektronik, kimya gibi birçok farklı endüstriyel alanda kullanımlarının artmasıyla sonuçlanmaktadır. Endüstriyel çalışma ortamlarında görülen robot varlığındaki artışa rağmen, robot manipülatörler emniyet endişeleri sebebiyle çoğunlukla insanlardan izole bir şekilde faaliyet göstermektedir. Ancak robotların üretim ve imalat süreçlerine katılımının artması, insanların robotlarla aynı çalışma alanını paylaşmalarını ve dolayısıyla etkileşime girmelerini kaçınılmaz kılmaktadır. Teknolojik gelişmelerin beraberinde, bu etkileşim aynı zamanda robotun ve insanın kabiliyetlerini bir araya getiren bir iş birliği halini de alabilmektedir. İnsan ve robotun aynı çalışma alanını paylaşmasına sebep olan bu gerçek, beraberinde emniyet sorununu ortaya çıkarmaktadır. Bu bağlamda, sürdürülebilir bir insan-robot etkileşimi için ele alınması gereken temel gereksinimlerden biri emniyet olmaktadır. Tez kapsamında, emniyetli insan-robot etkileşimini sağlayabilmek için ilk olarak insan ile robot arasındaki minimum uzaklığın hesaplanmasına dayanan bir sistem önerilmektedir. Hesaplamanın hızlı bir şekilde gerçekleştirilebilmesi için insan ve robot geometrik ilkellerle (kapsüller) temsil edilmektedir. Kapsüllerin geometrileri gereği dışbükey cisimler olmasından ve hesaplamanın gerçek zamanlı uygulamalar için hızlı olması gerektiğinden dolayı uzaklık hesaplamaları için Gilbert–Johnson–Keerthi (GJK) algoritması kullanılmaktadır. Önerilen sistemde, insanın temsili için ise RGB-D algılayıcıdan alınan iskelet verisi kullanılmaktadır. Bu tezde, hem benzetim ortamındaki robot manipülatörle, hem de gerçek ortamdaki robot manipülatörle gerçekleştirilen deneyler sonucunda önerilen sistemin performansı değerlendirilmekte ve nicel analizi yapılmaktadır. Ek olarak, önerilen sistem gerçek zamanlı bir iş birlikçi senaryo gerçekleştirilerek test edilmektedir. Sonuçlar, sistemin gerçek zamanlı ve emniyetli insan-robot etkileşimi uygulamaları geliştirmek için kullanılabileceğini göstermektedir. Çalışmanın geri kalanında, emniyetli insan-robot etkileşiminin bir başka kriteri olan çarpışmasız robot hareketi için yeni bir yol planlama yaklaşımı önerilmektedir. Önerilen yaklaşım, optimizasyon temelli bir yaklaşım olup, stokastik ve yinelemeli bir yol planlama stratejisi içermektedir. Stratejiye göre, robot ile insan arasındaki uzaklığa bağlı olarak yeniden planlama yapılmakta ve bu aşamada optimizasyon temelli bir navigasyon yöntemi takip edilmektedir. Önerilen yöntem, benzetim ortamında farklı senaryolar için test edilmektedir. Sonuçlar algoritmanın emniyetli etkileşim için etkinliğini göstermektedir.

Anahtar Kelimeler: İnsan-Robot Etkileşimi, Endüstriyel Robotlar, Robot Manipülatörler için Yol Planlama, Minimum Uzaklık Hesaplaması

SUMMARY

The benefits of industrial robot manipulators, especially in terms of efficiency, result in their increasing use in many different industrial areas such as automotive, electronics, chemistry. Despite the increase in the presence of robots in industrial working environments, industrial robot manipulators mostly operate in isolation from humans due to safety concerns. However, the increasing participation of robots in production and manufacturing processes makes it inevitable for humans to share the same workspace and therefore interact with robots. Along with technological developments, this interaction can also become a collaboration that brings together the capabilities of the robot and the human. This fact, which causes the human and the robot to share the same workspace, brings about the problem of safety. In this context, one of the key requirements to be addressed for a sustainable human-robot interaction is safety. Within the scope of the dissertation, a framework based on calculating the minimum distance between human and robot is firstly proposed in order to ensure safe human-robot interaction. Human and robot are represented by geometric primitives (capsules) so that the calculation can be performed quickly. Since the capsules are convex bodies due to their geometry and the calculation must be fast enough for real-time applications, the Gilbert–Johnson–Keerthi (GJK) algorithm is used for distance calculations. In the proposed framework, the skeletal data from the RGB-D sensor is used for the representation of the human. In this dissertation, the performance of the proposed framework is evaluated and quantitative analysis is made as a result of the experiments carried out both with the simulated robot manipulator and with the real robot manipulator. In addition, the proposed framework is tested by performing a real-time collaborative scenario. The results show that the framework is utilizable for developing real-time safe human-robot interaction applications. In the rest of the study, a novel path planning approach is proposed for collision-free robot motion, which is another criterion of safe human-robot interaction. The proposed approach is an optimization-based approach and includes a stochastic and iterative path planning strategy. According to the strategy, replanning is done depending on the distance between the robot and the human, and an optimization-based navigation method is followed at this stage. The approach is tested for different scenarios in the simulation environment. The results demonstrate the effectiveness of the algorithm for safe interaction.

Keywords: Human-Robot Interaction, Industrial Robots, Path Planning for Robot Manipulators, Minimum Distance Calculation

İÇİNDEKİLER

Sayfa

ÖZET . . . . . **vi**

SUMMARY . . . . . **vii**

TEŞEKKÜR . . . . . **viii**

İÇİNDEKİLER . . . . . **ix**

ŞEKİLLER DİZİNİ . . . . . **xi**

ÇİZELGELER DİZİNİ . . . . . **xiv**

SİMGELER VE KISALTMALAR DİZİNİ . . . . . **xv**

1. GİRİŞ VE AMAÇ **1**

2. LİTERATÜR ARAŞTIRMASI **5**

2.1. İnsan-Robot Etkileşimi ve Emniyet 5

2.2. Emniyetli Çalışma Ortamının Algılanması 7

2.3. Minimum Uzaklık Hesaplaması 9

2.4. Çarpışmasız Robot Hareketi ve Robot Denetimi 10

3. İNSAN-ROBOT ETKİLEŞİMİ **15**

3.1. Etkileşim Türleri 15

3.2. İş Birlikçi Operasyonlar 17

3.3. Geleneksel Endüstriyel Robotlar ve İş Birlikçi Robotlar 20

4. GÜVENLİ ÇALIŞMA ORTAMININ ALGILANMASI VE MODELLENMESİ **22**

4.1. Geometrik İlkeller 22

4.1.1. Kapsül geometrisi 23

4.2. İnsan ve Robotun Temsili 24

4.2.1. Robotun geometrik temsili 24

4.2.2. İnsanın algılayıcı temelli geometrik temsili 26

4.2.2.1. RGB-D algılayıcı ile iskelet takibi 27

4.2.2.2. İnsana ait kapsüllerin algılayıcıya göre dönüşümü 27

4.2.2.3. Algılayıcının el-göz kalibrasyonu 29

İÇİNDEKİLER (devam)

Sayfa

5. GÜVENLİ ETKİLEŞİM YÖNTEMİ 35
5.1. Minimum Uzaklık Hesaplama 35
5.1.1. GJK uzaklık algoritması 35
5.1.2. Sistem tasarımı 38
5.2. Çarpışmasız Yerel Yol Planlama 39
5.2.1. Tekdüze dağılımlı rastgele nokta örnekleme 40
5.2.1.1. Varsayımsal küre yüzeyinde örnekleme 40
5.2.1.2. Varsayımsal yarım küre yüzeyinde örnekleme 43
5.2.2. Emniyetli mesafe parametreleri 48
5.2.3. Yerel yol planlama algoritması 51
5.2.4. Optimizasyon problemi 53

6. BULGULAR VE TARTIŞMA 57
6.1. İskelet Takibi Temelli Minimum Uzaklık Hesaplama için Deneysel Çalışmalar 57
6.1.1. Deney kurulumu 57
6.1.1.1. Robot temsili 58
6.1.1.2. İnsan temsili 59
6.1.2. Benzetim ortamındaki robot ile gerçekleştirilen deneylere ait sonuçlar 60
6.1.3. Gerçek ortamdaki robot ile gerçekleştirilen deneylere ait sonuçlar . 67
6.2. Çarpışmasız Yol Planlama için Deneysel Çalışmalar 74
6.2.1. Tek engel durumu 76
6.2.2. Çoklu engel durumu 80
6.2.3. Hareketli engel durumu 84
6.2.4. Ağırlık parametrelerinin analizi 92
6.2.5. Hesaplama sürelerinin analizi 94

7. SONUÇ VE ÖNERİLER 97

KAYNAKLAR DİZİNİ 101

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

3.1 İnsan-robot etkileşimi türleri: (a) etkileşim yok, (b) bir arada bulunma, (c) birlikte çalışma, (d) sıralı çalışma, (e) iş birliği 17
3.2 İş birlikçi operasyon tipleri: (a) emniyet dereceli gözlenen durdurma (SMS), (b) elle yönlendirme (HG), (c) hız ve ayırma gözlemi (SSM), (d) güç ve kuvvet sınırlama (PFL) 19
3.3 Geleneksel endüstriyel robot manipülatörleri: (a) Kawasaki RS005L (Kawasaki, 2020), (b) ABB IRB 120 (ABB, 2020), İş birlikçi endüstriyel robot manipülatörleri: (c) KUKA LBR iiwa (KUKA, 2020), (d) Universal Robots UR5 (UniversalRobots, 2020) 21

4.1 Uzaklık hesaplama sistem şeması 22
4.2 Kapsül geometrisi 23
4.3 Bir endüstriyel robot manipülatörün 3-B modeli ve geometrik temsili 25
4.4 Robot ile algılayıcının koordinat sistemleri ve iskelet eklemlerinin orijinleri . . 27
4.5 İnsan temsili için kullanılan kapsüllerin konumunun ve yöneliminin hesaplanması 28
4.6 ”El-göz kalibrasyon kurulumu 30
4.7 ARUCO MIP 36h12 sözlüğünde bir ArUco işaretçisi (Garrido-Jurado vd., 2016) 31
4.8 AX = XB kalibrasyon problemi 33
4.9 El-göz kalibrasyonu: T_P^{TCP} dönüşümünün hesaplanması 34

5.1 İnsan ve robot kapsülleri arasındaki minimum uzaklık hesaplamasının Rviz’de gösterimi 37
5.2 Önerilen sistemin uygulama şeması 38
5.3 Önerilen yöntemin genel görünümü 39
5.4 Küresel koordinatlar ve diferansiyel yüzey alanı elemanı 41
5.5 Yarım küre yüzeyindeki noktaların dönüşümü 45
5.6 Önerilen yönteme göre nokta örneklenmesi: (a) birim yarım küre yüzeyinde (b) birim küre yüzeyinde 47
5.7 Emniyetli mesafe parametreleri: L_{safe}, L_{crit}, L_{danger}, insan kapsülü $C_j^{'}$, robot kapsülü C_i ve hesaplanan minimum uzaklık D_{min} 49
5.8 Sınır parametrelerine göre yol planlama ve örnekleme 50

6.1 Sekiz deney sırasında elde edilen imgeler (sırasıyla ilk kare, ara karelerin birleşimi, son kare) 61

ŞEKİLLER DİZİNİ (devam)

Şekil		Sayfa
6.2	İlk deney esnasında insan iskeletinin eklem konumları ve onlarla ilişkili zaman damgaları (Yaklaşık 450 ms zaman aralığı ile sadece belirli örnekler)	62
6.3	Her bir insan kapsülünden üç robot kapsülüne minimum uzaklık hesaplaması sırasında geçen süreler t_d: (a) minimum geçen süre, (b) ortalama geçen süre, (c) maksimum geçen süre	64
6.4	Minimum uzaklık hesaplama sürelerine ait verilerin istatistiği: medyan, alt ve üst çeyrekler, minimum ve maksimum değerler	65
6.5	Uzaklık hesaplama esnasında geçen sürelere ait histogram	66
6.6	Senaryo-1'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 100 mm/sn)	68
6.7	Senaryo-1'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 250 mm/sn)	69
6.8	Senaryo-2'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 100 mm/sn)	70
6.9	Senaryo-2'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 250 mm/sn)	71
6.10	Senaryo-3'te robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 100 mm/sn)	72
6.11	Senaryo-3'te robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 250 mm/sn)	73
6.12	ABB IRB 120 6-eksenli endüstriyel robot manipülatörün geometrik temsili (ABB, 2020)	75
6.13	Planlama ortamı: küresel yol planına göre uç noktasının izlediği yol (kırmızı), robotun başlangıç (camgöbeği) ve hedef (turuncu) konfigürasyonları, insan kapsülleri (yeşil)	75
6.14	Tek engel senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı	77
6.15	Tek engel senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülü (sol alt kol C_3') arasındaki minimum uzaklık hesaplamaları	77
6.16	Tek engel senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı	78
6.17	Tek engel senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı	79
6.18	Çoklu engel senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı	81

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

6.19 Çoklu engel senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülleri (sol alt kol C_3', sağ alt kol C_4') arasındaki minimum uzaklık hesaplamaları 81
6.20 Çoklu engel senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı 82
6.21 Çoklu engel senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı 83
6.22 $\alpha = 0.9$, $\beta = 0.05$ ve $\gamma = 0.05$ için gerçekleştirilen benzetimde hesaplanan minimum uzaklıklar 84
6.23 Hareketli engel (gövde) senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı 85
6.24 Hareketli engel (gövde) senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülü (gövde C_1') arasındaki minimum uzaklık hesaplamaları 85
6.25 Hareketli engel (gövde) senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı 86
6.26 Hareketli engel (gövde) senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı 87
6.27 Önerilen yöntem sadece yarım küre yüzeyinde örnekleme yapılarak kullanıldığında benzetim sonuçları: (a) Simüle edilen yörünge, (b) Robot ucunun konumu, (c) Minimum uzaklık hesaplamaları 88
6.28 Hareketli engel (sol alt kol) senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı 89
6.29 Hareketli engel (sol alt kol) senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülü (gövde C_3') arasındaki minimum uzaklık hesaplamaları 89
6.30 Hareketli engel (sol alt kol) senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı 90
6.31 Hareketli engel (sol alt kol) senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı 91
6.32 Ağırlık parametrelerinin analizi sırasında planlama şeması 92
6.33 Ağırlık parametrelerindeki değişime göre seçilen yol noktasının sıralaması: (a) hedefe en yakın yol noktaları arasındaki sıralama, (b) engellerden uzak bir konfigürasyona taşıyan yol noktaları arasındaki sıralama, (c) düzgün bir yol izlenmesine öncülük eden yol noktaları arasındaki sıralama 94
6.34 Örnek nokta sayısına göre ortalama hesaplama süreleri 95
6.35 Kapsül sayısına göre ortalama hesaplama süreleri 96

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

4.1 Kinect v2 kamera iç parametreleri 32

6.1 Kawasaki RS005L endüstriyel robot manipülatör için kapsül boyutları 58
6.2 İnsan için kapsül boyutları 60
6.3 Hesaplama sürelerinin özeti 63

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar	**Açıklama**
FCL	Flexible Collision Library.
HG	Hand Guiding.
HRI	İnsan-Robot Etkileşimi.
IMU	Atalet Ölçüm Birimi.
PFL	Power and Force Limiting.
SMS	Safety-rated Monitored Stop.
SSM	Speed and Separation Monitoring.

1. GİRİŞ VE AMAÇ

Son zamanlarda robotların hem sosyal, hem de endüstriyel alanlardaki kullanımında önemli bir artış yaşanmaktadır. Özellikle endüstriyel robotlar, üretim ve imalat süreçlerinde sağladıkları yüksek verimlilik nedeniyle bu artışta başı çekmektedir. ISO 8373:2012 standardında da ifade edildiği üzere; ”endüstriyel robotlar, endüstriyel otomasyon uygulamaları için yere sabit veya mobil olarak kullanılabilen, üç veya daha fazla eksende programlanabilen, otomatik olarak kontrol edilebilen, yeniden programlanabilir çok amaçlı manipülatörlerdir” (ISO, 2012). Sahip oldukları mekanik yapıya göre farklı türlere ayrılan endüstriyel robotlar arasında kullanımına en sık rastlanan tür, bu tez çalışmasına da konu olan eklemli robotlardır (Teulieres vd., 2019). Geleneksel endüstriyel robot olarak da ifade edilen bu manipülatörler otomotiv, elektronik, kimya, makine gibi farklı sektörlerde sıklıkla kullanılmakta ve kullanım sıklıkları önceki yıllara göre sürekli olarak artış göstermektedir (IFR, 2020a). Bu artışın sebepleri arasında, günümüzde küçük ve orta ölçekli sanayi kuruluşlarının dahi bu robotlara uygun maliyetle erişebilmesi ve robotların kolay kullanımları amacıyla gerçekleştirilen gelişmeler de gösterilebilir.

Endüstriyel robotlar, üretim ve imalat süreçlerinde aktif olarak yer almalarına rağmen faaliyetlerini çoğunlukla insanlardan izole edilmiş ortamlarda sürdürmektedir. Yakın geçmişe kadar robot manipülatörlerin güvenlik bariyerleri ardına veya kapalı kafesler içerisine yerleştirilerek insan ile etkileşimlerinin kısıtlanması buna örnek olarak gösterilebilir. Ancak son zamanlarda endüstriyel ortamlardaki robot popülasyonunun artışı, insan ile robotun etkileşimini kaçınılmaz hale getirmektedir. Bir diğer yandan Endüstri 4.0’da olduğu gibi akıllı üretim ile verimlilik ve üretkenlik artışına odaklanan gelişmeler de bu durumun değişmesine ön ayak olmaktadır. Benzer şekilde, özellikle algılayıcı teknolojisinde yaşanan gelişmeler de bu değişimi desteklemektedir. Tüm bunlar göz önünde bulundurulduğunda, artık robotların insan ile aynı çalışma ortamını paylaşabileceği bir düzene ihtiyaç duyulmaktadır. İnsanın ve robotun aynı çalışma ortamını paylaşacağı bir düzen ise beraberinde etkileşimi getirmektedir. İnsan ile endüstriyel robot arasındaki etkileşim, aynı çalışma ortamını paylaşma ile sınırlı kalabileceği gibi iş birliği veya iş paylaşımı halini de alabilmektedir. Bu türden bir iş birliğinde, robotların tekrarlı veya tehlikeli endüstriyel görevleri yüksek hassasiyetle yerine getirebilmesi önemli rol oynamaktadır. İnsan ise özellikle karmaşık görevler söz konusu olduğunda sahip olduğu algı, esneklik ve zeka yetenekleriyle bu iş birliğine katkı sağlamaktadır. Her koşulda, etkileşimin türünden bağımsız olarak, insan ile endüstriyel robotun aynı çalışma ortamını paylaşması beraberinde emniyet sorununu ortaya çıkarmaktadır.

Sürdürülebilir ve başarılı bir insan-robot etkileşimi (İng. human-robot interaction) (HRI) için kullanılabilirlik, esneklik ve verimlilik her ne kadar önemli olsa da, literatürde de işaret edildiği gibi sağlanması gereken temel faktör emniyet olarak kabul edilmektedir (Maurtua vd., 2017b). Bu tez çalışmasında, emniyetli insan-robot etkileşimini sağlamaya yönelik ilk olarak minimum uzaklık hesabına dayanan bir sistem önerilmektedir. Önerilen sisteme göre, insan ile robot arasındaki minimum uzaklık hesaplanarak birbirleri arasında belirli bir ayırma mesafesinin (separation distance) korunması amaçlanmaktadır. Ayırma mesafesi, bazı iş birlikçi uygulamalarda insan ile robot arasındaki koruyucu mesafeyi ifade etmek için kullanılmaktadır (Villani vd., 2018; Matthias vd., 2019). Bu mesafenin korunması, hem robotun hem de insanın çalışma ortamında emniyetli bir şekilde ve birlikte faaliyet gösterebilmesine yardımcı olmaktadır.

İnsan ve endüstriyel robot etkileşiminin söz konusu olduğu çalışma ortamları doğaları gereği değişken ortamlardır. Dolayısıyla birçok iş birlikçi operasyonda olduğu gibi emniyet için insan ve robot arasındaki uzaklığın sürekli olarak ölçülmesi/hesaplanması ve takip edilmesi gerekmektedir. Emniyet zafiyeti oluşmaması için de bu uzaklık hesabının olabildiğince hızlı bir şekilde gerçekleştirilmesi büyük önem arz etmektedir. Tez çalışmasında sahip olunan bu motivasyonla gerçek zamanlı emniyetli etkileşim uygulamaları için kullanılabilecek bir sistem tasarımı hedeflenmektedir. Önerilen sistemde, insanın temsili için RGB-D algılayıcıdan alınan iskelet verisi kullanılmaktadır. Robot temsili için gerekli olan robot konfigürasyonu ise doğrudan denetleyiciden elde edilmektedir. Bu kapsamda, hesaplamanın hızlı bir şekilde gerçekleştirilebilmesi için ilk olarak insan ve robot geometrik ilkellerden biri olan kapsüllerle temsil edilmektedir. Kapsüller, basit yapıları ve matematiksel olarak kolayca ifade edilebilmeleri sayesinde uzaklık hesaplamalarında hız açısından önemli bir avantaj sağlamaktadır. Kapsül geometrisi, aynı zamanda insan ve robot uzuvlarının sahip olduğu geometriyle de benzerlik taşımaktadır. Benzer şekilde, hesaplamanın gerçek zamanlı uygulamalar için hızlı olması gerektiğinden dolayı uzaklık hesaplamaları için de Gilbert–Johnson–Keerthi (GJK) uzaklık algoritması kullanılmaktadır (Gilbert ve Foo, 1990). Bu algoritma, iki dışbükey nesne arasındaki Öklid mesafesini hesaplamak veya çarpışma tespiti yapmak için literatürde sıklıkla tercih edilen popüler bir algoritmadır. Geometrik ilkellerden olan kapsüller de sahip oldukları dışbükey geometri sayesinde bu algoritmaya uyum sağlamaktadır. Bu tez çalışmasında önerilen sistem bu haliyle literatürde sıklıkla tercih edilen bu iki yöntemi (temsil için geometrik ilkeller ve uzaklık hesabı için GJK algoritması) bir araya getirmektedir. Bu sayede hem hızlı ve basit, hem de özellikle insanın her bir vücut kısmının doğru şekilde dikkate alındığı bir sistem tasarımı önerilmektedir. Önerilen sistem tasarımı, hem benzetim ortamındaki, hem de gerçek ortamdaki bir robot kullanılarak test edilmektedir. Deneysel çalışmalar sonucunda ortaya konulan detaylı performans analizleri de bu yönüyle literatüre katkı sağlamaktadır.

Emniyetli insan-robot etkileşiminin bir başka önemli aşamasını da robot denetimi oluşturmaktadır. Öyle ki, etkileşim sırasındaki robot hareketinin çarpışmalardan kaçınacak nitelikte olması gerekmektedir. Buna bağlı olarak bu tezde gerçekleştirilmesi hedeflenen diğer amaç ise robot için çarpışmasız bir yol planı oluşturmayı içermektedir. Bunun için optimizasyon temelli yeni bir yaklaşım önerilmektedir. Önerilen yaklaşımda, stokastik ve iteratif bir yol planlama stratejisi takip edilmektedir. Yönteme göre, başlangıçta robotun önceden oluşturulmuş bir küresel yol planını takip ettiği ve hedef pozisyonunun bilindiği varsayılmaktadır. Robot ile insan arasındaki hesaplanan minimum uzaklığa göre, bu küresel plan terkedilmekte ve yerel planlayıcıya geçiş yapılmaktadır. Yerel planlayıcı, bu aşamada endüstriyel robotun takip edeceği yolu planlamak için kullanılmaktadır. Yerel planlayıcının takip ettiği optimizasyon temelli bu yöntem sayesinde, robot uç noktasının çarpışmasız ve düzgün bir yol izleyerek hedefe ulaştırılması amaçlanmaktadır. Önerilen yönteme benzer yaklaşımlar literatürde bulunsa da, robot uç noktasının takip edeceği olası pozisyonların örneklenmesinde ve bunların seçilmesinde izlenen strateji bu yöntemi benzerlerinden ayırmaktadır. Tez kapsamında önerilen yeni yol planlama yaklaşımı, benzetim ortamında farklı senaryolar için test edilmektedir.

Sonuç olarak bu tez çalışmasının ana başlıkları aşağıdaki gibi özetlenebilir:

- İnsan-robot etkileşim ortamında endüstriyel robot manipülatörlerin kontrol komutlarını üretmek için kullanılabilecek çarpışmasız bir çevrim içi yol planlama yöntemi önerilmiştir.
- Önerilen algoritma, robotun dağınık bir ortamda engellerden kaçınmasını ve uç noktasının ise düzgün bir yol izleyerek hedefine ulaşmasını sağlamaktadır.
- Çarpışmayı önlemek için GJK algoritmasının minimum uzaklık hesaplamasında, kapsüllerin ise insan ve robot temsilinde kullanıldığı bir sistem tasarımı önerilmiştir.
- Geleneksel endüstriyel robotlar kullanılarak benzetimler ve gerçek deneyler yapılmış ve hem sistem tasarımının, hem de yol planlama yaklaşımının farklı senaryolar kullanılarak doğrulaması yapılmıştır. Bunun yanında hem önerilen sistemin ve yöntemin performansı, hem de optimizasyon probleminde kullanılan parametreler hakkında detaylı bir tartışma yapılmıştır.

Bu tezin geri kalan kısmı altı bölümden oluşmaktadır. Bölüm 2'de insan-robot etkileşimi, emniyetli çalışma ortamının algılanması ve çarpışmasız robot hareketi üzerine gerçekleştirilen literatür araştırması verilmektedir. Bölüm 3'te insan-robot etkileşiminde sıklıkla karşılaşılan ve bu tezde de sıkça bahsedilen kavramlar açıklanmaktadır. Bölüm 4'te

emniyetli çalışma ortamının algılanmasına ve modellenmesine dair aşamalardan bahsedilmektedir. Öncelikle geometrik ilkeller ve kapsül geometrisi açıklanmakta, devamında ise insan ve robot temsili tüm detaylarıyla anlatılmaktadır. Bölüm 5'te emniyetli insan-endüstriyel robot etkileşimi için önerilen yöntemler açıklanmaktadır. İlk olarak minimum uzaklık algoritmasının ve önerilen sistem tasarımının anlatıldığı bu bölümün devamında, yeni yol planlama yaklaşımı tüm aşamalarıyla birlikte açıklanmaktadır. Bölüm 6'da gerçekleştirilen deneysel çalışmalar açıklanmakta ve elde edilen bulgular yine bu bölümde değerlendirilmektedir. Bölüm 7'de ise tüm sistem ve gerçekleştirilen deneysel çalışmalar sırasında elde edilen sonuçlar özetlenerek tez sonuçlandırılmaktadır. Gelecekteki çalışmalar için kullanılabilecek öneriler de yine bu bölümde verilmektedir.

2. LİTERATÜR ARAŞTIRMASI

2.1 İnsan-Robot Etkileşimi ve Emniyet

Son zamanlarda, endüstriyel robot manipülatörleri, kolay erişilebilmesi ve verimlilikleri nedeniyle endüstride giderek artan bir şekilde kullanılmaya başlamıştır (IFR, 2020a). Bu durum, endüstriyel alanlarda insanların ve endüstriyel robotların daha fazla alanı paylaşmasına ve kaçınılmaz bir şekilde birbirleriyle etkileşime girmesine neden olmaktadır. Aynı çalışma ortamını paylaşma, birlikte çalışma veya iş birliği gerektiren endüstriyel uygulamalar, artık robot ve insan arasına sınır koymak için tercih edilen güvenlik bariyerlerinin veya kapalı kafeslerin kullanımını da ortadan kaldırmaktadır (Magrini vd., 2020). Bu sayede insan zekâsı ve esnekliği, robotun hassasiyet ve doğruluk yetenekleriyle birleşmektedir. Bu birleşim aynı zamanda, özellikle algılayıcılar için yaşanan teknolojik gelişmelerle desteklenmektedir.

İnsan-robot etkileşiminin günümüz endüstrisinde bu seviyelere ulaşmış olması araştırmacıların da ilgisini çekmektedir (Tsarouchi vd., 2016). İnsan ve robot arasındaki etkileşimin türü, HRI alanında ele alınan önemli noktalardan birisidir. İnsan-robotun aynı ortamı paylaşması, insan-robotun koordineli çalışması ve insan-robot işbirliği, HRI'yı üç kategoriye ayırmak için kullanılmaktadır (Hentout vd., 2019). Bu sınıflandırma, robot ve insan tarafından paylaşılan çalışma alanı, çalışma zamanı, amaç ve etkileşim koşullarına dayanmaktadır. Tüm bu tanımlamalar arasında ve etkileşimin türüne bakılmaksızın birçok çalışma ve inceleme, emniyet sorununu insan-robot etkileşiminin ana problemi olarak ele almaktadır (Cherubini vd., 2016; Bdiwi vd., 2017; Maurtua vd., 2017b; Robla-Gómez vd., 2017; Villani vd., 2018; Karacığan, 2021). Geçtiğimiz yıllarda, insanı robot çalışma alanından uzak tutan veya robotun çalışmasını insanların varlığında durduran önlemler artık değişim geçirmekte ve farklı biçimler almaktadır. Özellikle algılayıcı teknolojisindeki son gelişmeler ve emniyeti sağlamak için öne sürülen yeni tasarımlar bu değişime katkıda bulunmaktadır (Ajoudani vd., 2018).

Endüstriyel robotların insanlarla etkileşimiyle ilgili olan emniyet düzenlemeleri, birçok çalışmada da belirtildiği gibi, genellikle ISO 10218 gibi standartlar tarafından belirlenmektedir (Robla-Gómez vd., 2017; Villani vd., 2018; Dağlı, 2019; Magrini vd., 2020). Özellikle yakın zamanda ilan edilen teknik şartname ISO / TS 15066, iş birliğine dayalı endüstriyel operasyonlar için yeni düzenlemeler getirmektedir. Mevcut ISO 10218-1/2 standartlarını tamamlayan ve ek yönlendirmeler sağlayan bu şartname, aynı

zamanda ”emniyet dereceli gözlenen durdurma - safety-rated monitored stop (SMS)”, ”elle yönlendirme - hand guiding (HG)”, ”hız ve ayırma gözlemi - speed and separation monitoring (SSM)” ve “güç ve kuvvet sınırlama - power and force limiting (PFL)” olmak üzere dört iş birliği (ortak çalışma) biçimi tanımlamaktadır (Bdiwi vd., 2017; Villani vd., 2018; Magrini vd., 2020). Bu çalışma biçimlerinin, yakın zamanda geliştirilen iş birlikçi robotlarla (İng. cobot) uygulanması nispeten kolay olsa da, geleneksel endüstriyel robotlar, algılama sistemleri ve yeni kontrol stratejileri gibi ek önlemler ve yöntemler gerektirmektedir (Dilibal ve Şahin, 2018). Hem iş birlikçi robotlar, hem de geleneksel endüstriyel robotlar için bu konuda emniyeti garanti etmek adına farklı stratejiler ve tasarımlar geliştirilmektedir. Avrupa Birliği (AB) tarafından finanse edilen FourByThree projesi, bu alanda farklı emniyet önlemlerinin geliştirildiği güncel çalışmalar arasında yer almaktadır (Maurtua vd., 2017b). Proje dahilinde, çok modlu etkileşim için anlamsal (semantic) bir yaklaşım geliştirilmiş ve yeni hareket planlama ve programlama yöntemleri sunmak gibi çeşitli araştırma çalışmaları yürütülmüştür (Maurtua vd., 2017a; Pellegrinelli vd., 2017). Başka bir AB projesi olan SYMBIOTIC, simbiyotik yapılarında aktif çarpışmadan kaçınma ve uyarlanabilir robot kontrolünü birleştirerek emniyetli bir çalışma ortamı için alternatif çözümler getirmektedir (Wang vd., 2017a). ROBO-PARTNER projesi de benzer şekilde insanların ve robotların ortak bir çalışma ortamına entegrasyonuna odaklanmaktadır. Bunu başarmak için sezgisel arayüzler, verimli programlama yöntemleri ve esnek entegrasyon ve iletişim mimarisi dahil olmak üzere bir hibrit üretim paradigması izlenmektedir (Michalos vd., 2014). Bingol ve Aydogmus (2019) tarafından geliştirilen güvenli insan-robot etkileşim yazılımı GİREY de, evrişimli sinir ağı tabanlı bir yazılım olarak çalışma ortamındaki insanın emniyetini çok kanallı iletişim yoluyla sağlamaktadır.

Literatürde insan-robot etkileşiminde emniyet üzerine gerçekleştirilen birçok çalışma, çalışma alanı izleme ve çarpışmadan kaçınma gibi tipik uygulamaları içermektedir (Schmidt ve Wang, 2014; Papanastasiou vd., 2019; Cherubini vd., 2019). Robot çalışma alanında insan konumunu ve duruşunu izlemek, özellikle SMS ve SSM tipi iş birliği çeşitlerinde çarpışmadan kaçınmak önemlidir. Öyle ki, SMS çalışma tipinde robot çalışma alanında operatör varlığı gözlendiğinde durmakta veya duraklamaktadır. SSM çalışma tipinde ise bir operatör robota minimum emniyet mesafesinden daha fazla yaklaştığında robot yavaşlamakta veya duraklamaktadır. Bu bağlamda, bu tür iş birlikçi operasyonlar için harici algılama sistemleri ve aktif çarpışmadan kaçınma yöntemlerine ihtiyaç duyulmaktadır ve minimum emniyet mesafesi, emniyetli bir SSM uygulaması gerçekleştirmek için temel ölçütlerden biridir (Marvel ve Norcross, 2017). Literatürdeki birçok çalışmada görüldüğü gibi, kontrol ve planlama stratejileri robot ile insan arasındaki emniyetli mesafeye dayanmaktadır. Lasota vd. (2014), bunun için emniyetli insan-robot etkileşimine izin veren gerçek zamanlı bir emniyet sistemi önermiştir. Emniyet sistemine göre, çalışma alanındaki insan konumunu izlemek için bir hareket yakalama sistemi

kullanılmakta ve insan ile robot arasındaki mesafeyi ölçmek için ise robot denetleyicisinden robot konfigürasyonu alınmaktadır. Başka bir çalışmada, operatörün robotun tehlike bölgesinden uzaklığı bir RGB-D algılayıcı kullanılarak tespit edilmektedir (Salmi vd., 2016). Operatörün konumu, hızına ve yönüne göre tahmin edilmekte ve robotun hızı buna göre ayarlanmaktadır. Bunun yanında, ilk sistem arızalandığında etkinleştirilen ve güvenli lazer tarayıcıdan oluşan ikincil bir sisteme de tasarımda yer verilmektedir. Benzer bir çalışmada, çoklu RGB-D algılayıcıları kullanılarak derinlik bilgisi elde edilmekte ve robot ile engeller arasındaki en yakın mesafe belirlenmektedir (Mohammed vd., 2017). Long vd. (2018) tarafından gerçekleştirilen çalışmada ise üç adet lazer algılayıcı kullanılan 3-B dinamik emniyet bölgesi uygulamasına yer verilmektedir. Önerilen yaklaşımda lazer algılayıcı yardımıyla algılama düzlemleri oluşturulmakta ve operatörün bu düzlemlerden herhangi biriyle kesişip kesişmediği tespit edilmektedir. Bu türden bir tespit yapılması durumunda, mesafe bilgisi operasyon tipini belirlemek için kullanılmaktadır. Duman ve Düven (2019) tarafından geliştirilen güvenlik sistemi, görüntü işleme yöntemleriyle insan/nesne tespiti ile konum ve mesafe çıkarımına dayanmaktadır. Bunun için, robotun uç ekleminin üzerine kameralar yerleştirilmekte ve elde edilen görüntüler işlenerek robotun çalışma alanına dair çıkarımlar yapılmaktadır.

Emniyetli insan-robot etkileşimi üzerine literatürde yer alan çalışmalar incelendiğinde, çalışma alanının izlenmesinin ve çarpışmadan kaçınmanın gerçekleştirilmesinin iki önemli başlık olduğu görülmektedir. Bu tez çalışmasında da olduğu gibi, hemen hemen bütün çalışmalarda emniyet için bu iki unsur üzerinde durulmaktadır.

2.2 Emniyetli Çalışma Ortamının Algılanması

Yapılan araştırmalara göre robot ile çevresindeki nesneler arasındaki uzaklığın hesaplanmasının, emniyetli etkileşim yöntemlerinin geliştirilmesinde ve çalışma ortamının algılanmasında önemli bir rol oynadığı sonucuna varılabilir. Hesaplama için robotun ve çevresindeki nesnelerin konumunun ve duruşunun bilinmesi gerektiği de açık bir şekilde görülmektedir. Robotun konfigürasyonu, günümüz robotik sistemlerinde çoğunlukla robot denetleyicisi aracılığıyla gerçek zamanlı olarak elde edilebilmektedir. Bununla birlikte, robot çalışma ortamındaki nesneler özellikle de insanlar için farklı işlemlerin gerçekleştirilmesi gerekmektedir.

Literatürdeki yaklaşımların bazılarında, insanın algılanmasına ait veriler veya robot konfigürasyonu doğrudan kullanılmaktadır (Lasota vd., 2014; Salmi vd., 2016). Bazı çalışmalarda ise bu veriler, uzaklık hesaplamalarında geometrik temsiller veya

basitleştirmeler gerçekleştirmek için kullanılmaktadır (Kong ve Yu, 2014). Literatürde robot ve insan geometrisini temsil ederken farklı türden temsiller kullanılmaktadır (Aslan, 2018). Temsil türü, robot hareket planlaması sırasında çarpışmadan kaçınma ve uzaklık hesaplama gibi işlemlerin performansını zaman, bellek kullanımı ve doğruluk açısından etkilemektedir. Safeea vd. tarafından gerçekleştirilen çalışmada, etkileşimli bir ortamda yer alan insan ve robot kapsül şeklinde geometrik ilkellerle temsil edilmektedir (Safeea ve Neto, 2019). Atalet ölçüm birimlerinden (IMU) ve bir lazer tarayıcıdan elde edilen verilere göre duruşları belirlenen dört kapsül, sağ ve sol kolların alt ve üst kısımlarını temsil ederken, vücudu başa kadar temsil etmek için ise bir başka kapsül kullanılmaktadır. Çalışmada bahsedilen iş birlikçi robotun temsili için ise üç kapsül kullanılmakta ve bu kapsüllerin duruşu, robotun eklem açılarına göre belirlenmektedir. Önerilen yaklaşımda gövde kapsülü insan başını da kapsadığından, baş için gövdenin duruşu esas alınmaktadır. Böyle bir gösterim, zaman ve bellek tüketimi ve hesaplama hızı açısından avantajlı olabilir, ancak doğruluğu da azaltabilir. Bu çalışmaya benzer şekilde, farklı algılayıcı (manyetik izleme sensörleri, IMU'lar ve lazer tarayıcı) konfigürasyonları kullanılarak da endüstriyel ortamda çevrim içi çarpışmadan kaçınma yöntemleri gerçekleştirilmektedir (Safeea vd., 2019). Buna göre, robot ve insan vücudu kapsüller ile temsil edilmekte ve robot denetleyicisinden elde edilen eklem konfigürasyonlarına düz kinematik uygulanarak robotla ilgili olanların duruşları belirlenmektedir.

Literatürde geometrik ilkellerin yanı sıra 3-B nokta bulutu veya benzer yapıları içeren çalışmalar da bulunmaktadır. Bu çalışmalardan birinde, Kinect RGB-D algılayıcıları tarafından elde edilen derinlik ve iskelet verileri kullanılarak insan tespiti gerçekleştirilmektedir (Nikolakis vd., 2019). İnsana ait nokta bulutu, bulut noktalarının iskelet verileriyle eşlenmesiyle oluşturulmakta ve ardından oluşturulan nokta bulutu, insanın bütününü temsil eden dışbükey bir gövde oluşturmak için kullanılmaktadır. Buna benzer bir başka çalışmada ise insan koluna ait nokta bulutunu elde etmek için yine bir RGB-D algılayıcı kullanılmaktadır (Han vd., 2018). Nokta bulutu, önce 3-B octree haritaya dönüştürülmekte ve ardından bu harita mesafe hesaplama algoritmasında kullanılmak üzere dışbükey gövde (İng. convex hull) ile temsil edilmektedir. Bu gösterim, ham nokta bulutu verilerine kıyasla bellek kullanımını azaltabilir, ancak ek hesaplama süresine neden olmaktadır. Benzer şekilde, RGB-D algılayıcıdan elde edilen ortama ait nokta bulutu, 3-B octree haritasına dönüştürülüp robot hareket planlamasında da kullanılmaktadır (Brito vd., 2017). Ancak bu durumda insan çevreden ayrı olarak algılanmamaktadır. Bir başka çalışmada, iki RGB-D algılayıcıdan elde edilen nokta bulutunun, sınırlayıcı kutu (İng. bounding box) ile bir bütünü tanımlamak için kullanıldığı da görülmektedir (Mohammed vd., 2017). Bu yöntem, çarpışma tespiti için gerekli hesaplamaları hızlandırmak için kullanılsa da, doğruluğu önemli ölçüde etkilemektedir.

Algılayıcılardan elde edilen bilgilerin herhangi bir geometrik temsil veya model oluşturmadan doğrudan çarpışmadan kaçınma algoritmalarında kullanıldığı çalışmalar da literatürde yer almaktadır. Bu çalışmalardan birinde, insan koluna yerleştirilmiş işaretlerden oluşan bir hareket yakalama sistemi, bir zaman serisi modeliyle tekrarlayan insan hareketini tahmin etmek için kullanılmaktadır ve tut-ve-yerleştir uygulamasını gerçekleştiren bir insan kolunun verileri daha sonra kullanılmak üzere kaydedilmektedir (Wang vd., 2017b). Gerçek zamanlı gerçekleştirilen uygulamada, bir RGB-D algılayıcı kullanılarak insan iskeleti tanıma yöntemi uygulanmakta ve dirsek açılarını tahmin etmek için insan kol eklemlerinin konumları elde edilmektedir. Bu tahmin daha sonra robot ve insan kolu arasında emniyetli bir mesafe bırakarak çarpışmasız bir yörünge oluşturmak için kullanılmaktadır. Meziane vd. (2014) tarafından gerçekleştirilen çalışmada ise operatörün IMU algılayıcı taşıyan bir kask yardımıyla endüstriyel ortamda konumunun tespiti yapılmakta ve bu verilere göre robotun çarpışmadan kaçınması gerçekleştirilmektedir. Zhang vd. (2020) tarafından gerçekleştirilen çalışmada ise RGB-D algılayıcı ve IMU'ların verilerini doğrusal olmayan bir optimizasyon yöntemiyle birleştiren bir insan hareket yakalama sistemi önerilmektedir.

Bu çalışmalardan, emniyetli insan-robot etkileşiminde çarpışma tespiti ve mesafe hesaplaması için insan vücudunun belirli kısımlarının çeşitli temsillerle hesaba katıldığı ve çeşitli algılayıcı konfigürasyonları tarafından algılandığı sonucuna varılabilir. Geometrik ilkellerin kullanıldığı temsilin, özellikle gerçek zamanlı uygulamalarda minimum uzaklık hesaplamaları için hızlı ve verimli bir çözüm sağladığına da dikkat çekilebilir (Safeea vd., 2017).

2.3 Minimum Uzaklık Hesaplaması

İnsan-robot etkileşimi için emniyetli bir gerçek zamanlı uygulama geliştirmenin ana zorluklarından biri, minimum uzaklığı daha önce de belirtildiği gibi hızlı ve verimli bir şekilde hesaplamaktır. Literatürde, minimum uzaklık hesaplamalarında kullanılmak üzere insan ve robotların geometrik temsili için birçok yaklaşım yer almaktadır. Ancak bu yaklaşımların çoğunda, minimum uzaklık hesaplaması için dışbükey çok yüzlü geometriler (İng. polyhedral) dikkate alınmaktadır. Han vd. (2018) tarafından belirtildiği gibi dışbükey modellere dair minimum uzaklık algoritmalarında Voronoi-bölge-tabanlı yöntemler ve simpleks-tabanlı yöntemler olmak üzere iki ana kategori yer almaktadır. Bu aşamada, Gilbert-Johnson-Keerthi (GJK) tabanlı algoritmalar gibi simpleks tabanlı yöntemlerin, Lin-Canny (LC) algoritması gibi Voronoi tabanlı yöntemlerden daha sağlam ve daha basit bir bellek yapısına sahip olduğu da belirtilmektedir. Hızlı ve uygulaması kolay yapıları sayesinde GJK tabanlı algoritmalar özellikle uzaklık hesaplama ve çarpışma önleme

uygulamalarında sıklıkla kullanılmaktadır. Bernabeu ve Tornero (2002) tarafından gerçekleştirilen çalışmada da görüldüğü gibi, yeni çarpışma tespit teknikleri geliştirmek için yapılan bazı girişimler de literatürde yer almaktadır. Önerilen yöntem, iki dışbükey-politop (İng. convex-polytope) modeli arasındaki minimum uzaklığın hesaplanmasına bağlıdır ve uzaklık algoritması GJK uzaklık algoritmasına dayanmaktadır. Yakın tarihli bir başka çalışmada ise uzaklık hesaplamasına ve soyut algılamaya dayalı dinamik bir engelden kaçınma yöntemi önerilmekte ve iki 7-DOF bir endüstriyel robot modeli kullanılarak bir simülasyon ortamında test edilmektedir (Han vd., 2018). Önerilen yöntem için, RGB-D algılayıcıdan elde edilen insan koluna ait nokta bulutu önce 3-B octree haritasına dönüştürülmekte ve daha sonra octree haritanın dışbükey gövdesini hesaplamak için kullanılmaktadır. GJK algoritmasına dayalı minimum uzaklık hesaplaması, en yakın uzaklığın emniyetli değere ulaşıp ulaşmadığını kontrol etmek için robot modeli ve insan kolunun dışbükey gövdesi kullanılarak gerçekleştirilmektedir. Başka bir çalışmada ise çoklu derinlik kameraları kullanılarak tespit edilen nesneler dışbükey gövde olarak temsil edilmekte ve minimum mesafe hesaplamaları için GJK algoritması kullanılmaktadır (Fischer ve Henrich, 2009). Tüm bu çalışmaların ortak özelliği, dinamik çarpışmadan kaçınma yöntemlerini uygulamak için hızlı ve basit bir uzaklık hesaplama algoritmasına ihtiyaç duymalarıdır.

Bu sebeplerden, bu tez çalışmasında uzaklık hesaplamasını hızlı bir şekilde yapabilmek için;

- insanın ve robotun geometrik ilkellerle temsilinin ve
- hızlı bir uzaklık hesaplama algoritmasının

bir araya geldiği bir sistem tasarımı önerilmektedir.

2.4 Çarpışmasız Robot Hareketi ve Robot Denetimi

Gerçek zamanlı çarpışmadan kaçınma gerçekleştirecek bir yaklaşım, ortamın algılanması, çarpışmadan kaçınma algoritması ve robot denetimi olmak üzere üç unsurdan oluşmaktadır (Flacco vd., 2012). İnsan-endüstriyel robot etkileşiminde ortamın algılanmasından sonra emniyet için önemli problemlerden bir diğeri, endüstriyel robot manipülatörlerinin tanımlanan görevi yerine getirirken çevredeki engellerle, özellikle de insanlarla, çarpışmaktan kaçınması gerektiğidir. Bu gibi durumlarda, mevcut görevi, yani görev kısıtlı hareketi gerçekleştirirken robotun çarpışmasız bir yol takip edebilmesi için bir planlayıcıya ihtiyaç duyulmaktadır. Gereğinden çok serbestlik dereceli (İng. redundant)

robot manipülatörlerin engellerden kaçınma gibi ek hedefleri takip ederken bu görevleri yerine getirebilecek kabiliyete, gereğinden çok serbestlik dereceleri sebebiyle sahip oldukları hali hazırda bilinmektedir (Unver, 2004; Qiu ve Cao, 2018).

Literatürdeki mevcut planlayıcılar, Kartezyen uzay tabanlı yöntemler ve eklem uzayı tabanlı yöntemler olarak ikiye ayrılabilir (Gao vd., 2015). Eklem uzayı tabanlı yöntemlerde, eklem konfigürasyonu zamanın bir fonksiyonu olarak ifade edilmekte ve başlangıç ile hedef konfigürasyonları arasındaki kısıtlar dikkate alınarak planlama yapılmaktadır. Kartezyen uzay tabanlı yöntemlerde, planlama yapılırken robotun uç noktasının konumu ve yönelimi dikkate alınmakta ve ters kinematik hesaplamaları ile eklem konfigürasyonu elde edilmektedir (Boyacıoğlu, 2016). Endüstriyel uygulamalarda asıl görevi robotun uç noktasının gerçekleştirdiği düşünüldüğünde, robotun verilen görevi yerine getirebilmesi için eklem uzayındaki planlama tek başına yeterli olmayabilir (Gao vd., 2015). Bu nedenle, robotun çarpışma içermeyen görev kısıtlı hareketini gerçekleştirebilmesi için Kartezyen uzay tabanlı yaklaşımlara da ihtiyaç duyulmaktadır.

Çarpışmasız yol planlama probleminin çözümü için literatürde izlenen stratejiler de küresel ve yerel stratejiler olarak ikiye ayrılabilir (Wang vd., 2018). Küresel yol planlaması üzerine yapılan yoğun çalışmalar sonucunda günümüze kadar, C-Space yöntemi, rastgele ağaçların hızla keşfedilmesi, olasılıksal yol haritaları, potansiyel alan yöntemi ve grafik-tabanlı arama algoritmaları gibi birçok öncü yaklaşım önerilmiştir (Lozano-Perez, 1990; LaValle, 2006; Khatib, 1986; Stentz, 1997). Çözüm mümkünse, bu yaklaşımlar kullanılarak ilk konfigürasyondan hedef konfigürasyona çarpışmasız bir yol oluşturulabilir. Ancak bu yöntemler, hesaplama yükleri, tam yörünge bilgisine ve çevrenin tam temsiline ihtiyaç duymaları ve çoğunlukla statik ortamlar için uygun olmaları nedeniyle gerçek zamanlı kontrol için genellikle tercih edilmemektedir (Wang vd., 2018; Zavlangas ve Tzafestas, 2000). Bu nedenle, dinamik ortamlarda çarpışmadan kaçınma probleminin çözümü için yerel stratejiler geliştirilmektedir. Yerel yöntemler genellikle sensörlerden elde edilen geri besleme bilgisine dayalı olarak adım adım yörünge değişikliği gerçekleştirmekte ve mevcut robot konfigürasyonunun Jacobian matrisini kullanmaktadır. Bu yöntemler, genellikle kinematik kontrole ve kinematik artıklığın (İng. redundancy) çözümlenmesine dayanmaktadır (Siciliano, 1990). Bu sebeple, literatürdeki mevcut çalışmaların önemli bir kısmı Jacobian tabanlı yöntemlerin ve gradyan projeksiyonu gibi boşluk uzayı (İng. null space) tabanlı yöntemlerin kullanımını içermektedir (Lee ve Buss, 2007; Kelemen vd., 2018; Lillo vd., 2019; Liu vd., 2020).

Daha önce belirtilen çalışmalarda uygulanan çarpışmadan kaçınma stratejileri, çoğunlukla yapay potansiyel alanlara dayanmaktadır. Yapay potansiyel alan tabanlı stratejilerde, vektör alanlarını tanımlamak için hedef ile robotun uç noktası arasındaki

uzaklık ve robotun uzuvları ile engeller arasındaki uzaklık kullanılmaktadır. Buna göre de robot için çekici ve itici kontrol komutları üretilmektedir. Birçok yeni çalışma, çarpışmadan kaçınma için potansiyel alan tabanlı stratejiler önermektedir (Long vd., 2018; Wang vd., 2018; Flacco vd., 2012; Kivelä vd., 2018; Safeea vd., 2020). Potansiyel alan tabanlı yöntemler için en çok bilinen sorunlardan biri, hedeften kaynaklanan çekici kuvvet ile engellerden gelen itici kuvvetlerin birbirini zıt yönlerde etkilemesi ve bu nedenle de engellerden kaçınma etkisinin azalması veya robotun yerel minimumda takılması olarak ifade edilebilir. Diğer bir dezavantaj ise potansiyel alanın robot manipülatörü ile insan arasındaki bağıl hareketi yakalayamaması olarak ifade edilmektedir (Tsai, 2014). Ayrıca potansiyel alan tabanlı stratejilerin çarpışmadan kaçınma açısından reaktif yaklaşımlar olduğunu söylemek mümkündür (Cefalo vd., 2018). Öyle ki, bu yaklaşımlarda engellere yapay potansiyel alanlar sayesinde tepki verilirken, kısa vadede bile olsa herhangi türden bir öngörü veya yeniden planlama içermezler. Bu nedenle, bu tez çalışmasında çarpışmadan kaçınma için yeniden planlamaya izin veren ve robot ucunu hedef noktasına taşıyan optimizasyon tabanlı bir yöntem önerilmektedir.

Potansiyel alan temelli yaklaşımların dışında, optimizasyon temelli yaklaşımlar da çarpışmadan kaçınma probleminin çözümünde kullanılmaktadır. Yakın zamanda çarpışmasız yol planlaması için birçok optimizasyon tabanlı strateji önerilmiştir (Lin vd., 2016; Mao ve Xiao, 2019; Khan vd., 2019). Optimizasyon temelli yaklaşımlarda, çarpışmasız bir yörünge belirlemek için robotun mevcut durumu ve engellerin konumu esas alınarak optimizasyon problemi çözülmektedir. Bu yöntem kullanılarak gerçekleştirilen çalışmaların birçoğunda, genellikle küresel bir hareket planlayıcı yerel bir reaktif denetleyici ile birleştirilmekte ve yeniden planlama yapılmaktadır (Cefalo vd., 2018). Lin vd. (2016), insan güdümlü programlama senaryoları için çevrimiçi hız tabanlı bir çarpışmadan kaçınma algoritması önermektedir. Bu çalışmada çarpışmadan kaçınma problemi bir optimizasyon problemi olarak formüle edilmekte ve maliyet fonksiyonu insanın istenen hareketi ile uygulanan komut arasındaki hız farkından oluşmaktadır. Robot komutunun, kontrol noktasındaki itici vektöre eşdeğer bir hız üretmesini sağlamak için bir eşitlik kısıtı kullanılmaktadır. Ancak yöntem, SMS ve SSM işbirliği formları için uygun olmamakla birlikte, sadece insan güdümlü programlama senaryoları ile sınırlandırılmaktadır. Mao ve Xiao (2019) tarafından, genel RAMP yaklaşımında görev kısıtlamalarını içeren bir algoritma önerilmektedir. Bu yöntem sayesinde, çoklu optimizasyon kriterlerinden oluşan maliyet fonksiyonu kullanılarak yörüngeler değiştirilmekte ve çarpışmasız bir hareket elde edilmektedir. Buna rağmen önerilen yöntem, anahtarlama (İng. switching) stratejisi nedeniyle dağınık ortamlar için uygulanabilir olmayabilir. Khan vd. (2019) tarafından ise takip kontrolü ve engellerden kaçınma problemi için tek kısıtlı optimizasyon sistemi önerilmektedir. Bu yaklaşıma göre, robotu engellere yaklaştıran eklem konfigürasyonlarını cezalandıran bir ceza terimi sayesinde

optimize edici engelden kaçındığı için ödüllendirilmektedir. Optimizasyonu gerçek zamanlı olarak uygulamak için bir böceğin davranışını taklit eden meta-sezgisel bir yaklaşım kullanılmaktadır. Algoritma, manipülatör-engel mesafesini maksimize eden eklem uzay yörüngeleri üretmektedir. Ancak, önerilen algoritmanın performansı tek bir statik engel kullanılarak değerlendirilmekte ve birden fazla engel için sergileyeceği performans bilinmemektedir. Çarpışmasız yol planlaması için başka bir optimizasyon tabanlı strateji ise Balan ve Bone (2006) tarafından önerilmektedir. Çarpışmadan kaçınma algoritması, bir dizi sabit arama yönü boyunca olası robot uç noktası hareketlerinin analizine dayanmaktadır ve ilgili yaklaşım bu haliyle bu tez çalışmasında önerilen yaklaşıma benzemektedir. Bununla birlikte, bu çalışmada arama yönleri sabittir ve üç boyutlu uzayda uzamsal (İng. spatial) olarak eşit dağıtılmaktadır. Bu da, robotun uç noktasını sürekli bir şekilde önceden belirlenmiş sabit yönlerde hareket etmeye zorlamaktadır. Engellerden kaçınırken robotu hedef konumuna hareket ettiren arama yönlerinden birini seçmek için ise optimizasyon gerçekleştirilmektedir. Robot ve insan, çalışmada kürelerle temsil edilmekte ve aralarındaki minimum uzaklık, maliyet fonksiyonunda bir ceza faktörünü hesaplamak için kullanılmaktadır. İnsan modelinde 46, robot modelinde 41 küre kullanılarak geometrik temsillerden faydalanılmaktadır. Uzaklık hesabı birkaç temel hesaplama ile ele alınsa da, küre sayısındaki artışın hesaplama maliyetini arttıracağı aşikârdır. Önerilen yaklaşımın doğrulaması, algoritmanın insan ve robot arasındaki çarpışmaları önlediğini gösteren Monte Carlo simülasyonları uygulanarak yapılmaktadır.

Bu tez çalışmasında, optimizasyon tabanlı bir yöntem olarak tanımlanabilecek çarpışmasız bir yol planlama stratejisi önerilmektedir. Önerilen strateji, robotu engellerin olduğu bir ortamda hedefe yönlendirmek için yeniden planlamaya ve bir navigasyon yöntemine dayanmaktadır. Navigasyon yöntemine göre, robotun uç noktasında varsayımsal (İng. hypothetical) bir küre (veya yarım küre) tanımlanmakta ve robotun izleyebileceği yol için küre (veya yarım küre) yüzeyinde bir nokta örneklemesi yapılmaktadır. Takip edilen yöntem için, küresel bir yol planının önceden çevrim dışı olarak oluşturulduğu ve robotun uç noktasının hedef pozisyonunun bilindiği varsayılmaktadır. İnsan ve robot arasındaki minimum uzaklık ise emniyetli insan-robot etkileşimi dikkate alınarak sürekli olarak hesaplanmaktadır. Hesaplanan uzaklık, emniyetli ayırma mesafesi eşiğinin altına düştüğünde, küresel yol planı terk edilmekte ve yerel planlayıcıya geçilmektedir. Önerilen çevrim içi yerel planlayıcı ise navigasyon yöntemini kullanarak ve bir optimizasyon problemini çözerek yol planı oluşturmaktadır. Optimizasyon problemi, ağırlıklı maliyet fonksiyonunda ceza terimleri kullanılarak kısıtlı bir minimizasyon problemi olarak formüle edilmektedir. Ağırlıklı maliyet fonksiyonu, her biri tek bir amaca yönelik üç terime sahiptir; robotun uç noktasını hedefe taşımak, robot ile insan arasında emniyetli mesafeyi korumak, robotun uç noktasının düzgün bir yol izlemesini sağlamak. Önerilen yol planlama yaklaşımı, hem takip edilen yol bulma yöntemi ile, hem de optimizasyon problemindeki

kriterler ile literatürdeki benzer yöntemlerden farklılık göstermektedir. Bu yönleriyle yeni bir yol planlama stratejisi olarak literatüre katkı sağlayacağı öngörülmektedir.

3. İNSAN-ROBOT ETKİLEŞİMİ

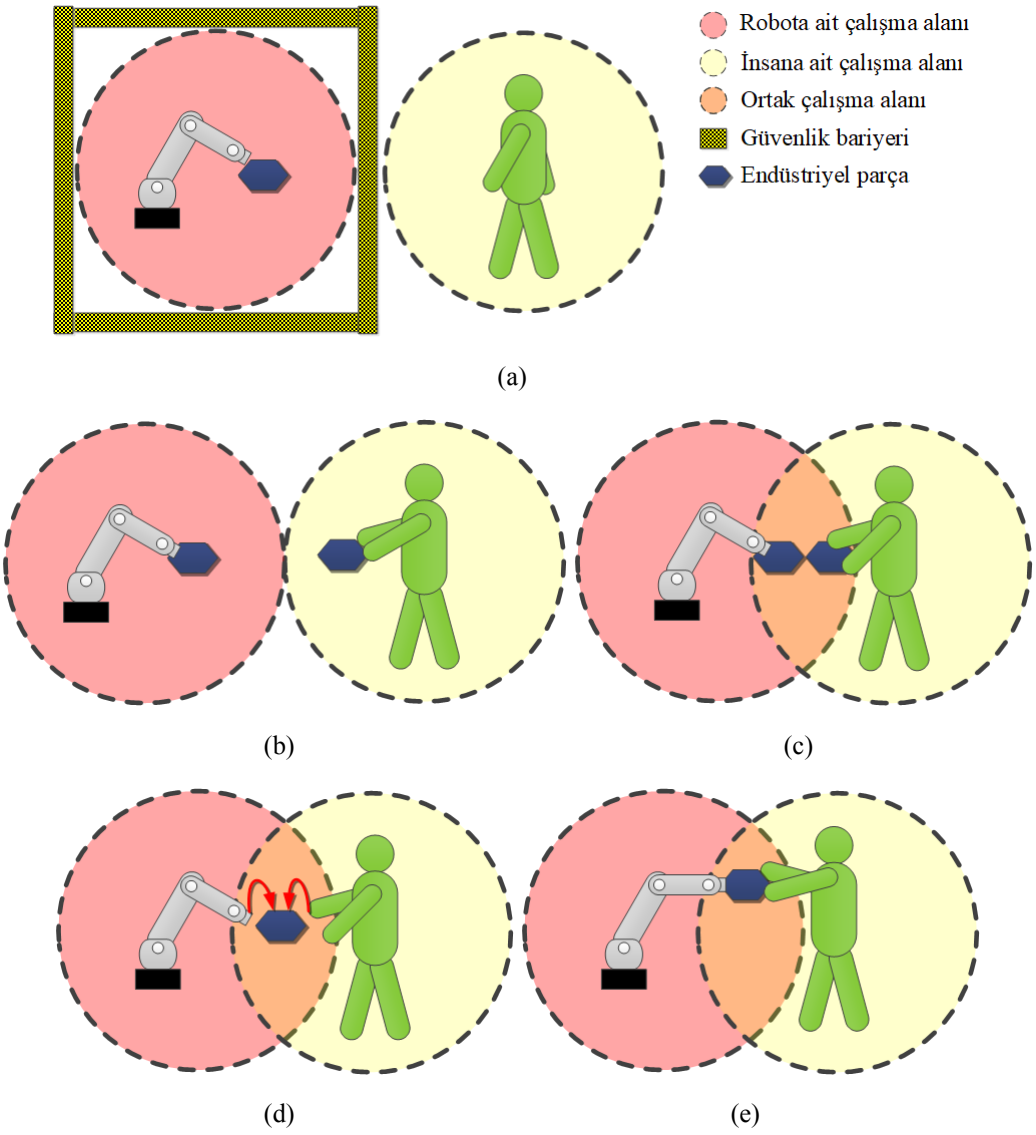
İnsan-robot etkileşimi, insanlar ile robotlar arasındaki her türden etkileşimin ve iletişimin incelendiği bir çalışma alanıdır. Bu çalışma alanının temel amacı, insan ile robot arasındaki etkileşimin niteliğini tanımlamak ve buna göre robotlara insanlarla etkileşime girebilecek şekilde yetkinlik kazandırmaktır. Bu etkileşim, bazen sadece insan ve robotun aynı ortamı paylaşması şeklinde olabileceği gibi, bazen de robotun ve insanın ortak bir görevi yerine getirmek için bir araya geldiği fiziksel bir iş birliği halini alabilmektedir. Bu durum da, çoğunlukla insan-robot iş birliği (ing. Human-Robot Collaboration) olarak adlandırılmakta ve diğer etkileşim türlerinde olduğu gibi yine insan-robot etkleşimi kapsamına girmektedir.

3.1 Etkileşim Türleri

İnsan ile robot arasındaki etkileşim, çeşitli faktörlere göre farklı türlere ayrılmaktadır. Literatürde Michalos vd. (2015), Behrens vd. (2015) ile Hentout vd. (2019) tarafından gerçekleştirilen çalışmalar incelendiğinde, etkileşim türü sınıflandırılırken genellikle çalışma alanı, çalışma zamanı, amaç ve temas şekli gibi çeşitli faktörlerin dikkate alındığı görülmektedir. Çalışma alanı, etkileşime konu olan ve hem insan, hem de robot tarafından kullanılan alanı ifade etmektedir. Bu alan, robot ve insan tarafından duruma göre kısmi paylaşımlı, tam paylaşımlı veya ayrı olarak kullanılmaktadır. Çalışma zamanı, robotun veya insanın çalışma alanı içerisinde bulunduğu veya faaliyet yürüttüğü zaman aralığını tanımlamak için kullanılmaktadır. Bu tanıma göre, robot ve insan tarafından yürütülen faaliyetler eş zamanlı veya ayrı zamanlı olarak da ifade edilmektedir. Amaç ise çalışma alanındaki bir unsurun yürüttüğü faaliyet kapsamında başarmaya çalıştığı hedefi tanımlamaktadır. Çalışma alanındaki unsurlar aynı amacı taşıyabileceği gibi, farklı amaçlar da taşıyabilmektedir. Temas şekli de genellikle robotun insanlarla ne denli bir temasa girebildiğini belirtmek için kullanılmaktadır. Çoğu zaman robotun insanlarla temas etmesi (özellikle kaza şeklinde) veya yakınlaşması istenmeyen bir durumdur. Ancak bazı iş birliği operasyon türleri için robotun insanla temas halinde olması da bilinçli bir şekilde istenmektedir. Söz konusu faktörler dikkate alındığında, etkileşime konu olan türlerin; bir arada bulunma (İng. coexistence), birlikte çalışma (İng. cooperation), sıralı çalışma (İng. synchronized) ve iş birliği (İng. collaboration) şeklinde sınıflandırıldığı görülmektedir (Behrens vd., 2015; Wilhelm vd., 2016; Aaltonen vd., 2018; Hentout vd., 2019).

Bir arada bulunma türünden etkileşim, insan ve robotun aynı çalışma ortamını ortak bir göreve sahip olmadan paylaşması anlamı taşımaktadır. Buradaki alan paylaşımı ifadesi, daha çok insanın ve robotun kendilerine özel olarak ayrılmış alanlarda faaliyet göstermesini ifade etmektedir. Ayrılmış alan ile güvenlik bariyerleri ve kapalı platformların yer aldığı hücreler kastedilmemektedir. Bu tarz bir etkileşimde, insan ve robotun aynı amacı taşıması da beklenmemekte ve yürüttükleri faaliyetler birbirinden bağımsız olmaktadır. Buna göre, etkileşimin sürdürülebilmesi için de genellikle çarpışmanın önlenmesine ihtiyaç duyulmaktadır. Robotun insanla herhangi bir şekilde temasa girmesi veya insana yakınlaşması arzu edilmemektedir. Bir başka etkileşim türü ise robot ve insanın birlikte çalışması şeklinde gerçekleşmektedir. Bu türden bir etkileşimde etkileşim düzeyinin bir arada bulunmaya göre daha yüksek olduğu rahatlıkla söylenebilir. Öyle ki, artık robot ve insan ortak bir amaca hizmet etmesi için kendi görevlerini yerine getirirken ortak bir alanda bir arada çalışmaktadır. Ancak robot ve insan arasında herhangi bir fiziksel temas yine söz konusu değildir. Bu çalışma, çoğunlukla eş zamanlı olarak gerçekleşebildiği gibi bazen de sıralı şekilde gerçekleşmektedir. Sıralı bir şekilde gerçekleştirilen çalışma ise literatürde çoğu zaman sıralı çalışma olarak isimlendirilmektedir. Buna göre robot ve insan ortak bir çalışma alanında faaliyet göstermesine rağmen, faaliyetlerini sırayla yerine getirmektedir. Genellikle insan bir parça üzerinde faaliyet gösterirken robot durmaktadır. Etkileşim türlerinin sonuncusu ise insan-robot iş birliğine dayanmaktadır. İş birliği ifadesi her ne kadar birlikte çalışmaya benzese de, daha çok robot ve insanın ortak bir amacı gerçekleştirmek için aynı anda ve aynı ortamda bir araya gelerek ortak bir eylem içinde bulunmasını ifade etmektedir. Robot ve insan bu etkileşim türünde fiziksel temas halinde olabileceği gibi, farklı iletişim teknikleri sayesinde temas olmadan da ortak eylemi yürütebilmektedir. İş birliği, birlikte çalışma türüne kıyasla, insan ve robotun kabiliyetlerini birleştirerek ortak hedefin gerçekleştirilmesini ifade etmektedir. Endüstriyel uygulamalarda birlikte çalışma ve iş birliği arasındaki bu ayrım, endüstriyel parça üzerinden daha net olarak ifade edilebilir. Öyle ki, birlikte çalışmanın söz konusu olduğu uygulamalarda robot ve insan ayrı iki parça üzerinde işlem yürütmektedir. İş birliğinde ise robot ve insan aynı parça üzerinde faaliyet göstermektedir. Endüstriyel ortamda insan-robot arasındaki etkileşime konu olan türlerin temsili Şekil 3.1'de gösterilmektedir.

Etkileşim türleri arasında sıralı çalışma, bir arada bulunma türünden etkileşimle birlikte endüstride en çok karşılaşılan türlerdendir (IFR, 2020b). Ancak sahip oldukları potansiyel sebebiyle birlikte çalışmanın ve iş birliğinin yer aldığı uygulamalara da gün geçtikçe daha sık rastlanmaktadır. Sonuç olarak bu etkileşim türlerinin hepsi, literatürde iş birlikçi (ing. Collaborative) olarak ifade edilen operasyonlara konu olmaktadır. Aralarındaki farkı ise uygulamanın gerektirdiği etkileşim düzeyi ortaya koymaktadır.



Şekil 3.1: İnsan-robot etkileşimi türleri: (a) etkileşim yok, (b) bir arada bulunma, (c) birlikte çalışma, (d) sıralı çalışma, (e) iş birliği

3.2 İş Birlikçi Operasyonlar

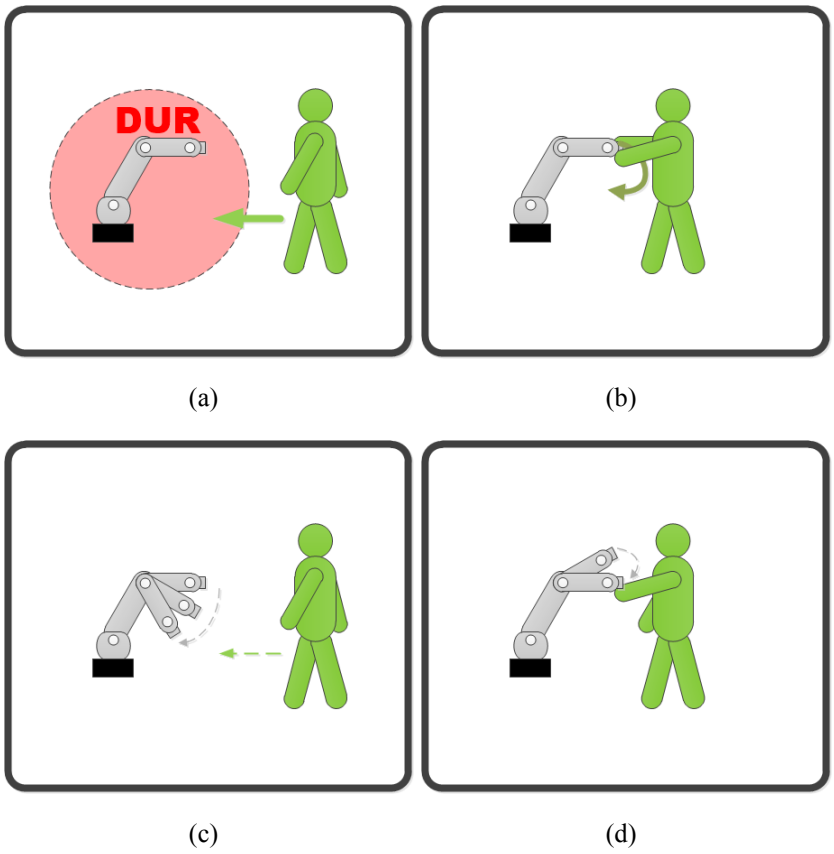
Mevcut etkileşim türlerinden de anlaşılacağı gibi, insan ve robotun iş birlikçi bir operasyon yürütebilmesi için aralarında ayrım oluşturan engellerin ortadan kaldırılması gerekmektedir. Bu sayede insan ve robot, birlikte aynı ortamı paylaşabilmekte, birlikte çalışabilmekte, birbirine yardım edebilmekte veya birbirini tamamlayabilmektedir. Ancak bu durum beraberinde de yeni ihtiyaçlar doğurmaktadır. Emniyet sınırlarının ortadan kalkmasıyla beraber, artık engellerin algılanmasına, çarpışmaların önlenebilmesi için uygun kaçınma stratejilerinin uygulanmasına ve önlenemeyen çarpışma durumlarında ise insana verilen zararın en aza indirilmesine ihtiyaç duyulmaktadır. Yaşanan bu değişimler ise yeni

tanımların ortaya çıkmasına ve yeni düzenlemelerin yapılmasına neden olmaktadır. ISO (2011) tarafından ISO 10218-1:2011 standardında yapılan güncellemeler de bunu göstermektedir. Endüstriyel robotlar için emniyet gereksinimlerini düzenleyen bu standarda yapılan güncellemelerde, iş birlikçi operasyon, iş birlikçi çalışma alanı ve iş birlikçi robot tanımlarına yer verilmektedir. Benzer şekilde yine ISO (2016) tarafından ilan edilen ISO/TS 15066:2016 teknik şartnamesi de, ISO 10218-1:2011 ve ISO 10218-2:2011 standartlarını tamamlayarak, iş birlikçi endüstriyel robot sistemlerine ve iş birlikçi çalışma ortamı için emniyet gereksinimlerine açıklık getirmektedir. Bu teknik şartnameyle beraber aynı zamanda iş birlikçi operasyonların da tanımı yapılmaktadır.

ISO 10218-1/2 standartlarında ve ISO/TS 15066:2016 teknik şartnamesinde yapılan tanımlara göre iş birlikçi operasyonlar dört farklı tipte sınıflandırılmaktadır. Bu iş birlikçi operasyon tiplerinin her biri birbirinden bağımsız olup, bir endüstriyel uygulamada ayrı ayrı veya kombinasyon halinde uygulanabilmektedir (Magrini vd., 2020).

İş birlikçi operasyon tiplerinden ilki ”emniyet dereceli gözlenen durdurma - safety-rated monitored stop (SMS)” olarak ifade edilmektedir. Bu formdaki bir operasyonda insan, iş birliğine konu olan çalışma alanına girdiğinde robot durdurulmakta veya duruma göre duraklatılmaktadır. Genellikle bu türden uygulamalarda insan varlığını algılamak için bir veya daha fazla algılayıcı kullanılmaktadır. Emniyeti sağlamak için takip edilen stratejiden de anlaşılacağı üzere SMS, iş birlikçi operasyonlar içinde en basit olanıdır. Bu sebeple uygulanması da diğer yöntemlere kıyasla daha kolaydır. Bir diğer iş birlikçi operasyon tipi ise ”elle yönlendirme - hand guiding (HG)” şeklinde ifade edilmektedir. Elle yönlendirmede, iş birlikçi çalışma alanındaki insana robot hareketi üzerinde doğrudan kontrol imkanı sağlanmaktadır. Robot herhangi bir arayüze gerek duymadan insan tarafından elle yönlendirilmekte ve robota hareket edeceği noktalar manuel olarak öğretilmektedir. Bu sıradaki robot hareketi düşük hızlarda gerçekleşmekte ve robotun statik konfigürasyonlarda durabilmesi için kendi ağırlığı denetleyici tarafından dengelenmektedir (Magrini vd., 2020). Emniyet de bu şekilde sağlanmaktadır. ”Hız ve ayırma gözlemi - speed and separation monitoring (SSM)” operasyon tipinde ise insanın bulunduğu konuma göre robotun hareketleri ayarlanmaktadır. Bunun için robotun hızı ve izlediği yol sürekli olarak takip edilmektedir. Bu sayede iş birliğine dayalı çalışma ortamında insan ve robot arasında yeterli mesafe korunmakta ve risk azaltılmaktadır. Bunu gerçekleştirebilmek için genellikle bir insan iş birliği alanına girdiğinde, robot hareket hızı yavaşlatılmaktadır. İnsanın robota daha fazla yaklaşması durumunda ise aynı SMS tipinde olduğu gibi robot hareketi durdurulmaktadır. Bu tür bir sistemdeki asıl zorluk, herhangi bir çarpışmayı önlemek için insanları ve robotları gerçek zamanlı olarak izlemektir. Çarpışmayı önlemek için hızlı ve güvenilir yöntemlere ihtiyaç duyulmaktadır. “Güç ve kuvvet sınırlama - power and force limiting (PFL)” ise iş birlikçi operasyon tiplerinin

sonuncusudur. Bu tarz bir iş birliğinde robot ve insan arasında istenen ve istenmeyen şekilde temas gerçekleşebilir. Emniyeti sağlamak için temas kuvveti, hasar veya yaralanmaların beklenmediği değerlerle sınırlandırılmaktadır. Güç ve kuvvet sınırlamanın söz konusu olduğu operasyon tipleri, bu iş için özel olarak tasarlanan endüstriyel robotlarla yürütülmektedir. Bunun için akımın takip edildiği, kuvvet-tork algılayıcıların kullanıldığı, düşük güçlü aktüatörlerin yer aldığı robotik sistemler tasarlanmaktadır. İş birlikçi uygulamalar için özel olarak tasarlanan iş birlikçi robotlar (ing. Collaborative Robots/Cobots) verilebilecek örneklerin başında yer almaktadır. Bahsi geçen tüm iş birlikçi operasyon tipleri Şekil 3.2'de özet olarak gösterilmektedir.



Şekil 3.2: İş birlikçi operasyon tipleri: (a) emniyet dereceli gözlenen durdurma (SMS), (b) elle yönlendirme (HG), (c) hız ve ayırma gözlemi (SSM), (d) güç ve kuvvet sınırlama (PFL)

IFR (2020a) raporlarına göre iş birlikçi robot kullanımında gün geçtikçe bir artış yaşanmaktadır. Bu durum, özellikle elle yönlendirme ve güç/kuvvet sınırlama tipi iş birlikçi operasyonlara daha sık başvurulması anlamına gelmektedir. Ancak yine aynı raporlara göre geleneksel endüstriyel robotların endüstrideki kullanımı hala ezici bir çoğunluğa sahiptir.

Bu da, SMS ve SSM tipi iş birlikçi operasyonlar için yeni yöntemlerin geliştirilmesi gerekliliğini ortaya koymaktadır. Bu sebeple, bu tez çalışmasında geliştirilen yöntemlerde genel olarak SMS ve SSM tipi iç birlikçi operasyonlar dikkate alınmaktadır.

3.3 Geleneksel Endüstriyel Robotlar ve İş Birlikçi Robotlar

Endüstriyel robot manipülatörler, uzunca bir süredir endüstriyel alanlarda faaliyet göstermektedir. Yakın zamana kadar kapalı sistemlerin ardında insanlardan izole şekilde faaliyet gösteren bu robotlar, değişen ihtiyaçlar ve gelişen teknolojiyle beraber yeni bir kimliğe bürünmektedir. Özellikle Endüstri 4.0 programında olduğu gibi iş birlikçi robotik, sanayi devriminin öncü teknolojilerinden biri olarak kabul edilmektedir (Magrini vd., 2020). Endüstriyel ortamlardaki iş uygulamalarındaki bu değişiklikler, haliyle robotik sistem tasarımlarının da değişmesine sebep olmaktadır. İş birlikçi robotların, yani cobotların, ortaya çıkış öyküsü ise tam olarak bu sebeplere dayanmaktadır. Bu yeni tasarımların ortaya çıkması hali hazırda kullanılan endüstriyel robotların ise artık geleneksel endüstriyel robotlar olarak adlandırılmasına yol açmaktadır.

İş birlikçi robotlar, endüstriyel ortamda görevlerini yerine getirirken insanlarla birlikte ve onlara yakın olarak çalışmak için özel olarak tasarlanmış robot manipülatörlerdir. Emniyetli bir etkileşim için ise çeşitli yakınlık sensörleri, yük sensörleri ve tehlikeli etkileşim riskini azaltan tasarımlarla donatılmaktadır. Bu sayede insanın yeteneklerini, kendi güç ve hassasiyet yetenekleriyle birleştirmekte ve yürütülen operasyonların verimini arttırmaktadır. Benzer şekilde bu robotların eğitilmeleri de geleneksel endüstriyel robotlara göre farklılık göstermektedir. Birçok iş birlikçi robot, programlama araçları kullanılarak programlanmak yerine, kolları manipüle ederek eğitim veren insanlar tarafından programlanmaktadır. Kolayca yeniden programlanabilir olmaları ise bu anlamda geleneksel endüstriyel robotlara göre daha esnek olmalarını sağlamaktadır. Böylece birbirinden çok farklı tekrarlayan işlerde bile hızlıca faaliyete geçebilmektedir. KUKA (2020) tarafından geliştirilen lbr iiwa modeli ve UniversalRobots (2020) tarafından geliştirilen UR5 modeli iş birlikçi endüstriyel robotların ilk örnekleri arasında yer almaktadır. Bu robotlar Şekil 3.3'de gösterilmektedir. Şu an hali hazırda bir çok üretici kendi iş birlikçi robotlarını tasarlamakta ve üretmektedir (Hentout vd., 2019). Cobotlar özel tasarımları nedeniyle geleneksel endüstriyel robotlara göre iş birlikçi operasyon tiplerine daha uygun, daha pratik ve daha az maliyetli çözümler sunmaktadır.



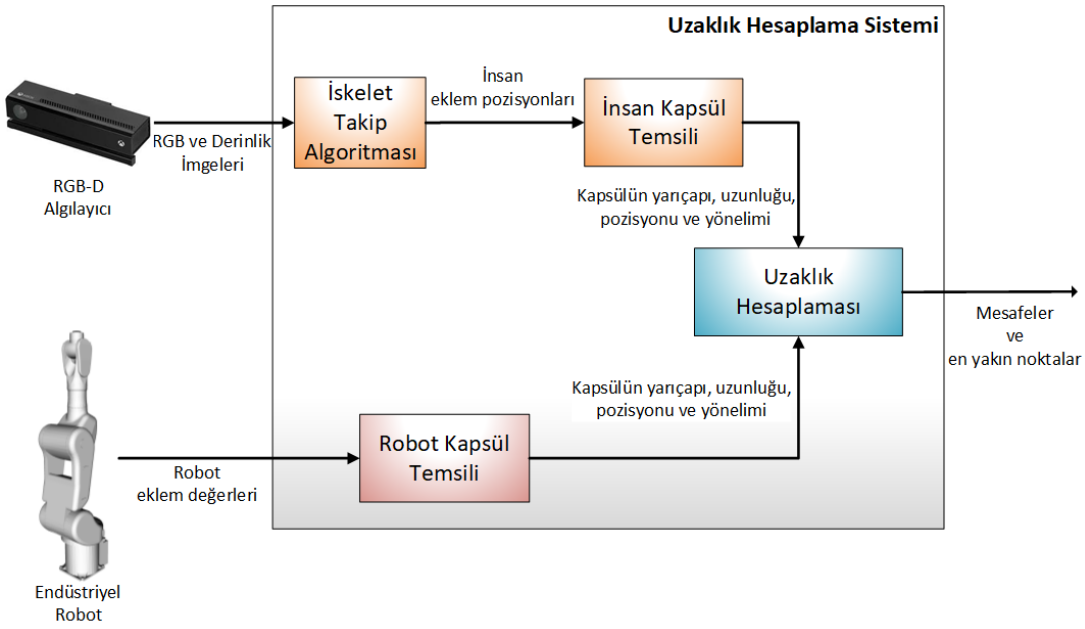
(a) (b) (c) (d)

Şekil 3.3: Geleneksel endüstriyel robot manipülatörleri: (a) Kawasaki RS005L (Kawasaki, 2020), (b) ABB IRB 120 (ABB, 2020), İş birlikçi endüstriyel robot manipülatörleri: (c) KUKA LBR iiwa (KUKA, 2020), (d) Universal Robots UR5 (UniversalRobots, 2020)

İş birlikçi robotların iş birlikçi operasyonlar açısından sundukları tüm avantajlara rağmen, günümüz endüstrisinde hala yoğun bir şekilde geleneksel endüstriyel robot manipülatörleri kullanılmaktadır. IFR (2020a) raporlarında açıkça görülebileceği üzere 2019 yılında gerçekleştirilen robot kurulumlarının sadece %4.8'i iş birlikçi robotları kapsamaktadır. Bu oran, önceki yıllara göre artış gösterse de hala yüksek seviyelerde değildir. Bunda iş birlikçi robotların tedariki, arzı ve maliyeti ile ilgili sıkıntıların etkili olduğu söylenebilir. Dolayısıyla bu veriler, kısa ve orta vadede bile olsa geleneksel endüstriyel robotların endüstrideki baskınlığının devam edeceği şeklinde yorumlanabilir. Bu kadar yaygın kullanılmalarına rağmen, Şekil 3.3'de gösterilen geleneksel endüstriyel robotlar çoğunlukla sahip oldukları eski teknoloji sebebiyle iş birlikçi robotların hemen hemen hepsinin sahip olduğu imkanlardan yoksundurlar. İş birlikçi robotların sahip oldukları donanımlar ve bunlara özel tasarımlar, geleneksel endüstriyel robotlar için ancak harici müdahalelerle sağlanmaktadır. Bu yüzden, geleneksel robotlar emniyetli bir insan-robot etkileşimi için özel tasarımlara ve yeni geliştirilen yöntemlere ihtiyaç duymaktadır. Tüm bu verilerin ışığında bu tez çalışmasında, geleneksel endüstriyel robotların özellikle SMS ve SSM tipinden iş birlikçi uygulamalarda kullanabileceği şekilde bir sistem tasarımı ve yeni bir yol planlama yöntemi önerilmektedir.

4. GÜVENLİ ÇALIŞMA ORTAMININ ALGILANMASI VE MODELLENMESİ

İnsan-robot etkileşiminde emniyetle ilgili temel konulardan biri, robot ile insan arasındaki mesafenin tespit edilebilmesidir. Bunun için, çalışma ortamında yer alan insan ve robotun algılanması ve modellenmesi gerekmektedir. Bu tez çalışması kapsamında, emniyetli insan-robot etkileşimini sağlamak için Şekil 4.1'de gösterildiği gibi gerçek zamanlı bir uzaklık hesaplama sistemi önerilmektedir (Secil ve Ozkan, 2022). Bu sisteme göre, insan bir RGB-D algılayıcı tarafından izlenmekte ve insanın eklem pozisyonları iskelet takip algoritması aracılığıyla elde edilmektedir. Robot ve insan temsili ise uzaklık hesaplamasını basitleştirmek ve hızlandırmak için geometrik ilkellerden biri olan kapsüllerle yapılmaktadır.



Şekil 4.1: Uzaklık hesaplama sistem şeması

4.1 Geometrik İlkeller

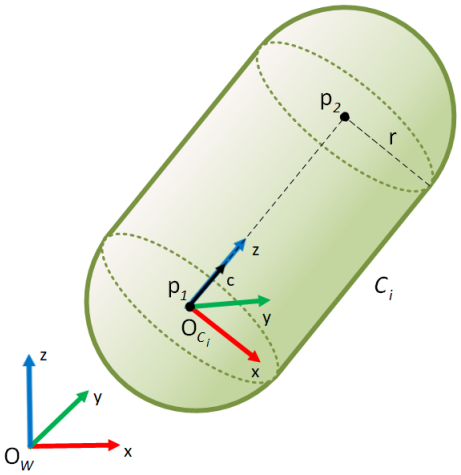
Engelden kaçınma ve robot hareket planlamasının söz konusu olduğu bir endüstriyel çalışma ortamında, emniyetli bir operasyon için uzaklık hesaplamasının yeterince hızlı olması gerekmektedir. Kısa bir zaman aralığında insan ve robot arasındaki minimum uzaklığı hesaplamak için ise insan ve robot geometrik ilkellerle temsil edilebilir. Geometrik

ilkeller (örneğin küreler), basit ve kolay uygulanabilir geometrileri nedeniyle hareket planlama, çarpışmadan kaçınma ve insan-robot gösterimi gibi robotik uygulamalarda yaygın olarak tercih edilmektedir. Geometrik ilkellerden biri olan kapsüller ise insan ve robot uzuvlarına benzerlikleri sebebiyle sıklıkla modelleme için kullanılmaktadır. Aynı zamanda kapsüllerin basit formu ve dışbükey yapısı, minimum uzaklık hesaplaması için kolay ve verimli bir çözüm sunmaktadır. Bu sebeple bu tez çalışmasında insan ve robotu temsil etmek için kapsüller kullanılmaktadır.

Robot ve insan, elverişli bir kapsama için genellikle sadece birkaç kapsülle temsil edilmektedir. Bununla birlikte, kapsül sayısı arttığında daha yüksek doğruluk elde edilmesine rağmen hesaplama maliyeti de artmaktadır (Safeea ve Neto, 2019). Endüstriyel bir ortamda gerçek zamanlı kontrol için düşük hesaplama maliyetine ve kısa hesaplama süresine ihtiyaç duyulmaktadır. Emniyetli etkileşim için ise bunların yanında yüksek doğruluğa sahip olmak gerekmektedir. Kapsüllerle temsil edilmeyen insan uzuvları (baş, kollar vb.) bu anlamda bir emniyet açığı oluşturabilmektedir. Ancak her bir uzvun gereğinden fazla kapsül ile temsil edilmesi de hesaplama süresini artıracağından, dinamik bir ortamda engellerden kaçınmak uzun hesaplama süreleri nedeniyle mümkün olmayabilir. Bu sebeple, insan ve robotun temsilinde yeterli ve gerekli sayıda kapsülün kullanılması gerekmektedir.

4.1.1 Kapsül geometrisi

Bir kapsül C_i, Şekil 4.2'de gösterildiği gibi yarıçapı r ve yüksekliği $L(\overline{p_1 p_2})$ olan bir silindir ile bu silindirin her iki ucunda yer alan r yarıçaplı iki yarım küreden oluşmaktadır. Bu sebeple kapsüller, aynı zamanda küresel silindir (ing. Spherocylinder) olarak da adlandırılmaktadır.



Şekil 4.2: Kapsül geometrisi

Bir kapsülü tanımlayan dışbükey küme C, Denklem (4.1)'de olduğu gibi, r yarıçaplı S küresine ve L uzunluğundaki doğru parçasına ait dışbükey kümelerin Minkowski toplamı şeklinde ifade edilmektedir.

$$C = S \oplus L = \{s + l : s \in S, l \in L\} \tag{4.1}$$

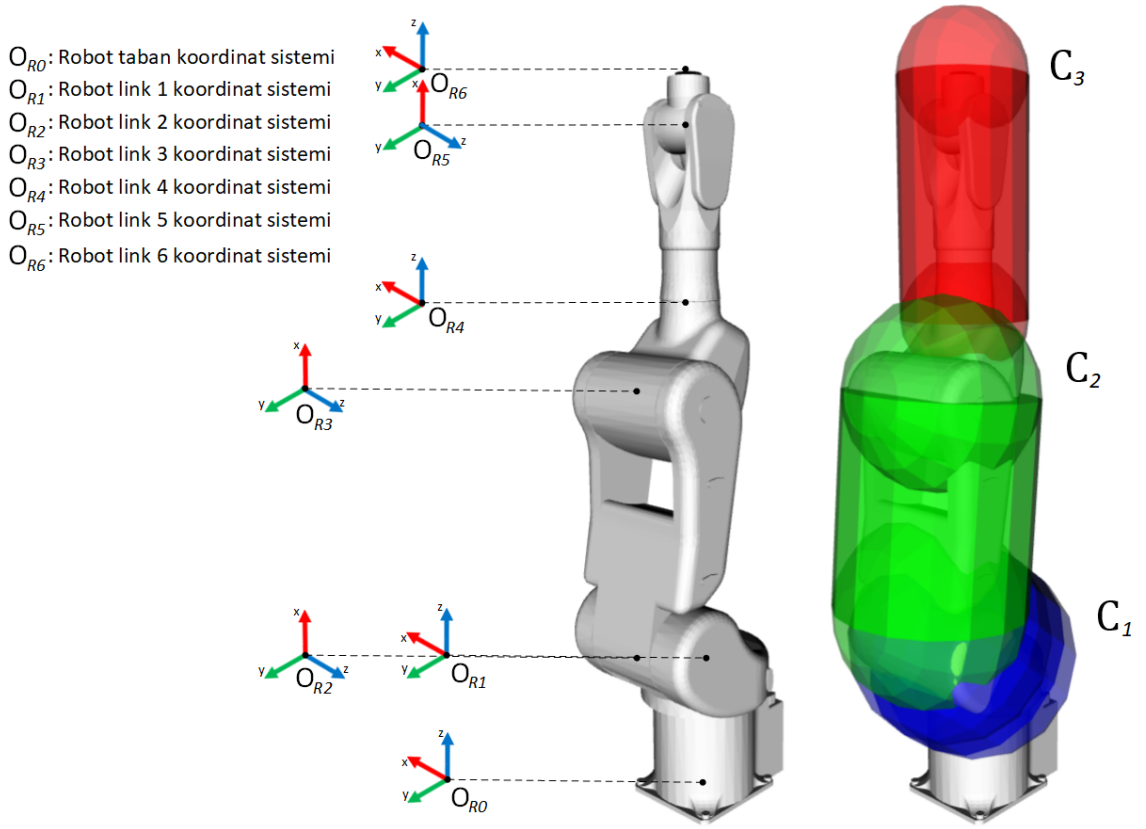
Kapsül ekseni $c \in \mathbb{R}^3$ ise çoğunlukla kapsül koordinat sistemi O_{C_i}'nin z ekseni boyunca yönlendirilmektedir. Kapsülün başlangıç noktasının, doğru parçası $\overline{p_1 p_2}$'nin merkezinde olduğu veya bu doğru parçasının uç noktalarından birinde bulunduğu kabul edilmektedir. Bu tezde kapsül koordinat sistemine ait başlangıç noktasının, uç noktalardan birinde yer aldığı varsayılmaktadır. Buna göre, robot/insan temsilini oluşturan her bir kapsül dünya koordinat sistemi O_W'ye göre konumlandırılmakta ve yönlendirilmektedir. Herhangi bir kapsülün konumu ve yönelimi ise $T_{C_i}^W$ ile ifade edilmektedir.

4.2 İnsan ve Robotun Temsili

Hem endüstriyel robot manipülatörler, hem de insan vücudu, kapsüllerle temsil edilmek için uygun bir geometriye sahiptir. Ancak etkili ve verimli bir şekilde uzaklık hesabı yapabilmek için hem kapsül miktarının, hem de kapsül boyutlarının uygun bir şekilde belirlenmesi gerekmektedir. Bu nedenle kapsüllerin sayısı ve boyutu, robotun veya insan vücudunun fiziksel yapısına göre yeteri kadar kısım ve hacmi kapsayacak şekilde seçilmelidir. Bunun yanında, bu tezde ağırlıklı olarak geleneksel endüstriyel robotlar dikkate alındığından, kapsüllerin sayısının, boyutunun ve konumunun farklı endüstriyel kurulumlarda (örneğin endüstriyel mobil manipülatörler) değişikliğe uğrayabileceği göz önünde bulundurulmalıdır. Ayrıca, kapsüllerin konumlandırılması da algılayıcının türüne (RGB-algılayıcı, IMU vb.) ve uzuvlar hakkında elde edilen bilginin içeriğine (pozisyon ve yönelim) göre farklılık gösterebilmektedir. Bu kısımda, RGB-D algılayıcı aracılığıyla iskelet takip algoritmasından elde edilen insana ait eklem pozisyonlarının ve robot denetleyicisinden elde edilen eklem konfigürasyon verilerinin işlenmesi anlatılmaktadır.

4.2.1 Robotun geometrik temsili

Robot manipülatörlerin temsilinde her bir uzuv için birer adet kapsül kullanılabileceği gibi, basitleştirme adına daha az sayıda kapsül de kullanılabilmektedir. Şekil 4.3'te üç kapsül $C_i, i = 1, 2, 3$ ile temsil edilen bir 6-DOF endüstriyel manipülatörü yer almaktadır. Bu robot modelinde, C_1 kapsülü link 1'i, C_2 kapsülü link 2'yi ve C_3 kapsülü link 3-6'yı kapsamaktadır. Burada son üç eklem yalnızca küresel bilek dönüklüğünü belirlediğinden, dört farklı robot uzvu tek bir kapsülle temsil edilmektedir.



Şekil 4.3: Bir endüstriyel robot manipülatörün 3-B modeli ve geometrik temsili

Herhangi bir anda, n serbestlik derecesine sahip bir robot manipülatörün eklem değerleri $q_i, i = 1, ..., n$, robotun eklem konfigürasyonunu $Q = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}^T$ belirlemektedir. Robota ait eklem konfigürasyonu Q ise daha sonra m tane kapsülün konumunu ve yönelimini $T_{C_i}^W, i = 1, ..., m$ hesaplamak için kullanılmaktadır.

Endüstriyel robot manipülatörlerin kinematiği her zaman bilinmektedir. Ayrıca robot denetleyicileri de, robota ait eklem değerlerini (q_i) gerçek zamanlı olarak sağlamaktadır. Böylece, robot link koordinat sistemlerinden robot tabanına olan dönüşüm de $T_{Ri}^{R0}, i = 1, ..., n$ hesaplanabilmektedir. Kapsül koordinat sistemleri de robot linklerine ait koordinat sistemlerine sabitlenecek şekilde tanımlanırsa, kapsül koordinat sistemlerinin konumu ve yönelimi linklere ait koordinat sistemlerinin konumu ve yönelimi tarafından belirlenmektedir. Kapsül koordinat sistemlerinin konumu ve yönelimi ise bu sayede Denklem (4.2)'de verilen eşitlik kullanılarak hesaplanabilir.

$$T_{C_j}^{R0} = T_{Ri}^{R0} T_{C_j}^{Ri}, \forall j \in [1, m], \forall i \in [1, n] \tag{4.2}$$

Burada $T_{C_j}^{Ri} \in \mathbb{R}^{4x4}$ matrisi, kapsül (C_j) koordinat sisteminin robot link (Ri) koordinat sistemine göre ötelemesini ve döndürmesini içeren homojen dönüşüm matrisidir. Bir kapsülü tanımlamak için $C_j = \left\{T_{C_j}^{R0}, h_j, r_j\right\}$ ifadesi kullanılmakta ve h_j ile r_j değerleri sırasıyla j'nci kapsülün yüksekliği ile yarıçapını ifade etmektedir. Örnek olarak, Şekil 4.3'teki robot manipülatörü üç kapsül $C_j, j = 1, 2, 3$ ile temsil edilmektedir. Bu kapsüller sırasıyla $R1$, $R2$ ve $R4$ linkleri ile ilişkilidir. Bu sebeple, ilgili dönüşüm matrisleri ($T_{C_j}^{R0}$) sırasıyla Denklem (4.3), Denklem (4.4) ve Denklem (4.5)'te verilen eşitlikler kullanılarak hesaplanmaktadır.

$$T_{C_1}^{R0} = T_{R1}^{R0} T_{C_1}^{R1} \tag{4.3}$$

$$T_{C_2}^{R0} = T_{R2}^{R0} T_{C_2}^{R2} \tag{4.4}$$

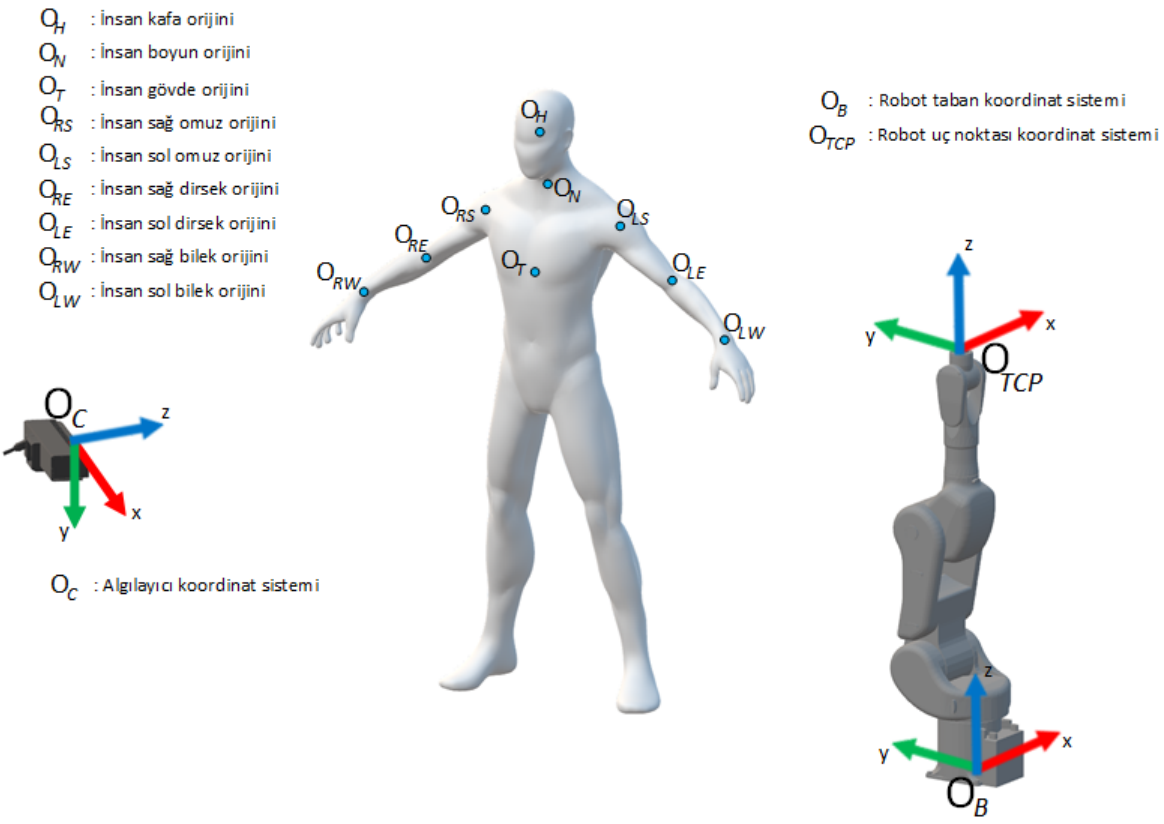
$$T_{C_3}^{R0} = T_{R4}^{R0} T_{C_3}^{R4} \tag{4.5}$$

4.2.2 İnsanın algılayıcı temelli geometrik temsili

Uzaklık hesaplamaları için, robot uzuvlarının yanında insan vücut kısımlarının da pozisyonuna ve yönelimine ihtiyaç duyulmaktadır. Hesaplamaları basitleştirmek için insan da, endüstriyel robot manipülatörüne benzer şekilde kapsüllerle temsil edilmektedir. Endüstriyel ortamlar düşünüldüğünde, özellikle insan vücudunun üst kısımları endüstriyel robotlarla temas riski taşımaktadır. Bu nedenle birçok endüstriyel uygulama için çoğunlukla bir insanın sadece üst bedeninin bir temsilini oluşturmak gerekli ve yeterlidir (Bdiwi vd., 2017). Bu tez çalışmasında, insanın üst vücudunu temsil etmek için altı kapsül kullanılmaktadır; biri kafa için, biri gövde için, ikisi üst kollar için ve ikisi de alt kollar için. Elbette mobil robot manipülatörleri içeren farklı senaryolarda insan alt vücut temsiline de ihtiyaç duyulabilir. Ancak bu durumda da alt vücut konfigürasyonu, üst vücut ile aynı şekilde temsil edilecektir. Böyle bir durumda, alt vücudu temsil etmek için iki (bütün bacak) veya dört (üst ve alt bacakların her biri için ikişer) kapsül kullanılabilir. Robotun kapsüllerle temsil edilişine benzer şekilde, kapsüllerin boyutu, konumu ve yönelimleri de insan için ayrıca tanımlanmaktadır. Kapsül boyutlarının uygun şekilde ayarlanması, doğruluğu ve dolayısıyla etkileşimin emniyetini de doğrudan etkilemektedir.

4.2.2.1 RGB-D algılayıcı ile iskelet takibi

İnsana ait kapsüllerin konumunu ve yönelimini hesaplayabilmek için Şekil 4.4'te gösterildiği gibi kafanın (O_H), gövdenin (O_T), sol/sağ omuzların (O_{LS},O_{RS}), sol/sağ dirseklerin (O_{LE},O_{RE}) ve sol/sağ bileklerin (O_{LW},O_{RW}) başlangıç noktaları (orijinleri) kullanılabilir. Orijin adı verilen iskelet eklemleri, farklı türden algılayıcılarla olabileceği gibi, bir RGB-D algılayıcıdan alınan veriler kullanılarak da tespit edilebilmektedir. RGB-D algılayıcının sağladığı veriler, iskelet izleme algoritmaları tarafından gerçek zamanlı olarak orijinlerin konumlarını belirlemek için kullanılmaktadır. Bunun için, günümüzde birçok ticari ve açık kaynaklı iskelet izleme yazılımı mevcuttur (Cubemos, 2020; Nuitrack, 2020; LIPS, 2020).



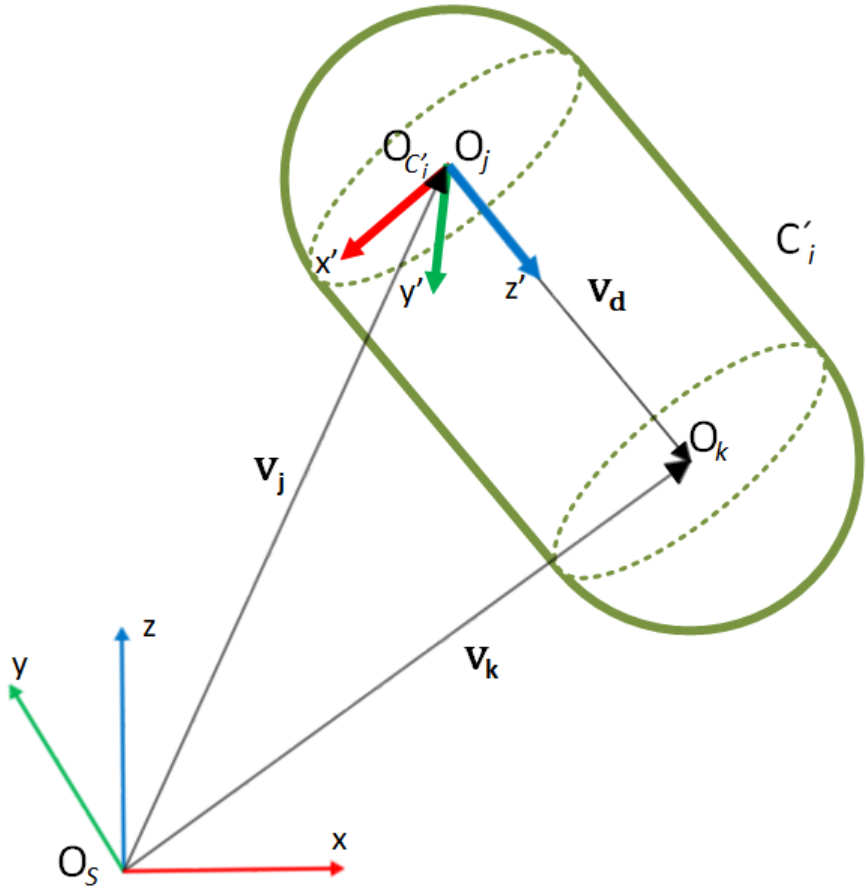
Şekil 4.4: Robot ile algılayıcının koordinat sistemleri ve iskelet eklemlerinin orijinleri

4.2.2.2 İnsana ait kapsüllerin algılayıcıya göre dönüşümü

İskelet takip algoritmaları sayesinde iskelet eklemlerine ait pozisyonlarının elde edilmesiyle beraber, uzaklık hesabı için insanı temsil eden kapsüllerin konumunun ve yöneliminin hesaplanmasına ihtiyaç duyulmaktadır. Bu hesaplama ilk olarak algılayıcı koordinat sistemine göre yapılmaktadır. Şekil 4.5'te bir insan vücut kısmını temsil eden

koordinat sistemi $O_{C_i'}$ ile kapsüllerin $C_i', i = 1, ..., 6$ hesaplanması için kullanılan parametreler gösterilmektedir. Burada $V_j \in \mathbb{R}^3$ ve $V_k \in \mathbb{R}^3$, algılayıcı koordinat sistemine (O_S) göre, O_j ve O_k'nin konumunu gösteren vektörlerdir. O_j ve O_k birbirine sıralı bir şekilde bağlı iskelet eklemlerini temsil etmektedir $\{j, k\} \in \{\{RS, RE\}, \{RE, RW\}, \{LS, LE\}, \{LE, LW\}, \{N, T\}, \{N, H\}\}$. $V_d \in \mathbb{R}^3$ ise, O_j'den O_k'ye doğru tanımlanan bir vektördür ve Denklem (4.6)'te gösterildiği gibi hesaplanmaktadır.

$$V_d = V_k - V_j \tag{4.6}$$



Şekil 4.5: İnsan temsili için kullanılan kapsüllerin konumunun ve yöneliminin hesaplanması

Algılayıcı koordinat sistemine göre her bir kapsüle ait dönüşüm matrisi Denklem (4.7)'de verilmektedir.

$$T_{C_i'}^{S} = \begin{bmatrix} R_{C_i'}^{S} & t_{C_i'}^{S} \\ 0_{1x3} & 1 \end{bmatrix} \tag{4.7}$$

Burada öteleme vektörü $t_{C_i'}^{S} = V_j$ ile hesaplanırken, döndürme matrisi ise $\widehat{r}_x^S, \widehat{r}_y^S, \widehat{r}_z^S \in \mathbb{R}^3$ birim vektörleri kullanılarak $R_{C_i'}^{S} = \begin{bmatrix} \widehat{r}_x^S & \widehat{r}_y^S & \widehat{r}_z^S \end{bmatrix}$ şeklinde tanımlanmaktadır. Bu birim vektörler daha sonra sırasıyla Denklem (4.8), Denklem (4.9) ve Denklem (4.10)'da verilen eşitlikler kullanılarak hesaplanmaktadır.

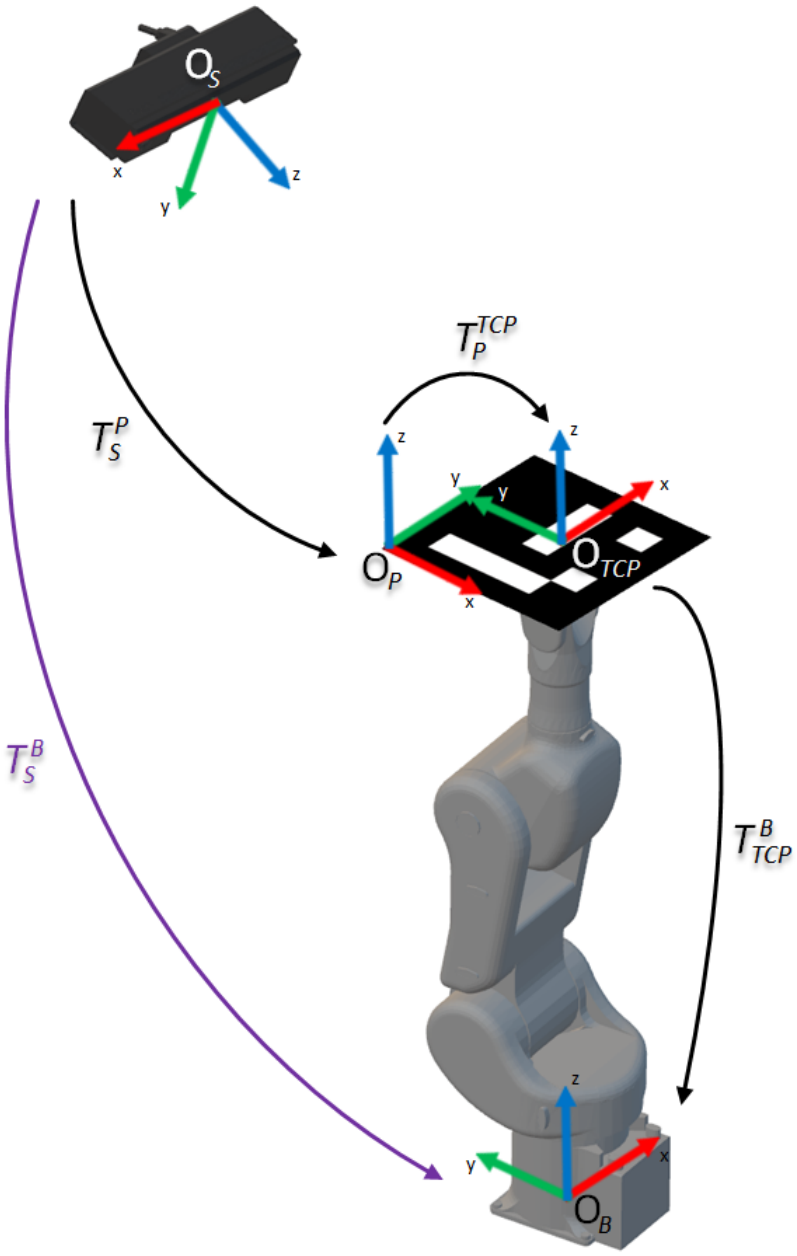
$$\widehat{r}_z^S = \frac{V_d}{|V_d|} \tag{4.8}$$

$$\widehat{r}_y^S = \frac{\Gamma}{|\Gamma|}, \Gamma = V_d \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.9}$$

$$\widehat{r}_x^S = \widehat{r}_y^S \times \widehat{r}_z^S \tag{4.10}$$

4.2.2.3 Algılayıcının el-göz kalibrasyonu

İnsan ve robot arasındaki uzaklık hesabı için, insanı ve robotu temsil eden tüm kapsüllerin konumunun ve yöneliminin aynı koordinat sisteminde tanımlanması gerekmektedir. Bu nedenle, C_i' kapsüllerinin konumu ve yönelimi, robot taban koordinat sistemine (O_B) göre $T_{C_i'}^{B} = T_S^B T_{C_i'}^{S}$ ile hesaplanmaktadır. Algılayıcı koordinat sisteminden, robot taban koordinat sistemine olan homojen dönüşüm matrisi T_S^B ise el-göz kalibrasyon problemini çözen herhangi bir yöntem kullanılarak hesaplanabilmektedir. Bu tez çalışmasında, sonucu gerçek zamanlı olarak gerçekleştirilen deneylerde kullanılmak üzere RGB-D algılayıcı ve endüstriyel robot arasında el-göz kalibrasyonu (ing. Hand-eye calibration) gerçekleştirilmektedir. Gerçekleştirilen el-göz kalibrasyon kurulumu Şekil 4.6'da gösterilmektedir. Bu kalibrasyon esnasında kalibrasyon modeli olarak ArUco işaretçilerinden yararlanılmaktadır. Kalibrasyon modeli robot uç noktasına sabitlenirken, algılayıcı ise robot çalışma alanını görebilecek bir konuma sabitlenmektedir.



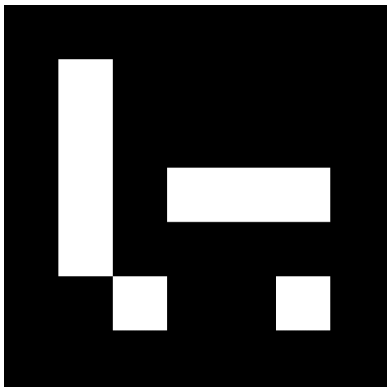
Şekil 4.6: ”El-göz kalibrasyon kurulumu

El-göz kalibrasyonun gerçekleştirilmesindeki amaç, algılayıcı koordinat sisteminden robot taban koordinat sistemine olan dönüşümü T_S^B bulmaktır. Şekil 4.6'da da görüleceği üzere bu dönüşüm, Denklem (4.11)'deki eşitlik kullanılarak hesaplanmaktadır.

$$T_S^B = T_{TCP}^B T_P^{TCP} T_S^P \tag{4.11}$$

Robot uç noktasından robot tabanına olan dönüşüm T_{TCP}^{B}, robot kinematiği ile hesaplanabilmektedir. Endüstriyel robotlar için bu dönüşüm, çoğunlukla robot denetleyicisi tarafından da sağlanmaktadır. Algılayıcı koordinat sisteminden kalibrasyon modeline olan dönüşüm T_{S}^{P} ise kamera poz tahmini problemi çözülerek elde edilmektedir. Geride bilinmeyen olarak kalibrasyon modeli ve robot uç noktası arasındaki dönüşüm kalmaktadır. Bu dönüşümü hesaplamak için ise Daniilidis (1999) tarafından ortaya konulan yöntem kullanılmaktadır.

Bu tez çalışmasında, algılayıcı koordinat sisteminden kalibrasyon modeline olan dönüşümü T_{S}^{P} bulmak için kamera poz tahmini problemi çözülmektedir. Bunun için ise kalibrasyon modeli olarak bir ArUco işaretçisinden yararlanılmaktadır. ArUco işaretçileri kamera poz tahmini probleminin çözümü için özel olarak geliştirilmiş referans belirteçleridir. Her ArUco işaretçinin ait olduğu sözlükte ikili kod tarafından belirlenen benzersiz bir kimliği bulunmaktadır. Bir ArUco sözlüğü, hem içerdiği işaretçilerin sayısı, hem de işaretçilerde bulunan bitlerin sayısı belirlenerek yapılandırılabilmektedir (Garrido-Jurado vd., 2014). ArUco işaretçiler kare yapıları sayesinde dört köşe noktası sağlayarak kamera poz tahminine olanak tanımakta ve ayrıca sahip oldukları siyah kenarlar nedeniyle algılayıcı imgelerinde tespiti kolaylaştırmaktadır. Örnek bir ArUco işaretçisi Şekil 4.7'de gösterilmektedir.



Şekil 4.7: ARUCO MIP 36h12 sözlüğünde bir ArUco işaretçisi (Garrido-Jurado vd., 2016)

ArUco işaretçilerinin algılanmasına ek olarak, kamera poz tahmini yapabilmek için kamera kalibrasyon parametrelerine de ihtiyaç duyulmaktadır. Bu aşamada, kamera poz tahmini yapmak için kameranın iç parametreleri ve bozulma katsayıları da kullanılmaktadır. Kullanılan kamera parametreleri, Çizelge 4.1'de verilmektedir. Kameranın işaretçiye göre olan duruşunu doğru bir şekilde tahmin edebilmek için doğru kalibrasyon parametrelerine sahip olmak önemlidir. Bu tezde, OpenCV kütüphanesinde bulunan "aruco" modülü,

algılayıcı koordinat sisteminden işaretleyici koordinat sistemine homojen dönüşüm matrisi T_S^P'yi hesaplamak için işaretçinin kenar uzunluğu ve kamera parametreleri ile birlikte kullanılmaktadır (OpenCV, 2020). Bu işlem sonucunda, ArUco işaretçisinin dört köşe noktası kullanılarak düzlemsel poz tahmini yapılmakta ve T_S^P dönüşümü elde edilmektedir. Eş zamanlı olarak robot uç noktasından robot tabanına olan dönüşüm T_{TCP}^B ise robot kinematiği sayesinde bilinmektedir. Bu dönüşümü elde edebilmek için robot denetleyicisinden okunan robot konfigürasyon verileri kullanılmaktadır.

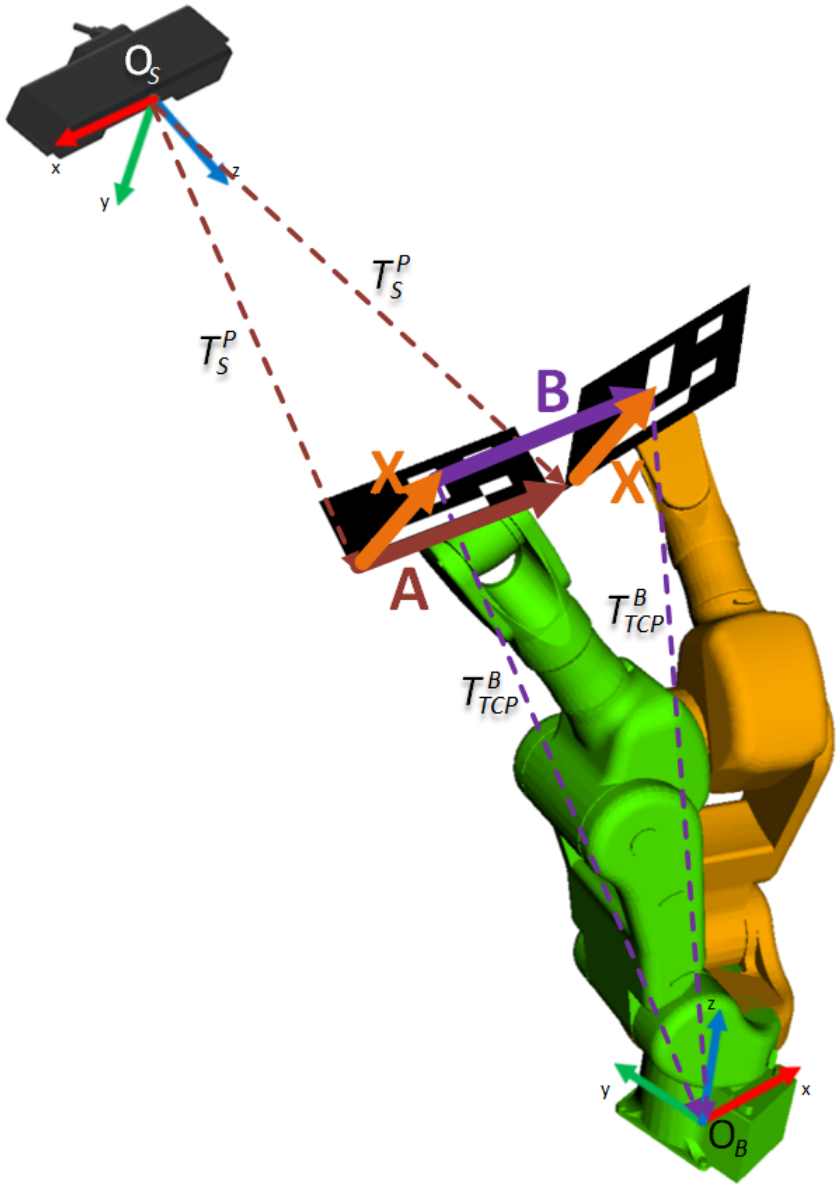
Çizelge 4.1: Kinect v2 kamera iç parametreleri

Kamera parametreleri		**RGB kamera**	**Derinlik kamerası**
Kare çözünürlüğü (piksel x piksel)		1920 x 1080	512 x 424
Odak uzaklığı (piksel)	α	1066.01203	366.38346
	β	1068.87083	367.14380
Görüntü merkez noktası (piksel)	u_0	945.58752	255.64531
	v_0	520.06994	203.98020
Bozulma katsayıları	k_1	0.05936	0.10952
	k_2	-0.06275	-0.31461
	p_1	-0.00157	-0.00126
	p_2	-0.00115	-0.00071

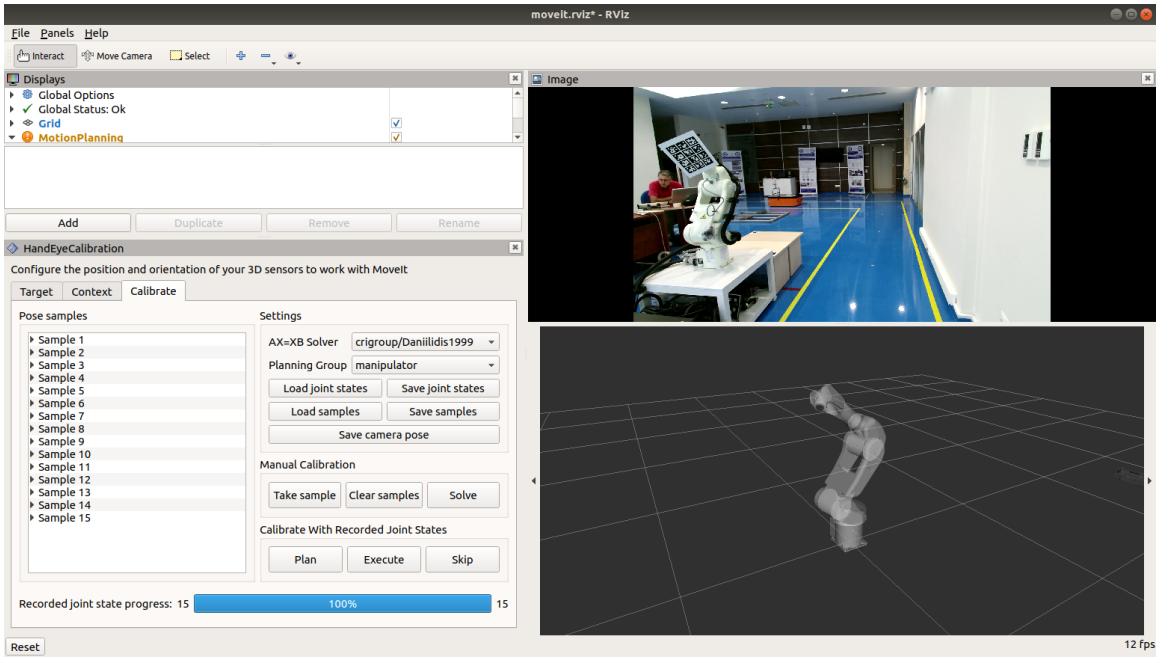
Kalibrasyon modeli ve robot uç noktası arasındaki dönüşüm olan T_P^{TCP}'yi bulabilmek için ise T_S^P ve T_{TCP}^B dönüşümleri kullanılmaktadır. Bunun için, literatürde ilk olarak Shiu ve Ahmad (1989) ile Tsai ve Lenz (1989) tarafından AX = XB olarak ifade edilen eşitliğin, X için çözülmesi gerekmektedir. Şekil 4.8'de gösterildiği üzere A matrisi, kalibrasyon modelinin hareketini ifade etmekte ve T_S^P dönüşümleri sayesinde bulunabilmektedir. B matrisi, robot uç noktasının hareketini ifade etmekte ve T_{TCP}^B dönüşümleri sayesinde hesaplanmaktadır. X matrisi ise hesaplanmak istenen ve kalibrasyon modelinden robot ucuna olan dönüşüm T_P^{TCP}'ye karşılık gelmektedir. Burada görüleceği üzere AX = XB eşitliği mevcuttur.

Kalibrasyon bundan sonra, birkaç robot hareketi için AX = XB eşitliğinin kullanılarak X matrisinin hesaplanmasına dayanmaktadır. Elbette eşsiz bir dönüşüm elde edebilmek için; en az iki, ortalama olarak da on ile on beş arasında farklı robot hareketi gerçekleştirerek (A,B) çiftlerinin elde edilmesi gerekmektedir. Literatürde bu eşitliğin çözümü için birçok yöntem (çözücü) bulunmaktadır (Park ve Martin, 1994; Daniilidis, 1999). Bu tez çalışmasında bu çözümü elde edebilmek için Daniilidis (1999) tarafından önerilen yöntem kullanılmaktadır. Bu yöntemin implementasyonu, aynı zamanda MoveIt yazılımına ait el-göz kalibrasyon eklentisinde de yer almaktadır (MoveIt, 2020). Bu yöntem

takip edilerek birkaç farklı robot hareketi için hesaplanan T_P^{TCP} dönüşümü de, son olarak Denklem (4.11)'e göre T_S^B'yi bulmak için kullanılmaktadır. T_P^{TCP} dönüşümünün MoveIt üzerinde bulunmasına dair süreç Şekil 4.9'da gösterilmektedir. Bu sürecin sonunda elde edilen T_S^B dönüşümü kullanılarak artık insana ve robota ait kapsülleri tek bir koordinat sistemine göre, yani robot taban koordinat sistemine (O_B) göre, ifade etmek mümkün olmaktadır. Böylece minimum uzaklık hesaplamalarında kapsüller için referans koordinat sistemi olarak robot taban koordinat sistemi dikkate alınmaktadır.



Şekil 4.8: AX = XB kalibrasyon problemi



Şekil 4.9: El-göz kalibrasyonu: T_P^{TCP} dönüşümünün hesaplanması

5. GÜVENLİ ETKİLEŞİM YÖNTEMİ

5.1 Minimum Uzaklık Hesaplama

Önerilen sisteme göre insan ve robot, insan-robot etkileşiminin olduğu bir ortamda kapsüllerle temsil edilmektedir. Bu nedenle, minimum uzaklığı elde etmek için bu insan-robot kapsül çiftleri arasında hesaplamalar yapılmaktadır. Bu hesaplama sonucunda ise her bir kapsül çifti arasındaki minimum uzaklıklar ve kapsül çiftleri üzerinde minimum uzaklığın görüldüğü birbirine en yakın noktalar bulunmaktadır.

5.1.1 GJK uzaklık algoritması

Minimum uzaklık hesaplaması, ilk olarak Gilbert vd. (1988) tarafından ortaya koyulan Gilbert-Johnson-Keerthi (GJK) uzaklık algoritmasının genişletilmiş versiyonuna dayanmaktadır (Gilbert ve Foo, 1990). GJK algoritması, dışbükey politoplar arasındaki uzaklık hesaplamalarında sıklıkla kullanılan simpleks tabanlı bir algoritmadır (Han vd., 2018). Önerilen sistemde GJK algoritmasının uygulanmasının nedeni, geometrik temsil için kapsül gibi dışbükey nesnelerin kullanılması ve algoritmanın dışbükey nesneler için basit ve hızlı çözümler elde etmeye uygun olmasıdır. Bu durumda GJK uzaklık algoritması, sırasıyla robot ve insan kapsülleri olarak belirlenen iki dışbükey küme C_i ve $C_j^{'}$ arasındaki minimum Öklid uzaklığını hesaplamak için kullanılmaktadır. Bir kapsül çifti arasındaki minimum uzaklık $dist_{min}(C_i, C_j^{'})$ ise Denklem (5.1)'deki gibi ifade edilmektedir.

$$dist_{min}(C_i, C_j^{'}) = \min \left\{ \left\| c_i - c_j^{'} \right\| : \forall c_i \in C_i, \forall c_j^{'} \in C_j^{'} \right\} \tag{5.1}$$

Algoritma, minimum Öklid uzaklığının yanı sıra $c_{imin} \in C_i$ ve $c_{j_{min}}^{'} \in C_j^{'}$ olmak üzere birbirine en yakın iki noktayı da döndürebilmektedir. Birbirine en yakın noktalar Denklem (5.2)'deki eşitliği sağlamaktadır.

$$\left\| c_{imin} - c_{j_{min}}^{'} \right\| = dist_{min}(C_i, C_j^{'}) \tag{5.2}$$

Uzaklık algoritması basit bir şekilde Denklem (5.3)'te ifade edilen Minkowski farkını kullanmakta ve elde edilen dışbükey $C_{i,j}$ kümesi ile orijin arasındaki mesafeyi hesaplamaktadır.

$$C_{i,j} = C_i \ominus C_j^{'} \tag{5.3}$$

Tüm Minkowski farkının hesaplanması pratik olmadığı için, GJK algoritması bir destek fonksiyonu kullanarak yinelemeli bir şekilde simpleks oluşturmaktadır. Minkowski farkı $C_{i,j}$ için destek fonksiyonu, Denklem (5.4)'te olduğu gibi tanımlanabilmektedir.

$$S_{C_{i,j}}(\mathbf{v}) = S_{C_i \ominus C_j^{'}}(\mathbf{v}) = S_{C_i}(\mathbf{v}) - S_{C_j^{'}}(-\mathbf{v}) \tag{5.4}$$

Bu eşitlikte $\mathbf{v}$ bir vektördür ve $C_{i,j}$ dışbükey kümesinin destek fonksiyonu $S_{C_{i,j}}$, $\mathbf{v}$ vektörünü $C_{i,j}$ kümesindeki bir destek noktasına eşlemektedir. Destek noktası $\mathbf{w}$ ise, $C_{i,j}$ kümesinin sınırında bulunmakta ve Denklem (5.5)'i karşılamaktadır.

$$\mathbf{v} \cdot S_{C_{i,j}}(\mathbf{v}) = \max\{\mathbf{v} \cdot \mathbf{c_{i,j}} : \mathbf{c_{i,j}} \in C_{i,j}\} \tag{5.5}$$

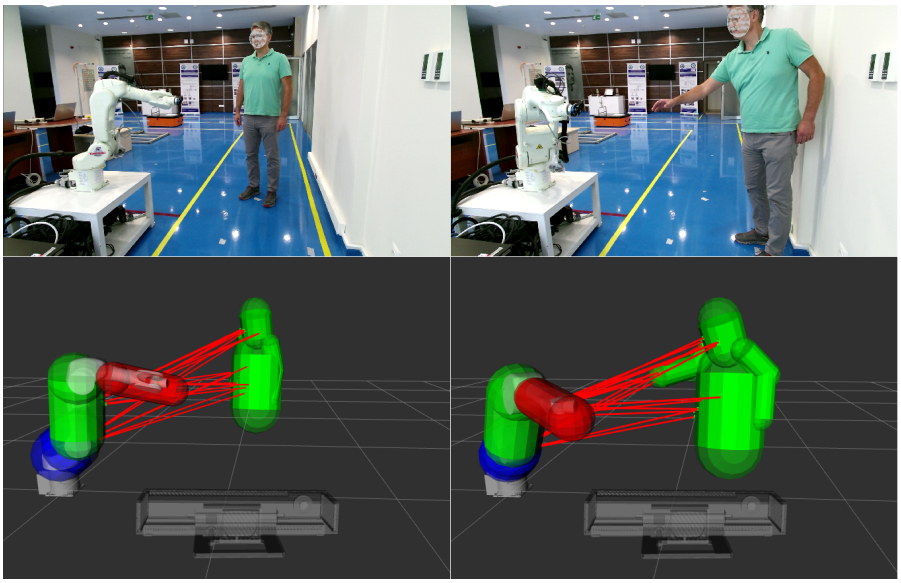
Destek noktası $\mathbf{w}$, algoritmanın her yinelemesinde simpleksin nokta kümesini W oluşturmak için kullanılmaktadır. Algoritma (k iterasyonlu), boş bir nokta kümesi $W_0 = \emptyset$ ve rastgele bir başlangıç noktası $\mathbf{v_0} \in C_{i,j}$ ile başlamaktadır. Nokta kümesi W, orijini içerdiğinde veya $-\mathbf{v_k}$ yönünde $\mathbf{v_k}$'nin kendisinden daha yeni bir destek noktası $\mathbf{w_k}$ bulmak mümkün olmadığında da sona ermektedir. Algoritma, nokta kümesi W_k orijini içermiyorsa, uzaklık olarak $\|\mathbf{v_k}\|$ değerini döndürmektedir. GJK uzaklık hesaplama algoritmasının özeti, Algoritma 1'de verilmektedir. Algoritmanın sonunda, en yakın nokta çifti c_{imin} ve $c_{j_{min}}^{'}$ de Bergen (1999) tarafından açıklanan yöntem izlenerek hesaplanabilmektedir.

Çoğunlukla geometrik modeller arasında çarpışma tespiti ve uzaklık hesaplamaları yapmak için kullanılan "Flexible Collision Library (FCL)", GJK algoritmasının implementasyonunu içermektedir (FCL, 2020b). Hesaplama için kapsüller tanımlanırken FCL'deki "Capsule" sınıfı kullanılmaktadır. Kapsüle ait tanımlamalar, yarıçap değeri ve z-ekseni yönündeki uzunluk değerinden oluşmaktadır. Bu tanımlar, robot ve insan kapsüllerinin her biri için ayrı ayrı yapılmaktadır. Kapsül, herhangi bir dönüşüm olmaksızın, z-ekseni boyunca yükselmektedir. Böylece alt yarım kürenin merkezi (yani sıfır noktası) küresel koordinat sisteminin orijininde olmaktadır. FCL 0.6.0 kütüphanesinde

yer alan GJK çözücü, girdi olarak aynı zamanda kapsüllerin dönüşümlerini de kullanmaktadır. Bölüm 4'te açıklandığı gibi insan uzuvları ve robot bağlantıları için hesaplanan dönüşümler, bu aşamada çözücü için girdi olarak kullanılmaktadır. Çıktı olarak ise, iki kapsül çifti arasındaki uzaklık ile kapsüller üzerindeki en yakın noktalar elde edilmektedir. İnsan ve robot kapsülleri arasındaki minimum uzaklık hesaplamasına dair gösterim Şekil 5.1'de yer almaktadır.

Algorithm 1 GJK uzaklık algoritması

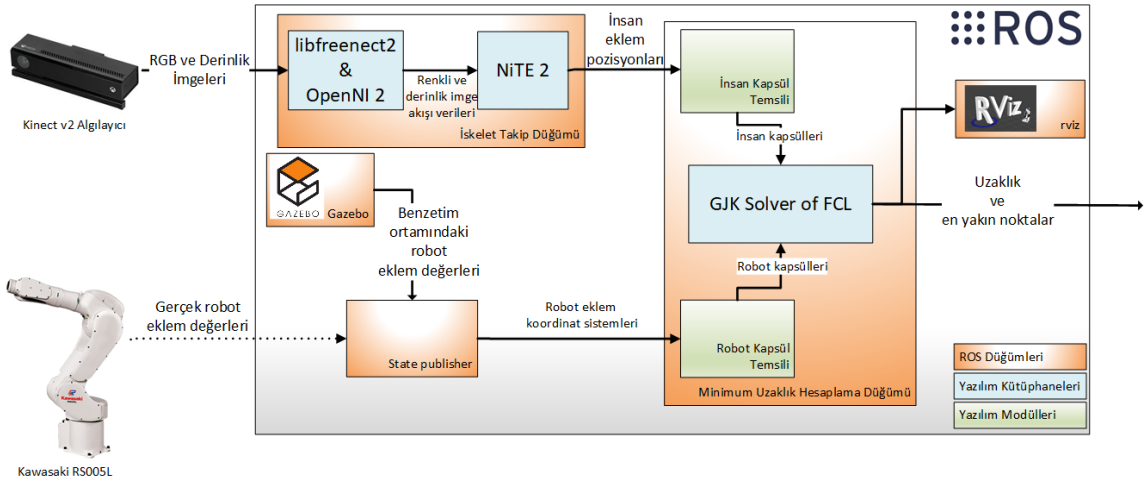
$k = 0$
$W_k = \emptyset$
$\mathbf{v_k} \leftarrow$ rastgele nokta $\in C_{i,j}$
for $iterasyon\ k$ **do**
  $-\mathbf{v_k}$ yönünde $\mathbf{w_k}$ hesapla
  if $\frac{\mathbf{w_k} \cdot \mathbf{v_k}}{\|\mathbf{v_k}\|} - \|\mathbf{v_k}\| \geq esik\ degeri$ **then**
    $\|\mathbf{v_k}\|$ değerini döndür
  end if
  $W_k \leftarrow \mathbf{w_k}$
  $\mathbf{v_{k+1}} \leftarrow W_k$'nın dışbükey gövdesinde orijine en yakın nokta
  $W_{k+1} \leftarrow W_k'$ ▷ $W_k' \subset W_k$ ve $\mathbf{v_k} \in W_k'$'nın dışbükey gövdesi
  $k = k + 1$
end for



Şekil 5.1: İnsan ve robot kapsülleri arasındaki minimum uzaklık hesaplamasının Rviz'de gösterimi

5.1.2 Sistem tasarımı

Bu tez çalışmasında önerilen sistem, Şekil 5.2'de görüldüğü gibi ROS üzerinde uygulanmaktadır (ROS, 2020). RGB-D türünden algılayıcı içeren iskelet takip algoritması, insanı izlemek ve onu kapsüllerle temsil etmek için kullanılmaktadır. İskelet takibi, libfreenect2, OpenNI 2 ve NiTE 2 adlı açık kaynak yazılımları ve kütüphaneleri kullanılarak gerçekleştirilmektedir (Xiang vd., 2016; OpenNI2, 2020). Bu yazılımlar içinden libfreenect2, Microsoft Kinect v2 için tanımlı bir kullanıcı sürücüsüdür. Bu sürücü, çeşitli derinlik kameralarıyla çalışmak için genel bir çerçeve sağlayan yazılım kitaplığı OpenNI ile entegre olarak, renk/derinlik verilerini yakalamak için kullanılmaktadır. NITE2 ise, insan algılama, duruş tahmini ve el izleme özelliklerine sahip ara katman yazılımıdır ve bu çalışmada iskelet takibi için kullanılmaktadır. Son yıllarda ticari veya ücretsiz olarak temin edilebilen cubemos, nuitrack, LIPS vb. gibi birçok etkili iskelet takip yazılım geliştirme kiti de bu işlem için olası alternatifler arasındadır (Cubemos, 2020; Nuitrack, 2020; LIPS, 2020).



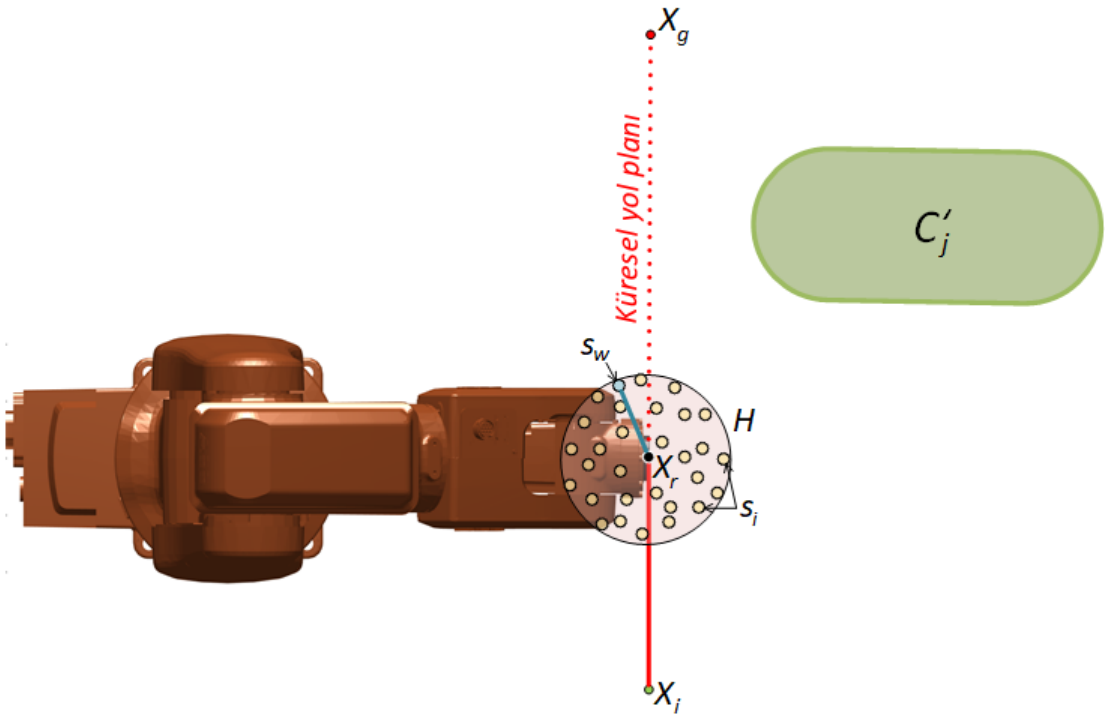
Şekil 5.2: Önerilen sistemin uygulama şeması

Önerilen sisteme göre, robot kapsülleri için robotun anlık konfigürasyonuna ihtiyaç duyulmaktadır. Bu sebeple robot eklem değerleri, endüstriyel robotun denetleyicisinden doğrudan alınabilmekte veya test sırasında emniyeti sağlamak için robot modelinin yüklendiği Gazebo simülatöründen okunabilmektedir (Koenig ve Howard, 2004). Burada "state publisher", girdi olarak robotun eklem değerlerini alan ve robotun önceden yüklenmiş modellerini kullanarak dönüşüm matrisleriyle ilişkili eklem koordinat sistemlerini üreten bir ROS düğümüdür.

Minimum uzaklık hesaplama düğümüne ait implementasyon, Bölüm 4 ve Bölüm 5.1'te sunulan yöntemlere dayanmaktadır. Buna göre, ilk olarak insan ve robota ait kapsüllerin konum ve yönelimleri üretilmektedir. Ardından, kapsüller için FCL kütüphanesindeki GJK çözücüden yararlanılmakta ve emniyetli insan-robot işbirliği görevlerinde kullanılmasını sağlamak için kapsüller arasındaki minimum uzaklıklar ve en yakın noktalar hesaplanarak yayınlanmaktadır. ROS yapısı altında üç boyutlu görselleştirici olarak yer alan Rviz de, deneylerin sonuçlarını görselleştirmek için kullanılmaktadır.

5.2 Çarpışmasız Yerel Yol Planlama

Bu bölümde, çarpışmasız yol planlama için önerilen optimizasyon tabanlı iteratif yöntem açıklanmaktadır. Bu yönteme göre, robotun uç konumunda ($\mathbf{X_r}$) merkezlenen ve Şekil 5.3'te gösterildiği gibi her iterasyonda onunla birlikte hareket eden varsayımsal bir küre (H) olduğu varsayılmaktadır. Robot ucuna ait yol planını oluşturacak ve robotun bir sonraki iterasyonda alacağı Kartezyen pozisyon için aday olacak örnek noktalar ($s_i, i \in [1, N_s]$), küre yüzeyinde tekdüze (ing. Uniform) dağılım ile rastgele seçilmektedir. Tüm örnek noktalar arasından yerel yol planını oluşturacak yol noktası (s_w) ise, optimizasyon probleminin çözümü ile elde edilmektedir. Bu süreç robot ucu hedef pozisyonuna erişinceye kadar iteratif olarak tekrarlanmaktadır.



Şekil 5.3: Önerilen yöntemin genel görünümü

Her bir iterasyonda örneklenen noktalar için robot konfigürasyonları ters kinematik ile elde edilmekte ve ardından robot ile insan kapsülleri arasındaki uzaklıklar Bölüm 5.1'te tanımlandığı gibi hesaplanmaktadır. Daha sonra, hesaplanan uzaklıklara dayalı olarak çarpışmasız bir yol oluşturmak için optimizasyon problemi çözülmektedir. Optimizasyon problemi için tanımlanan maliyet fonksiyonu, robot ucunu hedefe düzgün (ing. Smooth) bir şekilde hareket ettirecek ve aynı anda robotu engellerden uzak tutacak kısımlardan oluşmaktadır. Takip eden alt bölümlerde, yöntem ayrıntılı olarak açıklanmaktadır.

5.2.1 Tekdüze dağılımlı rastgele nokta örnekleme

Bu çalışmada, robot engellerden kaçınırken robot ucunun Kartezyen koordinatlarla belirlenen hedef konuma ulaşması için iteratif olarak yerel bir yol planı oluşturulmaktadır. Önerilen yaklaşıma göre, r_H yarıçaplı varsayımsal bir H küresi, her iterasyonda robot ucunun mevcut konumunda $X_r = \begin{bmatrix} x_r & y_r & z_r \end{bmatrix}^T$ ortalanmaktadır. Varsayımsal küre, Denklem (5.6)'deki gibi ifade edilmektedir.

$$H \triangleq \{X_r, r_H\}, X_r \in \mathbb{R}^3, r_H \in \mathbb{R} \tag{5.6}$$

Rastgele örnekler (s_i) ise, varsayımsal küre yüzeyinde tekdüze dağılımla belirlenmekte ve bu örneklerin kümesi S_t, Denklem (5.7)'deki gibi ifade edilmektedir.

$$S_t = \{s_i : i = 1, 2, \ldots, N_s\}, t \in \begin{bmatrix} t_0, & t_0 + \Delta t, & \ldots, & t_0 + (k-1)\Delta t \end{bmatrix} \tag{5.7}$$

Burada t bir sonraki yol noktasının hesaplanmasına dair zamanı, t_0 yerel yol planlamasının başladığı zamanı, k pozitif bir tam sayı olarak iterasyon sayısını, N_s toplam örnek sayısını ve Δt iki ardışık yol noktası arasındaki hareket için geçen zamanı ifade etmektedir.

5.2.1.1 Varsayımsal küre yüzeyinde örnekleme

Bir küre yüzeyindeki herhangi bir nokta, Şekil 5.4'te gösterildiği gibi küresel koordinatları temsil eden radyal mesafe r_h, kutup açısı $\theta_i \in [0, \pi]$ ve azimut açısı $\phi_i \in [0, 2\pi)$ kullanılarak ifade edilebilmektedir. Bu noktalar, daha sonra Denklem (5.8), Denklem (5.9) ve Denklem (5.10)'daki eşitlikler kullanılarak Kartezyen koordinatlarla ifade edilmektedir.

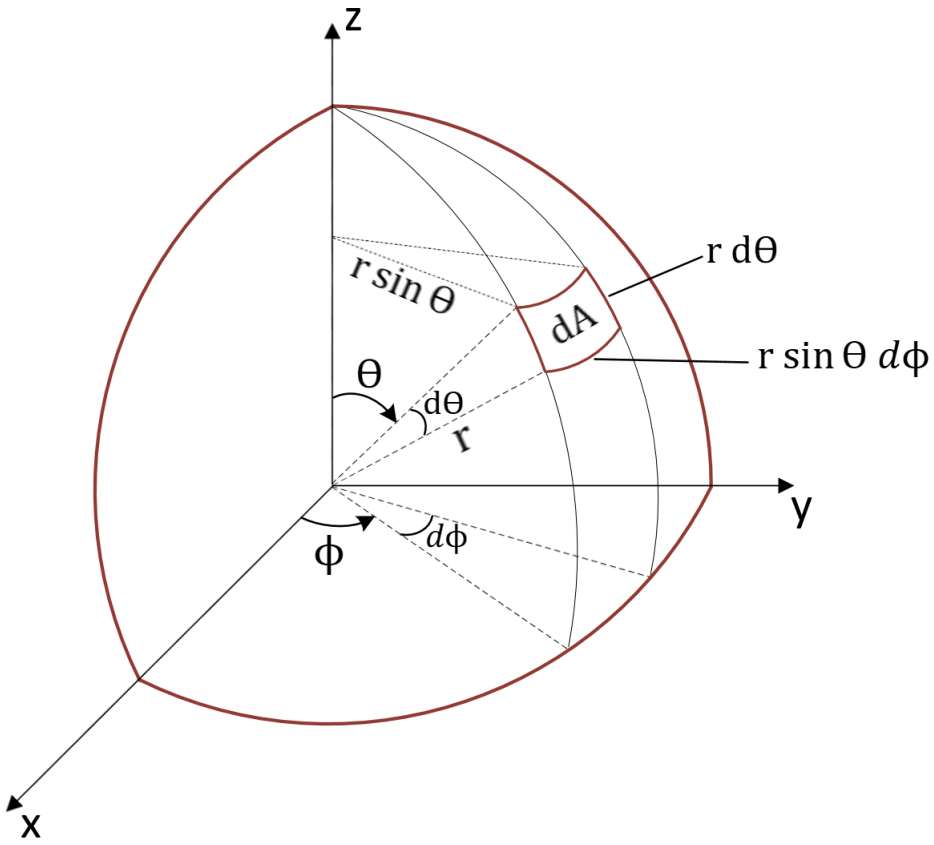
$$x_i = r_h \sin\theta_i \cos\phi_i \tag{5.8}$$

$$y_i = r_h \sin\theta_i \sin\phi_i \tag{5.9}$$

$$z_i = r_h \cos\theta_i \tag{5.10}$$

Bu Kartezyen koordinatlar, daha sonra s_i örnek noktalarını Denklem (5.11)'deki gibi tanımlamak için kullanılmaktadır.

$$s_i = X_r + \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{5.11}$$



Şekil 5.4: Küresel koordinatlar ve diferansiyel yüzey alanı elemanı

Bir küre yüzeyinde rastgele noktalar seçerken, $\theta_i \in [0, \pi]$ ve $\phi_i \in [0, 2\pi)$ aralıkları için küresel koordinatların tekdüze dağılımları kullanılabilmektedir. Ancak bu durum,

diferansiyel yüzey alanı elemanı $dA = r^2 \sin\theta d\theta d\phi$ ile tanımlandığından ve diferansiyel katı açısı $d\Omega = \sin\theta d\theta d\phi$ olduğundan, örnek noktaların kutup noktalarında ($\theta = 0$ ve $\theta = \pi$) yoğunlaşmasına neden olmaktadır. Bu sebeple, bu sorunu önlemek ve küre yüzeyi üzerinde doğru bir tekdüze dağılım elde etmek için Khungrun (2015) tarafından açıklandığı gibi ters dönüşüm örneklemesi uygulanmaktadır. Buna göre, alanı 4π olan bir birim küre yüzeyinde tekdüze dağılım için, olasılık yoğunluğunun $p(x) = 1/4\pi$ şeklinde sabit olması gerekmektedir. Aynı zamanda, bir noktanın diferansiyel yüzey alanında bulunma olasılığı ise $p(x)dA$ ile ifade edilebilir. Bu durumda Denklem (5.12), küre üzerinde tekdüze bir dağılıma eşleyen olasılık yoğunluk fonksiyonunu $p(\phi, \theta)$ bulmak için kullanılabilir.

$$p(x)dA = p(\phi, \theta)d\phi d\theta \tag{5.12}$$

Yüzey elemanı $dA = \sin\theta d\theta d\phi$ olduğundan, olasılık yoğunluk fonksiyonu bu sayede Denklem (5.13)'teki gibi ifade edilmektedir.

$$p(\phi, \theta) = \frac{\sin\theta}{4\pi} \tag{5.13}$$

Küresel koordinatlar θ ve ϕ, $p(\phi, \theta) = p(\phi)p(\theta)$ olarak birbirinden bağımsız olacak şekilde örneklenmek istendiğinde de olasılık yoğunluk fonksiyonları Denklem (5.14) ve Denklem (5.15)'teki gibi ifade edilmektedir.

$$p\left(\phi\right) = \int_0^{\pi} p\left(\phi, \theta\right) d\theta = \frac{1}{2\pi} \tag{5.14}$$

$$p\left(\theta\right) = \int_0^{2\pi} p\left(\phi, \theta\right) d\phi = \frac{\sin\theta}{2} \tag{5.15}$$

Denklem (5.14) ve Denklem (5.15)'te görüldüğü üzere, olasılık yoğunluğu $p(\theta)\ \sin\theta$ ile değişirken, $p(\phi)$ sabit bir dağılıma sahip olduğundan ϕ tekdüze bir şekilde dağılmaktadır. Bu nedenle, θ ters dönüşüm örneklemesi kullanılarak örneklenebilir.

Bu yönteme göre, ϕ ve θ değerlerini üretmek için $P(\phi)$ ve $P(\theta)$ kümülatif dağılım fonksiyonlarına ihtiyaç duyulmaktadır. Denklem (5.16) ve Denklem (5.17)'de sırasıyla ϕ ve θ için kümülatif dağılım fonksiyonları verilmektedir.

$$P\left(\phi\right) = \int_0^{\phi} p\left(\phi\right) d\phi = \frac{\phi}{2\pi} \tag{5.16}$$

$$P\left(\theta\right) = \int_0^{\theta} p\left(\theta\right) d\theta = \frac{1 - \cos\theta}{2} \tag{5.17}$$

Her ikisi de $[0, 1]$ aralığında tekdüze olan iki rastgele değişken u ve v, $P\left(\phi\right) = u$ ve $P\left(\theta\right) = v$ olacak şekilde seçilirse, kümülatif dağılım fonksiyonlarının ters fonksiyonları bu durumda Denklem (5.18) ve Denklem (5.19)'daki eşitlikleri vermektedir.

$$\phi_i = 2\pi u_i \tag{5.18}$$

$$\theta_i = \cos^{-1}\left(1 - 2v_i\right) \tag{5.19}$$

Sonuç olarak, ters dönüşüm örneklemesi sayesinde tekdüze dağılıma sahip küresel koordinatlar (5.18) ve Denklem (5.19)'deki eşitlikler kullanılarak hesaplanabilmektedir. Burada $u = \{u_i : i = 1, 2, \ldots, N_s\}$ ve $v = \{v_i : i = 1, 2, \ldots, N_s\}$, $[0, 1]$ aralığında tekdüze şekilde örneklenen iki bağımsız rastgele değişkeni ifade etmektedir. Bunun ardından, Denklem (5.8), Denklem (5.9) ve Denklem (5.10)'daki eşitlikler de tekdüze dağılmış noktaların Kartezyen koordinatlarını hesaplamak için kullanılabilmektedir.

5.2.1.2 Varsayımsal yarım küre yüzeyinde örnekleme

Önerilen yönteme göre, insan ve robot arasındaki uzaklığın nispeten büyük olduğu koşullarda da bir yarım küre $\widehat{H}$ yüzeyinde tekdüze örnekleme gerçekleştirilmektedir. Varsayımsal yarım küre, Denklem (5.20)'deki gibi ifade edilmektedir.

$$\widehat{H} \triangleq \left\{X_r, r_H, \mathbf{n}\right\}, X_r \in \mathbb{R}^3, r_H \in \mathbb{R}, \mathbf{n} \in \mathbb{R}^3 \tag{5.20}$$

Bu eşitlikte $\mathbf{n} \in \mathbb{R}^3$, yarım kürenin z-ekseni doğrultusunda orijinde tanımlanan birim normal vektördür.

Yarım küre yüzeyinde, yine ters dönüşüme dayanan örnekleme, küre yüzeyinde gerçekleştirilene benzemektedir. Ancak, yarım küre yüzeyindeki $\widehat{\theta}_i \in [0, \pi/2)$ ve $\widehat{\phi}_i \in [0, 2\pi)$ küresel koordinatları bu sefer Denklem (5.21) ve Denklem (5.22)'deki eşitlikler kullanılarak hesaplanmaktadır.

$$\widehat{\phi}_i = 2\pi u_i \tag{5.21}$$

$$\widehat{\theta}_i = \cos^{-1}(1 - v_i) \tag{5.22}$$

Burada da $u = \{u_i : i = 1, 2, \ldots, N_s\}$ ve $v = \{v_i : i = 1, 2, \ldots, N_s\}$, $[0, 1]$ aralığında tekdüze şekilde örneklenen iki bağımsız rastgele değişkeni ifade etmektedir. Yarım küre yüzeyindeki örnek noktalara ait Kartezyen koordinatlar daha sonra Denklem (5.23), Denklem (5.24) ve Denklem (5.25)'teki eşitlikler kullanılarak hesaplanmaktadır.

$$\widehat{x}_i = r_h \sin\widehat{\theta}_i \cos\widehat{\phi}_i \tag{5.23}$$

$$\widehat{y}_i = r_h \sin\widehat{\theta}_i \sin\widehat{\phi}_i \tag{5.24}$$

$$\widehat{z}_i = r_h \cos\widehat{\theta}_i \tag{5.25}$$

Önerilen yönteme göre, yarım küre yüzeyinde örneklenen noktaların ($\widehat{x}_i$, $\widehat{y}_i$, $\widehat{z}_i$) aynı zamanda hedef yöne doğru yönlendirilmelerini sağlamak için dönüşüm uygulanması gerekmektedir. Bunu gerçekleştirebilmek için, Şekil 5.5'te gösterildiği gibi aynı zamanda dönme merkezi de olan yarım kürenin orijininde ($O_{\widehat{H}}$) iki vektör tanımlanmaktadır: yarım kürenin z-ekseni yönünde birim normal vektörü $\mathbf{n} \in \mathbb{R}^3$ ve X_g hedef konumuna doğru birim yön vektörü $\mathbf{d} \in \mathbb{R}^3$. Bunun yanında, dönüşümden önceki ve sonraki koordinat sistemlerini temsil etmesi için sırasıyla iki dönüşüm matrisi T_B ve T_A da tanımlanmaktadır. Bu iki dönüşüm matrisini de hesaplamak için taban vektörleri bulunmaktadır. Bu taban vektörlerinden ilki, aynı zamanda rotasyon ekseni $\mathbf{r} \in \mathbb{R}^3$ olarak da ifade edilmekte ve Denklem (5.26)'daki gibi hesaplanmaktadır.

$$\mathbf{r} = \mathbf{n} \times \mathbf{d} \tag{5.26}$$

Diğer iki taban vektörü de daha sonra bu rotasyon ekseni kullanılarak Denklem (5.27) ve Denklem (5.28)'deki eşitlikler ile bulunmaktadır.

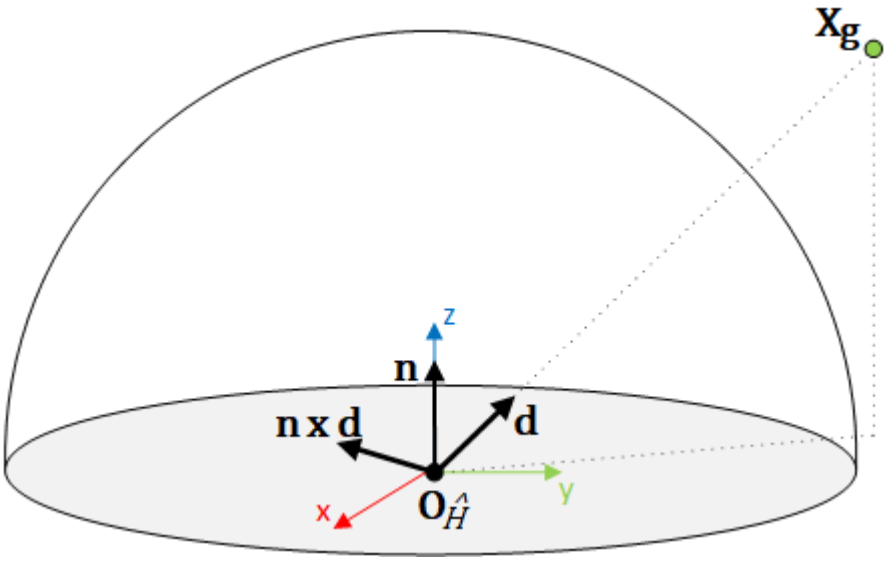
$$\mathbf{s} = \mathbf{r} \times \mathbf{d} \tag{5.27}$$

$$\mathbf{t} = \mathbf{r} \times \mathbf{n} \tag{5.28}$$

Bu taban vektörleri, sonuç olarak dönüşümden önceki ve sonraki koordinat sistemlerini temsil eden dönüşüm matrisleri T_B ve T_A'yı oluşturmak için Denklem (5.29) ve Denklem (5.30)'daki gibi kullanılmaktadır.

$$T_A = \begin{bmatrix} \mathbf{r} & \mathbf{s} & \mathbf{d} \end{bmatrix}^T \tag{5.29}$$

$$T_B = \begin{bmatrix} \mathbf{r} & \mathbf{t} & \mathbf{n} \end{bmatrix}^T \tag{5.30}$$



Şekil 5.5: Yarım küre yüzeyindeki noktaların dönüşümü

Bu yaklaşım, elbette yalnızca birim vektör **d**'nin birim normal vektörü **n**'ye paralel olmaması durumunda geçerlidir. Aksi takdirde, birim vektörlerin vektörel çarpımı sıfıra eşit olacaktır. Böyle bir durumda rotasyon ekseni **r**'yi bulmak için, hem **n**, hem de **d** birim vektörlerine dik olan bir **c** birim vektörü tanımlanmaktadır. Birim vektör **c** böyle bir durumda en az hizalanmış (ing. Least aligned) taban vektörü olarak seçilebilir. Örneğin, $\mathbf{n} = \mathbf{d} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ için $\mathbf{c} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ olarak belirlenebilir. Bu durumda, rotasyon ekseni **r** ise Denklem (5.31)'deki gibi hesaplanmaktadır.

$$\mathbf{r} = \mathbf{n} \times \mathbf{c} \tag{5.31}$$

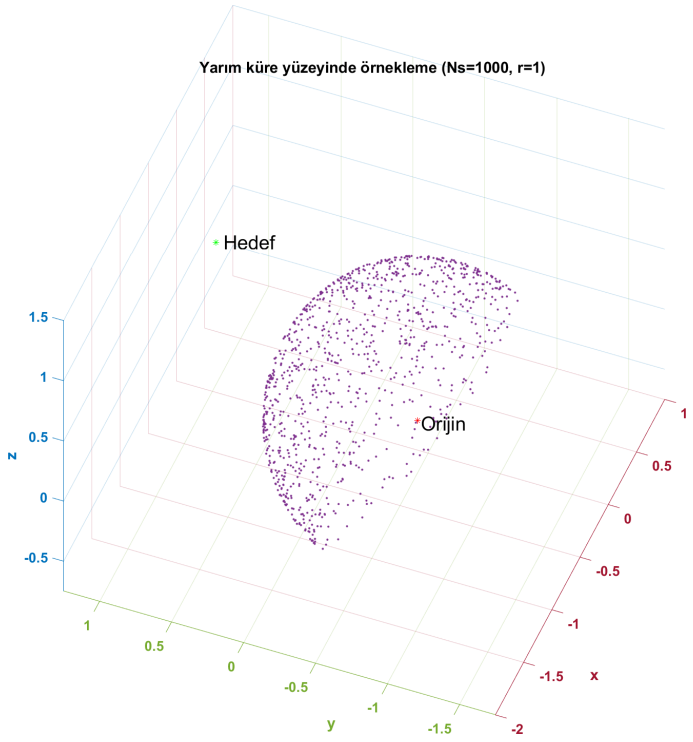
Yarım küre yüzeyindeki örnek noktaları ($\widehat{x}_i$, $\widehat{y}_i$, $\widehat{z}_i$) hedefe doğru yönlenecek şekilde döndürmek için gerekli olan dönüşüm ise sonuç olarak $T_R = {T_A}^{-1} T_B$ eşitliği kullanılarak hesaplanmaktadır. Döndürülmüş örnek noktalar ($\widehat{x}'_i$, $\widehat{y}'_i$, $\widehat{z}'_i$), daha sonra Denklem (5.32)'deki gibi ötelenerek örnek noktalara ait Kartezyen koordinatları hesaplamada kullanılmaktadır.

$$\widehat{s}_i = X_r + \begin{bmatrix} \widehat{x}'_i & \widehat{y}'_i & \widehat{z}'_i \end{bmatrix}^T \tag{5.32}$$

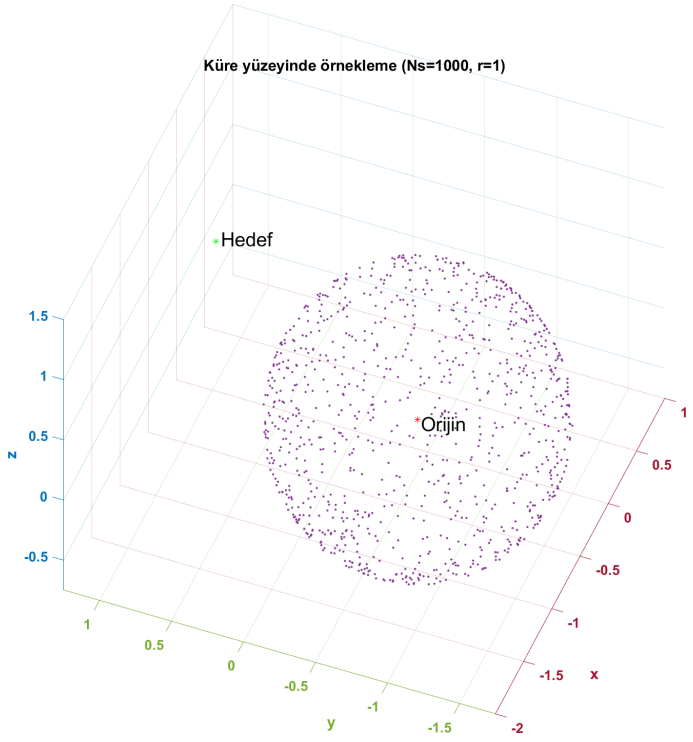
Bir yarım küre yüzeyi ve bir küre yüzeyi üzerinde gerçekleştirilen örneklemelere ait gösterim Şekil 5.6'da yer almaktadır.

Yarım küre yüzeyinde yapılan örneklemenin amaçları ise aşağıdaki gibi özetlenmektedir:

- İnsan ve robot arasındaki uzaklık izin verilen sınırın üzerinde olduğunda, robotun ucunu hedefe yaklaştıracak örnek noktaları oluşturmak daha mantıklıdır. Böyle bir durumda motivasyon, engelden kaçınmak yerine hedefe ulaşmak olacaktır.
- Emniyetli bir ayırma mesafesi zaten mevcut olduğundan ve örnekleme yoğunluğu hedef yönünde olduğundan robotu hedeften uzaklaştıracak olasılıklar elenerek azalacaktır.



(a)



(b)

Şekil 5.6: Önerilen yönteme göre nokta örneklenmesi: (a) birim yarım küre yüzeyinde (b) birim küre yüzeyinde

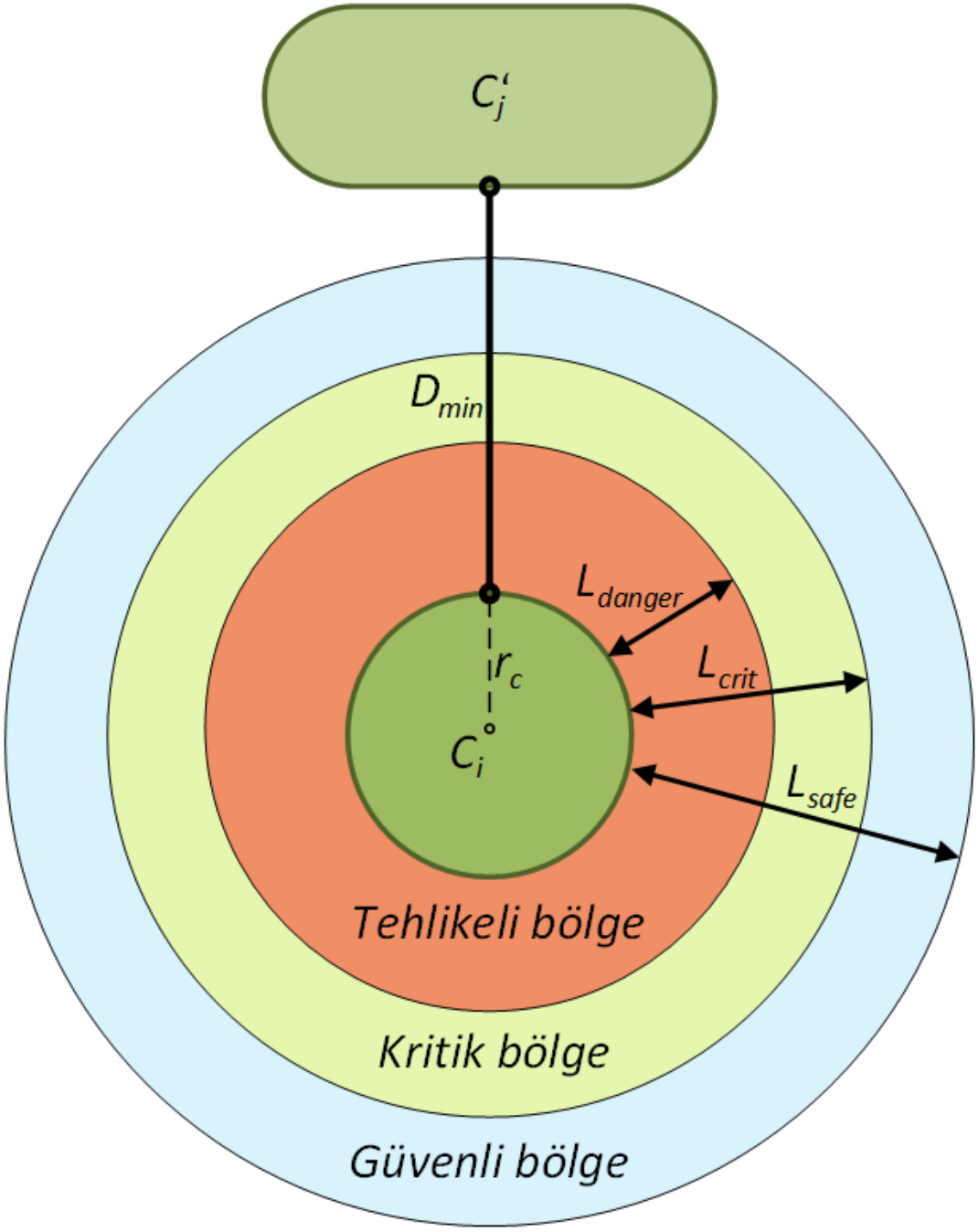
5.2.2 Emniyetli mesafe parametreleri

Önerilen yönteme göre, robot manipülatörün engellerden uzak kaldığı sürece önceden tanımlanmış küresel yolu takip etmesi hedeflenmektedir. Robot ve engeller arasındaki uzaklık bir eşik değerinin altına düştüğünde ise, yerel yol planlayıcı devreye girmektedir. Buna göre, robot uç noktası varsayımsal küre yüzeyindeki örnek noktalar aracılığıyla hedefe doğru hareket ettirilmektedir.

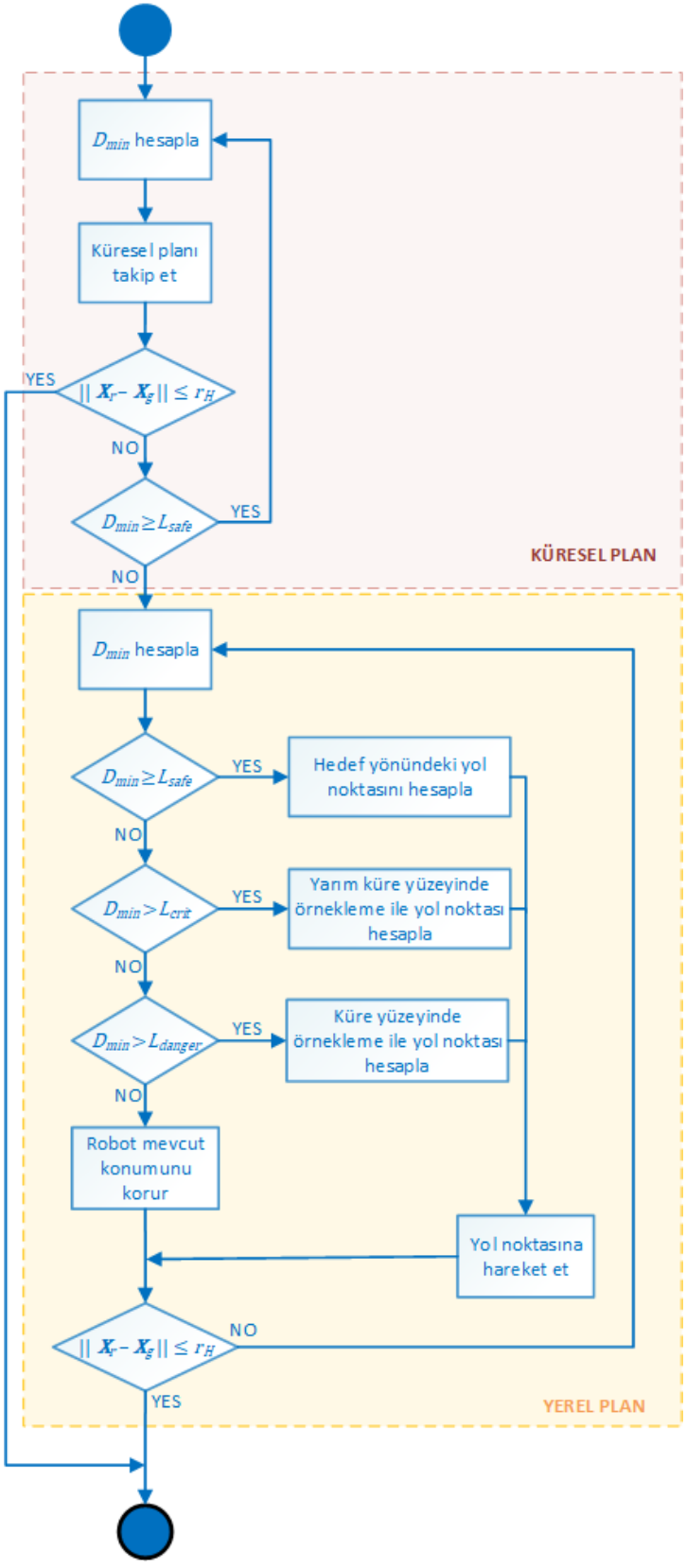
Robotun hedefe ulaşırken çalışma alanındaki insanlarla emniyetli bir şekilde etkileşime girebilmesi için emniyetli mesafe L_{safe}, kritik mesafe L_{crit} ve tehlikeli mesafe L_{danger} olmak üzere üç sınır parametresi tanımlanmaktadır. Bu parametrelerden L_{safe}, robot ile insan arasındaki emniyetli uzaklığı belirleyen sınırı ifade etmektedir. Robot ile insan arasındaki uzaklık L_{safe}'in altına düştüğünde, küresel plandan vazgeçilmekte ve yerel planlayıcı önerilen yönteme göre planlamayı gerçekleştirmektedir. L_{crit}, robot ile insan arasındaki izin verilen minimum uzaklıktır. Bu sınır parametreleri aynı zamanda varsayımsal küre yüzeyinde gerçekleştirilecek örneklemenin türünü de belirlemektedir. Son sınır parametresi L_{danger} ise, robotun insan ile etkileşime girebileceği en küçük mesafeyi belirtmekte ve optimizasyon probleminde kısıt olarak kullanılmaktadır. Bu sınır parametreleri Şekil 5.7'de gösterilmektedir.

Yerel planlamanın her iterasyon adımında, ilk olarak her bir robot ve insan kapsülü arasındaki minimum uzaklıklar $dist_{min}(C_i, C_j')$ hesaplanmaktadır. Hesaplanan minimum uzaklık $D_{min} = \min\left(dist_{min}(C_i, C_j')\right)$ değeri, $D_{min} \geq L_{safe}$ koşulunu sağlıyorsa robotun ucu hedefe yönlendirilmektedir. Bunun için, robot ucunun mevcut konumundan $X_r = \begin{bmatrix} x_r & y_r & z_r \end{bmatrix}$, hedef konumuna $X_g = \begin{bmatrix} x_g & y_g & z_g \end{bmatrix}$ doğru bir birim vektör $\mathbf{v_{rg}} \in \mathbb{R}^3$ tanımlanmaktadır. $D_{min} < L_{safe}$ durumunda ise, örnekleme D_{min} uzaklığına göre bir kürenin veya yarım kürenin yüzeyinde gerçekleştirilmektedir.

Küre yüzeyinde tekdüze örnekleme, hesaplanan uzaklık $L_{crit} \geq D_{min} > L_{danger}$ koşulunu sağladığında gerçekleştirilmektedir. Küresel koordinatlar bu durumda Denklem (5.18) ve Denklem (5.19) kullanılarak hesaplanmaktadır. Yarım küre yüzeyinde örnekleme ise, $L_{safe} > D_{min} > L_{crit}$ koşulu sağlandığında gerçekleştirilmektedir. Küresel koordinatlar ise bu sefer Denklem (5.21) ve Denklem (5.22) kullanılarak hesaplanmaktadır. Örnek noktaların Kartezyen koordinatları da daha sonra, Denklem (5.11) veya Denklem (5.32) kullanılarak bulunmaktadır. Örnekleme ve buna göre yol planlaması için izlenen prosedür Şekil 5.8'de gösterilmektedir.



Şekil 5.7: Emniyetli mesafe parametreleri: L_{safe}, L_{crit}, L_{danger}, insan kapsülü C_j', robot kapsülü C_i ve hesaplanan minimum uzaklık D_{min}



Şekil 5.8: Sınır parametrelerine göre yol planlama ve örnekleme

5.2.3 Yerel yol planlama algoritması

Bu tez çalışmasında robotun hareketini çarpışma olmadan gerçekleştirebilmesi için optimizasyona dayalı iteratif bir yaklaşım önerilmektedir. Yerel planlayıcının amacı, robot için hedef konuma çarpışmasız ve düzgün bir yol üretmektedir. Bu nedenle robot ucunun izleyeceği yol noktaları algoritmada çevrim içi olarak belirlenmektedir. Görevler çoğunlukla görev uzayında (yani Kartezyen uzayda) tanımlandığından, robot ucunun takip edeceği yol da aynı şekilde görev uzayında tanımlanmaktadır. Ancak endüstriyel robot manipülatör kontrolü çoğunlukla eklem uzayında gerçekleştirilmektedir. Bu nedenle, algoritma robot konfigürasyonunu tespit etmek için Jacobian tersini kullanmaktadır. Jacobian matrisi $J \in \mathbb{R}^{m \times n}$, eklem uzayındaki $q \in \mathbb{R}^n$ eklem açılarının değişim oranını, görev uzayında robotun uç konumunun $X \in \mathbb{R}^m$ değişim oranıyla ilişkilendirmek için kullanılmaktadır. Bu ilişki, Denklem (5.33)'teki gibi ifade edilmektedir.

$$dX = J\left(q\right) dq \tag{5.33}$$

Jacobian matrisinin, Denklem (5.34)'teki gibi tersine çevrilmesiyle birlikte, ters kinematik problemi çözülebilmektedir.

$$dq = J^{-1}\left(q\right) dX \tag{5.34}$$

Ancak Denklem (5.34)'teki eşitlik, Jacobian matrisi eksik ranklı olduğunda veya kare olmadığında çözülememektedir. Jacobian matrisi, eksik tahrikli manipülatörler ($n < m$) veya artık serbestlik derecesine sahip manipülatörler ($n > m$) için kare değildir veya tekil konfigürasyonlarda rank kaybetmektedir. Bu durumda ise ters kinematiği çözebilmek için, Denklem (5.35)'teki gibi Jacobian matrisinin Moore-Penrose sözde-tersi $J^+ \in \mathbb{R}^{n \times m}$ kullanılabilmektedir.

$$J^+ = \left(J^T J\right)^{-1} J^T \tag{5.35}$$

Jacobian matrisinin Moore-Penrose sözde-tersinin kullanıldığı ters kinematik hesabı ise Denklem (5.36)'daki eşitlik kullanılarak yapılmaktadır. Bu sayede, Denklem (5.33) için

minimum norma sahip bir ters çözüm bulunmakta ve görev uzayında tanımlanan görevi yerine getirebilecek eklem komutları üretilebilmektedir.

$$dq = J^{+}(q)\, dX \tag{5.36}$$

Önerilen yönteme göre, robot ile engel (yani insan) arasındaki uzaklık robotun hareketi esnasında sürekli olarak hesaplanmaktadır. Minimum uzaklık değeri, L_{safe}'in altına düştüğünde ise yerel planlayıcıya geçilmekte ve iterasyon başlamaktadır. Algoritma başlangıçta n serbestlik derecesine sahip manipülatörün mevcut konfigürasyonuna $q_r \in \mathbb{R}^n$ ve hedef konumunun Kartezyen koordinatlarına $X_g \in \mathbb{R}^3$ ihtiyaç duymaktadır. Ayrıca varsayımsal kürenin yarıçapı r_H ve küre yüzeyinde tekdüze dağılıma sahip olacak rastgele örnek noktalarının sayısı N_s de, algoritmanın kullandığı parametreler arasında yer almaktadır. Fark edileceği üzere, r_H aynı zamanda her iterasyon adımında robot ucunun yer değiştirme miktarını da belirlemektedir. Her bir iterasyon adımında eklem açılarındaki değişimi hesaplamak için ters Jacobian yöntemi kullanılacağından, r_H değerinin tutarlı bir doğrusal yakınlık için yeterince küçük belirlenmesi gerekmektedir. Benzer şekilde, N_s sayısının da çözünürlük ve hesaplama süresi arasındaki ödünleşim dikkate alınarak belirlenmesi gerekmektedir. Çok sayıda örnek, çözünürlüğü ve dolayısıyla olası hareket alternatiflerinin sayısını arttırmasına rağmen, uzun hesaplama süresini de beraberinde gerektirmektedir.

Yeniden planlamanın her iterasyonunda, önce robot ile engel arasındaki uzaklık kontrol edilmektedir. Hesaplanan uzaklık $D_{min} \geq L_{safe}$ koşulunu sağlıyorsa, robotun ucu doğrudan hedefe yönlendirilmektedir. Bunun için, mevcut uç konumundan hedef konuma doğru olan $\mathbf{v_{rg}}$ birim vektörü kullanılmaktadır. Bu durumda, robot ucunun pozisyon değişimi $dX_r = r_H \mathbf{v_{rg}}$ ile ifade edilmekte ve böylece uç noktası Kartezyen uzayda r_H kadar yer değiştirmektedir. Jacobian tersi ile ters kinematik hesabı yapılarak robot eklem açılarındaki değişimin hesaplanmasıyla da robot konfigürasyonunu güncelleyecek kontrol komutu elde edilmektedir. Elbette $L_{safe} > D_{min}$ olduğu durumda, robot ucunun hareket ettirilebileceği konumlar, küre veya yarım küre üzerinde örnekleme yapılarak belirlenmektedir.

Önerilen yöntemde, örnekleme sonucunda $i = 1, 2, \ldots, N_s$ için örnek Kartezyen koordinatları $X_{s_i} \in \mathbb{R}^3$ elde edilmektedir. Ardından, k'ncı iterasyon için $dX_{s_i}^k$ yer

değiştirmeleri, robotun ucunun mevcut konumu X_r^k ve örnek Kartezyen koordinatları $X_{s_i}^k$ kullanılarak Denklem (5.37)'ye göre hesaplanmaktadır.

$$dX_{s_i}^k = X_{s_i}^k - X_r^k \tag{5.37}$$

Her iterasyonda, robot ucunun hareket ettirileceği örnek pozisyonlar belirlendikten sonra ters kinematik problemi çözülmektedir. Ters kinematik çözümlerinin bir sonucu olarak; her bir örnek noktası için eklem açılarındaki değişim $dq_{s_i}^k$, her bir örnek noktası için hesaplanan yer değiştirme $dX_{s_i}^k$ kullanılarak yaklaşık olarak hesaplanmaktadır. Bu hesaplamanın sonuçları, $q_{s_i}^k$ örnek konfigürasyonlarını Denklem (5.38)'deki gibi belirlemek için kullanılmaktadır.

$$q_{s_i}^k = q_r^k + dq_{s_i}^k \tag{5.38}$$

Bu örnek konfigürasyonlar ile daha sonra robotun kapsülleri geçici olarak güncellenmekte ve insan ile robot arasındaki uzaklık hesaplamaları için kullanılmaktadır. Her bir örnek konfigürasyon için bulunan uzaklık değerleri de, robot ucunun hareket edeceği Kartezyen koordinatları X_w^k belirlemek için optimizasyon probleminin ağırlıklı maliyet fonksiyonunda kullanılmaktadır. Optimizasyon probleminin sonucunda robot ucunu belirtilen konuma hareket ettirecek eklem kontrol komutu q_w^k belirlendikten sonra, mevcut konfigürasyon ve robot uç konumu güncellenmekte ve iterasyon $\left\|X_g - X_r^k\right\| \leq r_H$ olana kadar devam etmektedir. Bu sayede robot, insan-robot etkileşiminin söz konusu olduğu ortamlarda engellerden kaçınarak hedefe yönlendirilmektedir. Önerilen algoritma, Algoritma 2'te yer almaktadır.

5.2.4 Optimizasyon problemi

Yerel planlayıcı, robotun her iterasyonda uç konumunu hareket ettireceği konumu seçmek için optimizasyonu kullanmaktadır. Her iterasyonda bir sonraki yol noktasının tanımlanması için takip edilen strateji aşağıdaki gibi özetlenmektedir:

- Robotun ucu hedef konuma yaklaşır.
- Robotun tüm uzuvları, hareket esnasında engellere emniyetli bir mesafede yer alır.
- Robotun ucu düzgün bir yol izler.
- Eklemler izin verilen sınırlar içinde kalır ve ayırma mesafesi kriterleri karşılanır.

Algorithm 2 Yerel yol planlama

Input: q_r, X_r, X_g, r_H, N_s

Output: q_w, X_w

$k \leftarrow 0$

do

D_{min} hesapla

if $D_{min} \geq L_{safe}$ **then**

$X_w^k \leftarrow r_H \mathbf{v_{rg}^k} + X_r$

else

if $D_{min} > L_{crit}$ **then**

$S^k \leftarrow \{\widehat{s}_i : i \in [1, N_s]\}$ ▷ $\widehat{s}_i$, Denklem 5.32 ile hesaplanır.

else if $D_{min} > L_{danger}$ **then**

$S^k \leftarrow \{s_i : i \in [1, N_s]\}$ ▷ s_i, Denklem 5.11 ile hesaplanır.

end if

for $i = 1$ to N_s **do**

$dX_{s_i}^k \leftarrow s_i - X_r$ ▷ Her örnek için robot ucunun yer değiştirmesi hesaplanır.

$dq_{s_i}^k \leftarrow J^+(q_r^k) dX_{s_i}^k$ ▷ Eklem açılarındaki değişim için ters kinematik çözülür.

$q_{s_i}^k \leftarrow q_r^k + dq_{s_i}^k$ ▷ Örnek konfigürasyonlar hesaplanır.

for $r = 1$ to n_r **do**

$T_{C_r}^{R0}$'yi hesapla ▷ $q_{s_i}^k$ kullanarak robot kapsülünün dönüşümü hesaplanır.

$C_r \leftarrow \left\{T_{C_r}^{R0}, h_C, r_C\right\}$

for $h = 1$ to n_h **do**

$C_h' \leftarrow \left\{T_{C_h'}^{R0}, h_C, r_C\right\}$

$dist_{r,h,i}^k \leftarrow dist_{min}(C_r, C_h')$

end for

end for

$H_{s_i}^k$'yı hesapla ▷ Denklem 5.39 ile maliyet hesaplanır.

end for

$q_w^k \leftarrow \min_{q_{s_i}^k} \quad H_{s_i}^k$ ▷ Optimizasyon problemi çözülür.

$X_w^k \leftarrow \mathcal{F}(q_w^k)$ ▷ İleri kinematik $\mathcal{F}(.)$ çözülür.

end if

$X_r \leftarrow X_w^k$

$k \leftarrow k + 1$

while $\|X_g - X_r\| \leq r_H$

$X_r \leftarrow X_g$

Buna göre, k'nci iterasyondaki herhangi bir s_i örneği için ağırlıklı maliyet fonksiyonu $H_{s_i}^k$, Denklem (5.39)'daki gibi iki ceza terimi kullanılarak tanımlanmaktadır.

$$H_{s_i}^k = \alpha dist_{s_i}^k + \beta \frac{dist_{s_i}^k}{\sum_{h=1}^{n_h} \sum_{r=1}^{n_r} dist_{r,h,i}^k} + \gamma \frac{dist_{s_i}^k}{1 + \mathbf{v}_{k-1,k} \cdot \mathbf{v}_{k,s_i}} \tag{5.39}$$

Burada $dist_{s_i}^k$ değeri örnek konumu $X_{s_i}^k$ ile hedef konumu X_g arasındaki Öklid uzaklığını, n_r robota ait kapsül sayısını, n_r insana ait kapsül sayısını, $dist_{r,h,i}^k$ i'nci örnek konfigürasyon için r'nci robot kapsülü ile h'nci insan kapsülü arasındaki minimum uzaklığı tanımlamaktadır. $\mathbf{v}_{k-1,k}$, $k-1$'nci iterasyondaki uç konumu X_r^{k-1}'den k'ncı iterasyondaki uç konumu X_r^k'e doğru olan birim vektörünü ifade etmektedir. Benzer şekilde $\mathbf{v}_{k,s_i}$, robot ucunun mevcut konumu X_r^k'den örnek bir konum $X_{s_i}^k$'e doğru olan birim vektörünü temsil etmektedir.

Ağırlıklı maliyet fonksiyonu ayrıca, hedefe yönelme, engellerden kaçınma ve robot ucu için düzgün bir yol planma arasındaki dengeyi ayarlamak için sırasıyla α, β, γ ağırlık parametrelerini içermektedir. Bu parametreler her zaman Denklem (5.40)'ta verilen eşitliği sağlamaktadır.

$$\alpha + \beta + \gamma = 1 \tag{5.40}$$

Optimizasyon probleminin tanımı, Denklem (5.41)'deki gibi bir minimizasyon problemi olarak yapılmaktadır. Denklem (5.42) ve Denklem (5.43) ise kısıtları ifade etmektedir.

$$\min_{q_{s_i}^k} \quad H_{s_i}^k \tag{5.41}$$

$$\text{s.t.} \quad q_{lim}^- < q_{s_i}^k < q_{lim}^+ \tag{5.42}$$

$$dist_{r,h,i}^k > L_{danger} \tag{5.43}$$

Burada q_{lim}^- ve q_{lim}^+ değerleri robotun eklem açı sınırlarını tanımlamakta ve L_{danger} değeri ise robotun çalışmaya devam edebilmesi için ayırma mesafesi sınırını belirlemektedir.

Eklem kısıtı ile eklem açılarının limitlerin dışına çıkması engellenmekte ve kısıtı karşılamayan örnekler elimine edildiğinden uzaklık hesabı yapılmamaktadır. Mesafe kısıtlaması ile de robot, bulunduğu ortamda emniyetli bir şekilde faaliyet göstermektedir. Örnek konfigürasyonların hiçbirinin verilen kısıtı sağlamadığı durum, robotun o iterasyon sonucunda mevcut konfigürasyonunu koruyacağı anlamı taşımaktadır. Verilen optimizasyon probleminin çözümü, k'ncı iterasyonda, aynı zamanda örnek konfigürasyonlardan biri olan bir konfigürasyon q_w^k sağlamaktadır. Bu konfigürasyon, aynı zamanda eklem kontrol komutunu ve robotun ucunun hareket ettirileceği konum olan X_w^k'yı da belirlemektedir.

6. BULGULAR VE TARTIŞMA

Tez kapsamında tasarlanan sistemin ve önerilen yöntemin doğrulanması için hem gerçek deneyler, hem de benzetim ortamında uygulamalar yapılmaktadır. Bölüm 5.1.2'de tasarlanan sistem kullanılarak benzetim ortamındaki robot üzerinde gerçekleştirilen deneyler ve sonuçları Bölüm 6.1.2'de, gerçek ortamdaki robot üzerinde gerçekleştirilen deneyler ve sonuçları da Bölüm 6.1.3'te anlatılmaktadır.

Önerilen yol planlama yönteminin doğrulandığı benzetim çalışmaları bu tez çalışmasında farklı senaryolar üzerinden gerçekleştirilmektedir. MATLAB ortamında gerçekleştirilen benzetim çalışmaları tek engel içeren senaryo için Bölüm 6.2.1'de, çoklu engel içeren senaryo için Bölüm 6.2.2'de ve hareketli engel içeren senaryo için de Bölüm 6.2.3'te açıklanmakta ve sonuçları yorumlanmaktadır. Bölüm 6.2.4'te optimizasyon problemine ait maliyet fonksiyonunda yer alan ağırlık parametreleri detaylı olarak tartışılmaktadır. Son olarak Bölüm 6.2.5'te ise kapsül miktarının ve örnek nokta sayılarının hesaplama süreleri üzerindeki etkisi incelenmektedir.

6.1 İskelet Takibi Temelli Minimum Uzaklık Hesaplama için Deneysel Çalışmalar

Tez kapsamında önerilen sistemi test etmek için, bir insan operatörün endüstriyel iş hücresini bir endüstriyel robot manipülatörü ile paylaştığı çeşitli senaryolar uygulanmaktadır. Her senaryoda insan, robotun çalışma alanının çevresinde veya içinde farklı pozisyonlarda durmakta ve farklı duruşlara sahip olmaktadır. Gerçekleştirilen deneylerde, insan ve robot uzuvları arasındaki minimum uzaklıklar tekrarlı bir şekilde hesaplanmakta ve sonuçlar ilk olarak benzetim ortamında analiz edilmekte ve görselleştirilmektedir. Benzetim ortamındaki robot ile gerçekleştirilen performans deneyleri dışında, SMS tipinde iş birlikçi işlemleri içeren bazı uygulamalar da gerçek bir robot manipülatörü kullanılarak gerçekleştirilmektedir.

6.1.1 Deney kurulumu

İskelet takibi temelli minimum uzaklık hesaplama deneyleri için oluşturulan endüstriyel iş hücresi kurulumu üç ana unsurdan oluşmaktadır: bir insan operatör, bir endüstriyel robot manipülatörü ve bir RGB-D algılayıcı. Gerçekleştirilen deneylerde endüstriyel robot manipülatörü olarak 6-eksenli Kawasaki RS005L robot manipülatörü

kullanılmaktadır. Benzetim ortamında ise yine aynı robota ait modele yer verilmektedir ve benzetim faaliyetleri Gazebo aracılığıyla ROS ortamı üzerinde yürütülmektedir. Algılayıcı olarak Microsoft Kinect v2 RGB-D algılayıcı kullanılmakta ve bu algılayıcı robotun çalışma alanını görüntüleyecek şekilde iş hücresinde yer alan bir konuma sabitlenmektedir. Deneyler öncesinde gerçekleştirilen el-göz kalibrasyonu da, robot taban koordinat sistemi ile algılayıcı koordinat sistemi arasındaki dönüşüm matrisini hesaplamak için kullanılmaktadır. Bölüm 4.2.2.3'de anlatılan süreç takip edilerek elde edilen bu dönüşüm matrisinden ise kapsüllere ait uzaklık hesaplamalarında yararlanılmaktadır. İnsan iskelet takibi için algılayıcıdan libfreenect2 sürücüsü aracılığıyla renkli/derinlik imgeleri elde edilmekte ve OpenNI2 yazılım kütüphanesinin beraberinde NITE2 ara katmanı sayesinde takip gerçekleştirilmektedir. Tüm senaryolar, Intel(R) Core(TM) i7-7500U işlemci ve Ubuntu 16.04.7 LTS işletim sistemine sahip bir bilgisayarda gerçekleştirilmektedir. Bölüm 5.1.2'de tasarlanan sistem, bu deney kurulumu ile doğrulanmakta ve performansı incelenmektedir.

6.1.1.1 Robot temsili

Minimum uzaklık hesaplama sistemi için gerçekleştirilen deneylerde, Şekil 4.3'te gösterildiği gibi tüm robot manipülatörünü temsil etmek için toplamda üç kapsül kullanılmaktadır. Altı eklemli endüstriyel robot manipülatörü Kawasaki RS005L, uç noktasının yönelimini değiştirebilmek için küresel bir bileğe sahiptir. Bu nedenle son üç robot uzvu, sınırlı bir yer değiştirme aralığı ile bilek hareketinden sorumludur. Bu yüzden, sadece robot tabanından itibaren ilk üç uzuv kapsüllerle kapsanmaktadır. Kapsül boyutları, Çizelge 6.1'de verilen robot uzuv boyutları dikkate alınarak tanımlanmaktadır.

Çizelge 6.1: Kawasaki RS005L endüstriyel robot manipülatör için kapsül boyutları

Kapsül	Kapsüllerle kaplı robot uzuvları	Dönüşüm matrisi	Kapsül boyutları	
			Uzunluk (mm)	Yarıçap (mm)
C_1	Link 1 (O_{R1})	$T_{C_1}^{R1}$	80	165
C_2	Link 2 (O_{R2})	$T_{C_2}^{R2}$	340	155
C_3	Link 3, 4, 5 ve 6 (O_{R3}, O_{R4}, O_{R5}, O_{R6})	$T_{C_3}^{R4}$	365	100

Kapsül koordinat sisteminden ilgili link koordinat sistemine (Ri) olan dönüşümü ifade eden sabit homojen dönüşüm matrisleri ise Denklem (6.1), Denklem (6.2) ve Denklem (6.3)'teki gibi tanımlanmaktadır.

$$T_{C_1}^{R1} = \begin{bmatrix} R_x(-\pi/3) & \begin{bmatrix} 0 \\ 0 \\ -60 \end{bmatrix} \\ 0_{1x3} & 1 \end{bmatrix} \tag{6.1}$$

$$T_{C_2}^{R2} = \begin{bmatrix} R_y(\pi/2) & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0_{1x3} & 1 \end{bmatrix} \tag{6.2}$$

$$T_{C_3}^{R4} = \begin{bmatrix} R_z(\pi/2) & \begin{bmatrix} 0 \\ 0 \\ -30 \end{bmatrix} \\ 0_{1x3} & 1 \end{bmatrix} \tag{6.3}$$

Burada $R_x(-\pi/3) \in \mathbb{R}^{3\times3}$ matrisi x-ekseni etrafında $-\pi/3$ dönüşü, $R_y(\pi/2) \in \mathbb{R}^{3\times3}$ matrisi y-ekseni etrafında $\pi/2$ dönüşü ve $R_z(\pi/2) \in \mathbb{R}^{3\times3}$ matrisi z-ekseni etrafında $\pi/2$ dönüşü temsil etmektedir.

6.1.1.2 İnsan temsili

Kafa, boyun, gövde, sol/sağ omuz, sol/sağ dirsek, sol/sağ bilek ile ilgili dokuz nokta olan eklem pozisyonları, Çizelge 6.2'de boyutları verilen insan kapsüllerinin ötelemesini ve döndürmesini hesaplamak için kullanılmaktadır.

Çizelge 6.2: İnsan için kapsül boyutları

Kapsül	Kapsüllerle kaplı insan uzuvları	Kapsül boyutları	
		Uzunluk (mm)	Yarıçap (mm)
C_1'	Kafa	150	120
C_2'	Gövde	500	200
C_3'	Sağ üst kol	200	70
C_4'	Sol üst kol	200	70
C_5'	Sağ alt kol	250	70
C_6'	Sol alt kol	250	70

6.1.2 Benzetim ortamındaki robot ile gerçekleştirilen deneylere ait sonuçlar

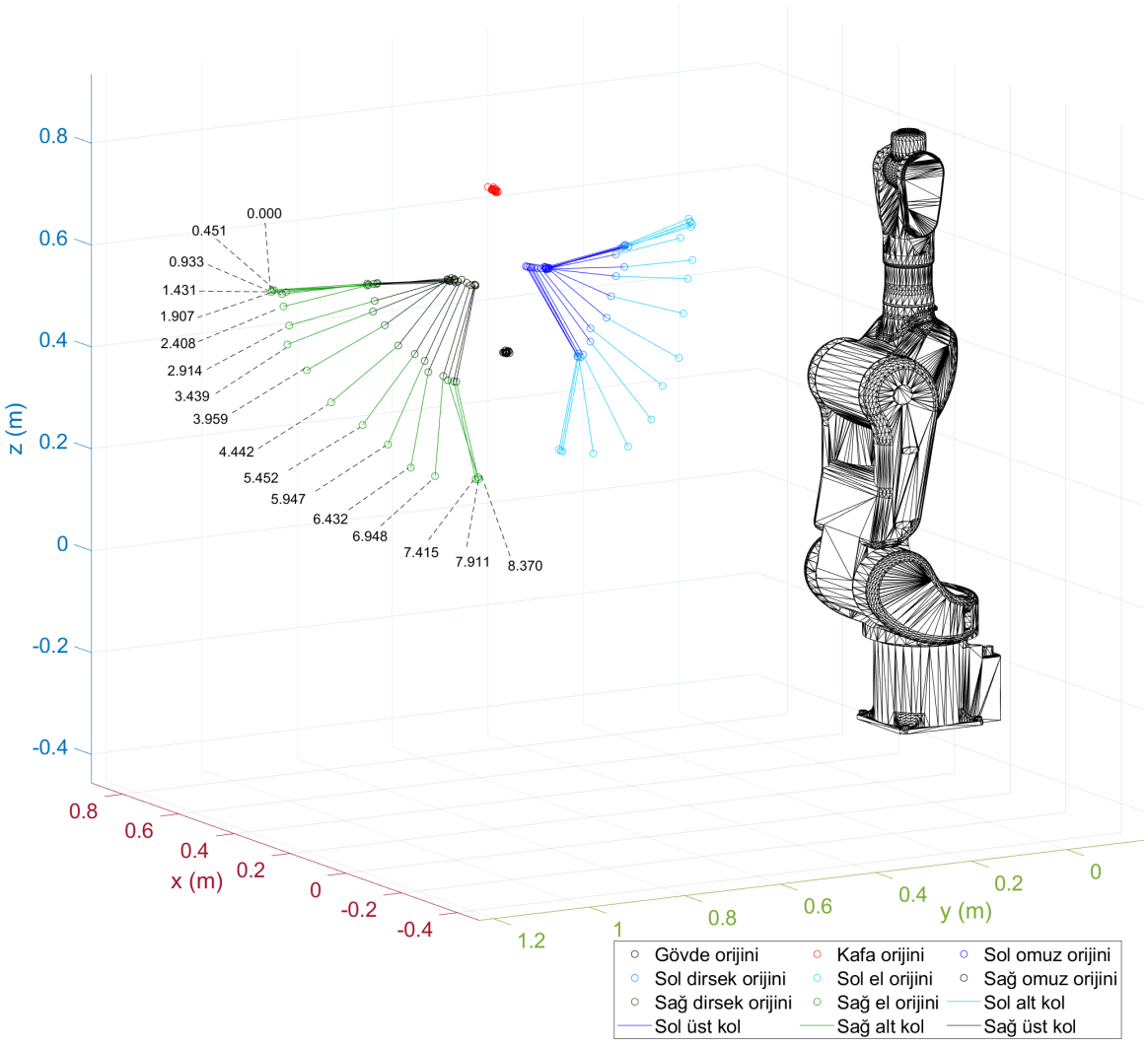
Önerilen hesaplama sistemi ilk olarak bir insan operatörün robotun çalışma alanı çevresinde ve içinde hareket ettiği deneylerle test edilmektedir. İnsan operatör her deney esnasında iskelet izleme performansını test etmek için farklı hızlarda hareket etmekte ve farklı duruşlara sahip olmaktadır. Bu sistemin temel amacı, emniyetli insan-robot etkileşimi için minimum uzaklığı hesaplamaktır. Ancak benzetim ortamındaki robotla yapılan deneyler sırasında herhangi bir çarpışmadan kaçınma yöntemi uygulanmamaktadır. Bu nedenle endüstriyel robot manipülatörü, insan operatörün hareketi sırasında sabit bir konumda tutulmaktadır. Buna rağmen sistemin performansını ve uygulanabilirliğini göstermek için, robot konfigürasyon verileri sürekli olarak ”state publisher” düğümünden okunmakta ve güncellenmektedir. Aynı şekilde okunan veriler için robot kapsül dönüşümleri de uygulanmaktadır. Öyle ki, her hesaplama döngüsünde, üç robot kapsülünün de dönüşümleri, konfigürasyon verilerine göre belirlenmektedir.

Benzetim ortamındaki robotun eklem durumları, bu deneylerde Gazebo benzetimi ile sağlanmaktadır. Buna göre, insanın robotun çalışma alanı çevresinde yürüdüğü ve farklı duruşlar aldığı sekiz deney yapılmaktadır. Bu deneylerin her birinde, insan operatör tarafından gerçekleştirilen hareketlerin hızları da farklılık göstermektedir. Algılayıcı tarafından yakalanan ve önerilen sisteme göre hesaplanan tüm veriler daha sonra analiz edilmek üzere kaydedilmektedir. Sekiz deneyin her birinde algılayıcı tarafından yakalanan imge verileri Şekil 6.1’de gösterilmektedir.



Şekil 6.1: Sekiz deney sırasında elde edilen imgeler (sırasıyla ilk kare, ara karelerin birleşimi, son kare)

Gerçekleştirilen deneylerin her birinde algılayıcıdan elde edilen imge kareleri zaman damgaları ile birlikte kaydedilmektedir. Benzer şekilde, iskelet takip yazılımı aracılığıyla takip edilen insan eklem pozisyonları da zaman damgaları ile birlikte kaydedilmektedir. Şekil 6.2, gösterimin daha kolay anlaşılabilmesi için belirli anlarda iskelet izleme düğümü tarafından yakalanmış insan iskeletinin eklem pozisyonlarına ait nicel verileri göstermektedir. Şekil 6.2'de de görüldüğü üzere, ilk deneyde insan operatör kollarını sonuna kadar açmış bir şekilde hareketine başlamakta ve olduğu yerde dururken kollarını indirerek hareketini 8.370 saniyede tamamlamaktadır.



Şekil 6.2: İlk deney esnasında insan iskeletinin eklem konumları ve onlarla ilişkili zaman damgaları (Yaklaşık 450 ms zaman aralığı ile sadece belirli örnekler)

Nicel analiz sonuçlarının ortaya koyduğu üzere, önerilen iskelet izleme performansı tatmin edici sonuçlar vermektedir. Ancak, gerçekleştirilen tüm deneylerde insanın

algılayıcının algılama menzili içerisinde yer aldığını ve algılayıcının insanı algılayabilecek net bir görüş açısına sahip olduğunu belirtmek gerekmektedir. Elbette algılayıcının görüşünün robot veya çeşitli engeller tarafından engellendiği durumlarda, çoklu algılayıcı füzyonu gibi ek yöntemlere ihtiyaç duyulabilmektedir. Böyle bir durumda, çoklu algılayıcı füzyonu da konum hatasının azaltılmasına ve dolayısıyla daha güvenilir verilerin elde edilmesine katkıda bulunabilir.

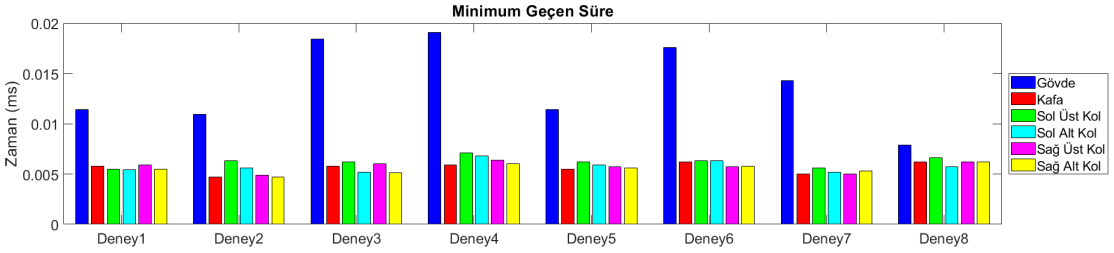
Önerilen sistem, ayrıca hesaplama süresi açısından da incelenmektedir. Bu nedenle, Kinect v2'nin kare hızı (yani ardışık iki kare arasında geçen süre t_f) ve iskelet izleme algoritmasının işlem süresi t_s ölçülmektedir. Yapılan ölçümlerde, algılayıcının kare hızının yaklaşık olarak 30 fps (saniye başına kare) değerinde ve ardışık iki kare arasında geçen ortalama sürenin de 32.8022 ms dolaylarında olduğu görülmektedir. Ayrıca, iskelet takip hesaplamaları için kare başına geçen işlem süresinin ortalama olarak 4.9481 ms olduğu görülmektedir.

Uzaklık hesaplama performansını değerlendirmek için ise gerçekleştirilen sekiz farklı senaryodan yararlanılmaktadır. Bu senaryoların her birinde insan, robota farklı yönlerden ve farklı hızlarda yaklaşmakta ve belirli vücut kısımlarını algılayıcı menzili içinde hareket ettirerek farklı duruşlar almaktadır. Her hesaplama döngüsünde de, altı insan kapsülünün her birinden üç robot kapsülüne olan minimum uzaklığı hesaplamak için GJK uzaklık çözücüsü kullanılmaktadır. Uzaklık hesaplaması sırasında geçen süre t_d, Şekil 6.3'te gösterildiği gibi kaydedilmektedir. Altı insan kapsülü ile robotun üç kapsülü arasındaki minimum uzaklığı hesaplamak için geçen süreler; ortalama 0.1407 ms, minimum 0.0371 ms, maksimum ise 10.4491 ms olarak görülmektedir. Sürelere ait istatistiki verilerin özeti; medyan, alt ve üst çeyrekler (İng. quartiles), aykırı değerler (çeyreklerin arası aralık kullanılarak hesaplanır) ve aykırı olmayan minimum ve maksimum değerler ise Şekil 6.4'te gösterilmektedir. Buna göre, bir imgenin elde edilmesinden tüm minimum uzaklıkların ve en yakın noktaların üretilmesine kadar olan her aşamanın toplam hesaplama süreleri de Çizelge 6.3'te verilmektedir.

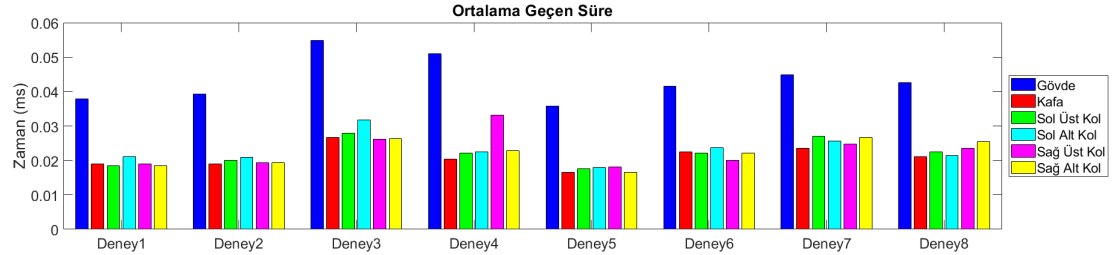
Çizelge 6.3: Hesaplama sürelerinin özeti

İşlem süresi † (ms)	Karenin elde edilmesi t_f	İnsan/Robot verilerinin elde edilmesi t_s	Minimum uzaklık ve en yakın noktaların hesaplanması t_d
Medyan	32.8022	4.9481	0.1407
5. Yüzdelik (5th percentile)	12.0429	0.1654	0.0767
95. Yüzdelik (95th percentile)	80.2257	16.6654	0.2237

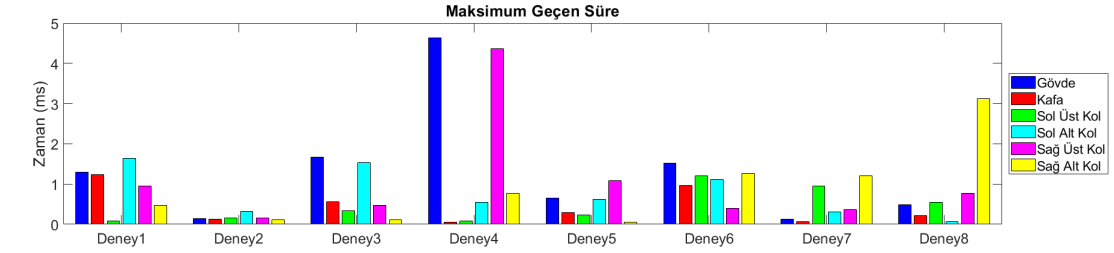
† Döngü süresini hesaplamak için, kare aktarımından gelen 41.7ms gecikme ve ROS tarafından kaynaklanan 7ms gecikme (t_f+t_s+t_d)'ye eklenmelidir.



(a)

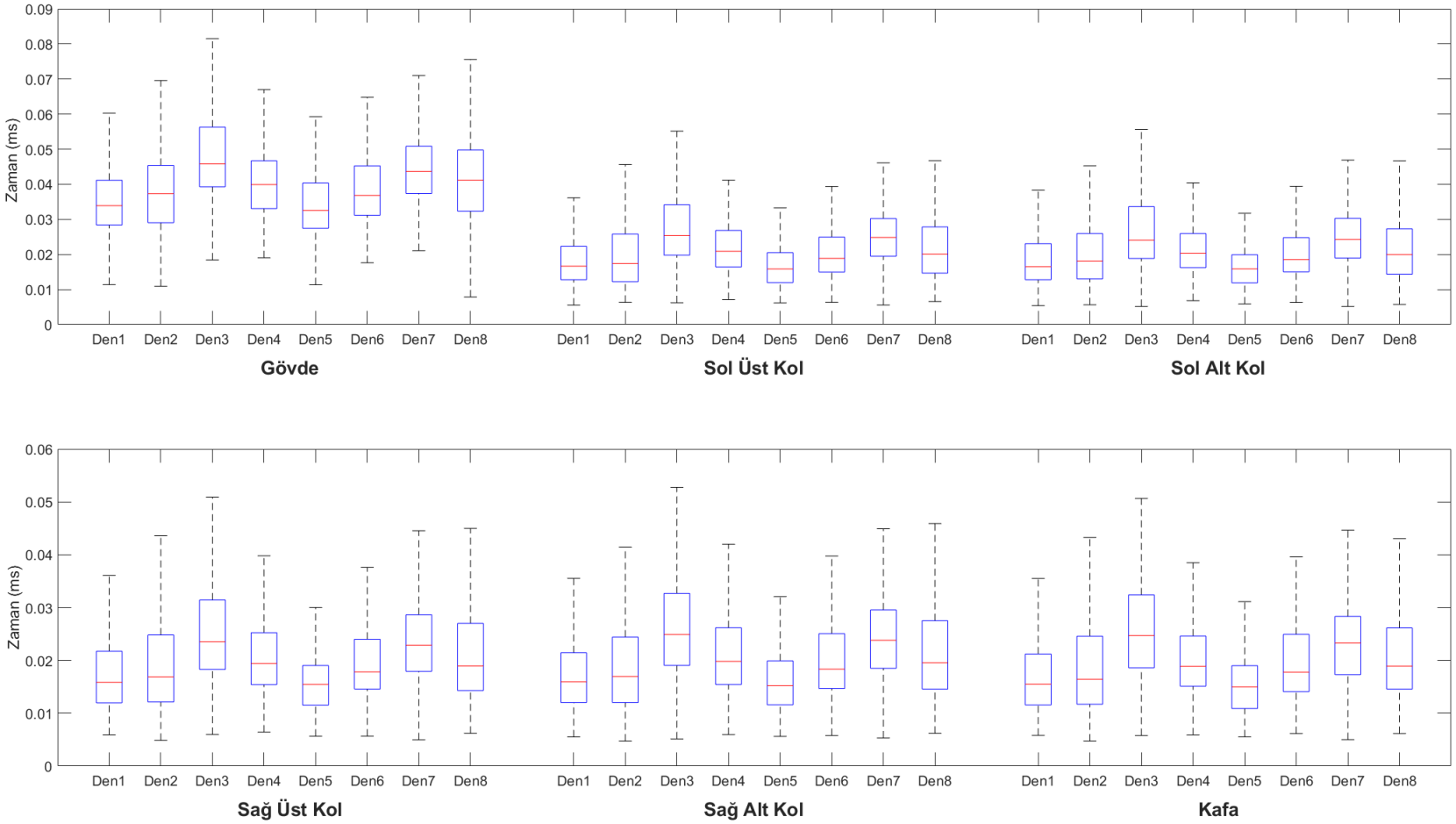


(b)



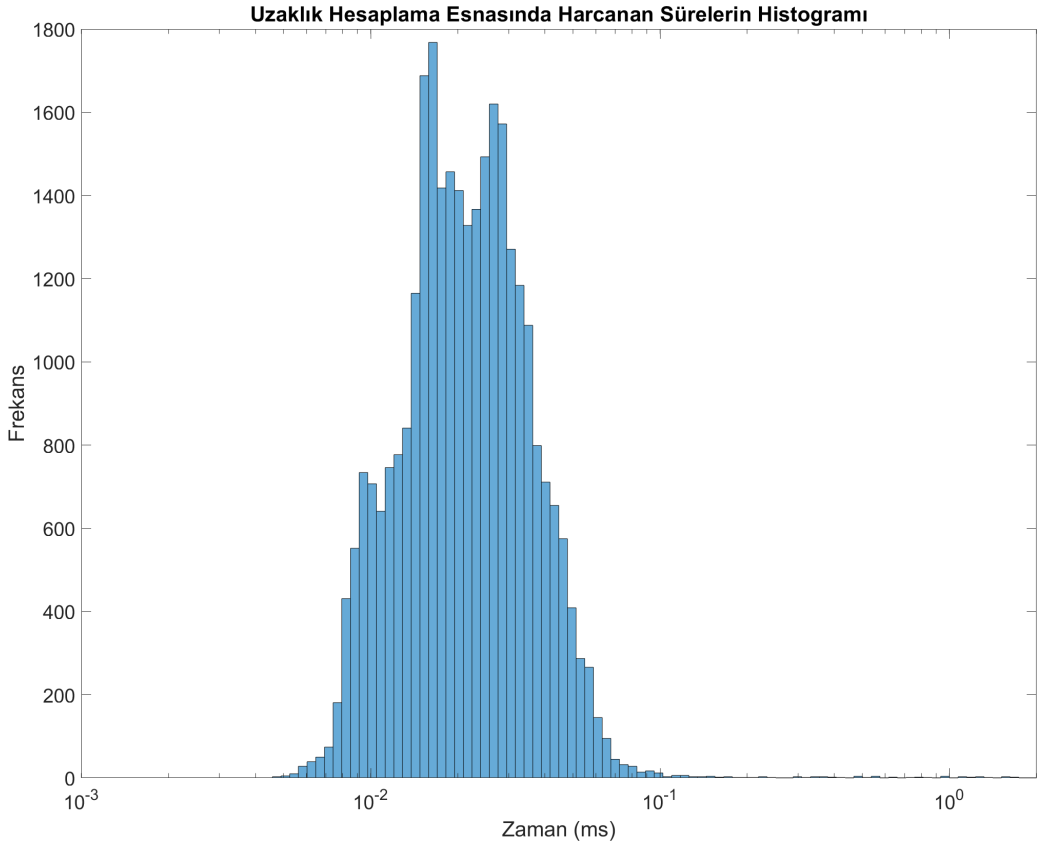
(c)

Şekil 6.3: Her bir insan kapsülünden üç robot kapsülüne minimum uzaklık hesaplaması sırasında geçen süreler t_d: (a) minimum geçen süre, (b) ortalama geçen süre, (c) maksimum geçen süre



Şekil 6.4: Minimum uzaklık hesaplama sürelerine ait verilerin istatistiği: medyan, alt ve üst çeyrekler, minimum ve maksimum değerler

Tüm bu veriler incelendiğinde ve her bir insan kapsülü için uzaklık hesaplamasında harcanan sürenin yaklaşık olarak 0,02 ms olduğu göz önüne alındığında, farklı senaryolarda (örn. mobil manipülatörleri içeren) kapsül sayısında yaşanabilecek nispeten küçük bir artış, özellikle gerçek zamanlı etkileşim açısından kritik değişikliklere neden olmayacaktır. Şekil 6.5'te yer alan ve bir insan kapsülünden üç robot kapsülüne gerçekleştirilen minimum uzaklık hesaplamalarının süresini gösteren histogram da bu görüşü desteklemektedir. Zabalza vd. (2019) açıklandığı gibi, hem robot hem de insan için tepki süresi, dinamik bir ortamdaki değişikliklere emniyetli bir şekilde yanıt verebilmek için büyük önem taşımaktadır. Bir robotun tepki süresi, algılama, planlama ve çalıştırma ile ilgili hesaplama sürelerine bağlıdır. Dolayısıyla emniyetli bir insan-robot etkileşimi için algılama aşamasında harcanacak hesaplama süresinin kısa olması beklenmektedir. Sonuçlar, önerilen sistemin bu emniyetli insan-robot etkileşimi gereksinimini karşılayabileceğini göstermektedir.



Şekil 6.5: Uzaklık hesaplama esnasında geçen sürelere ait histogram

Diğer aşamalarda olduğu gibi kapsüllerin tanımlanması ve dönüşümlerinin hesaplanması için de belirli bir zaman harcanmaktadır. Bu nedenle, bu aşamada harcanan

süre için başka bir nicel analiz daha yapılmaktadır. Geometrik ilkellere dayalı temsilin performansını test etmek için gerçekleştirilen bu çalışmalar, önerilen yaklaşımın birkaç kez çalıştırılmasına dayanmaktadır. Bu nedenle her çalıştırmada, her bir insan kapsülünün tanımlanması ve dönüşümlerin hesaplanması için harcanan zaman ölçülmektedir. Bir insan kapsülünü ve bununla ilgili dönüşümleri tanımlamak için harcanan ortalama süre yaklaşık olarak 0,0029 ms dolaylarında görülmektedir. Bu durum, bir kapsül tanımlamanın ve dönüştürmenin basit bir işlem olduğunu ve diğer hesaplama süreçlerine harcanan zamana kıyasla zaman alıcı olmadığını göstermektedir. Ek olarak, altı insan kapsülünün tamamını oluşturmak için iskelet takip verisi üzerinde gerçekleştirilen işlemin süresi ise ortalama olarak 7.1134 ms'dir (medyan 4.9481 ms). Dışbükey gövde yaklaşımının kullanıldığı Han vd. (2018)'e ait çalışmada, insan koluna (robot çalışma alanındaki tek vücut parçası) ait oktree haritasının sadece dışbükey gövdesini hesaplamanın ortalama süresi yaklaşık 1.501 ms'dir ve statik çarpışma algılaması ortalama 1.543 ms sürmektedir. Toplam süre, deneylerde dikkate alınan tek bir vücut parçası için yaklaşık olarak 3.044 ms'ye karşılık gelmektedir. Ancak, nokta bulutu görüntülerinin oktree haritalarına dönüştürülmesi sırasında geçen süre belirtilmemektedir. Ayrıca, diğer vücut kısımları da dikkate alındığında hesaplama süresinin nasıl değişeceğine dair bir gösterge bulunmamaktadır. Bu bağlamda, bu tez çalışmasında önerilen sistem ile tüm üst vücut için yapılan modellemenin, Han vd. (2018)'e ait yaklaşımdan daha yavaş göründüğü ancak daha doğru bir modelleme sağladığı söylenebilir.

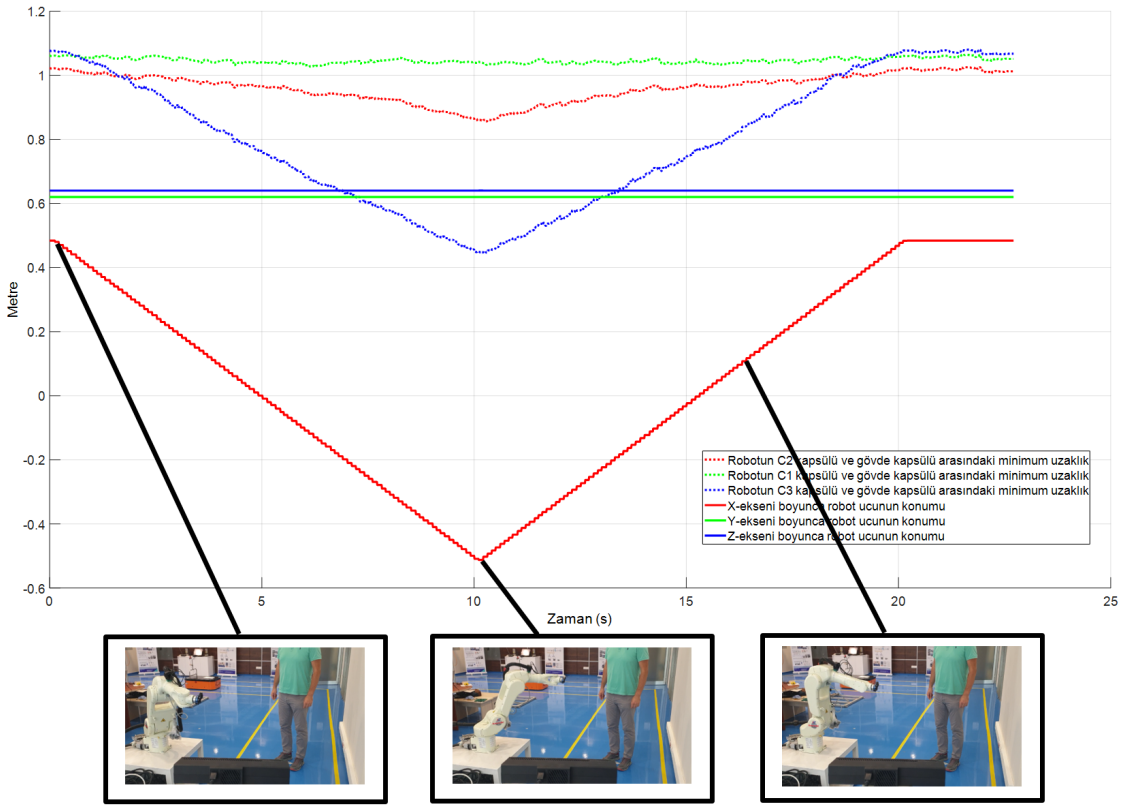
6.1.3 Gerçek ortamdaki robot ile gerçekleştirilen deneylere ait sonuçlar

Gerçek ortamdaki robot ile gerçekleştirilen deneylerde, önerilen sistemin fizibilitesini göstermek için bir endüstriyel operasyon tanımlanmaktadır. Endüstriyel operasyon, robot hızının ve robota göre insan konumunun farklı olduğu altı farklı senaryodan oluşmaktadır. Bu senaryolarda emniyet dereceli gözlenen durdurma (SMS) türü uygulandığından, robot ile insan arasındaki uzaklık önerilen sistem kullanılarak ve gerçek zamanlı olarak sürekli olarak ölçülmektedir. Amaç, robot ile insan arasındaki uzaklık bir emniyet mesafesi eşiğinin altına düştüğünde robot hareketini durdurmaktır. Bu nedenle, SMS çalışma türünün geçerli olduğu her senaryo için emniyet mesafesi eşiği 500 mm olarak belirlenmektedir.

Her senaryoda robot, uç noktasını doğrusal olarak (x-ekseni boyunca) görev alanında belirli bir konuma hareket ettirmektedir. Daha sonra aynı yolu izleyerek ilk konumuna geri dönmektedir. Robot hareketi esnasında uç noktası, hareket hızının etkisini göstermek için ilgili senaryoya bağlı olarak 100 mm/s veya 250 mm/s sabit doğrusal hıza sahip olmaktadır. Normal koşullarda 250 mm/sn, ISO 10218 standartlarına dayanan emniyet dereceli gözlenen durdurma sınırıdır (ISO, 2011; ISO, 2016). Gerçek ortamdaki

robot ile gerçekleştirilen operasyonlar sırasında robot eklem konfigürasyonu, Bölüm 6.1.3'dekinin aksine artık robot denetleyicisi tarafından gerçek zamanlı olarak sağlanmaktadır.

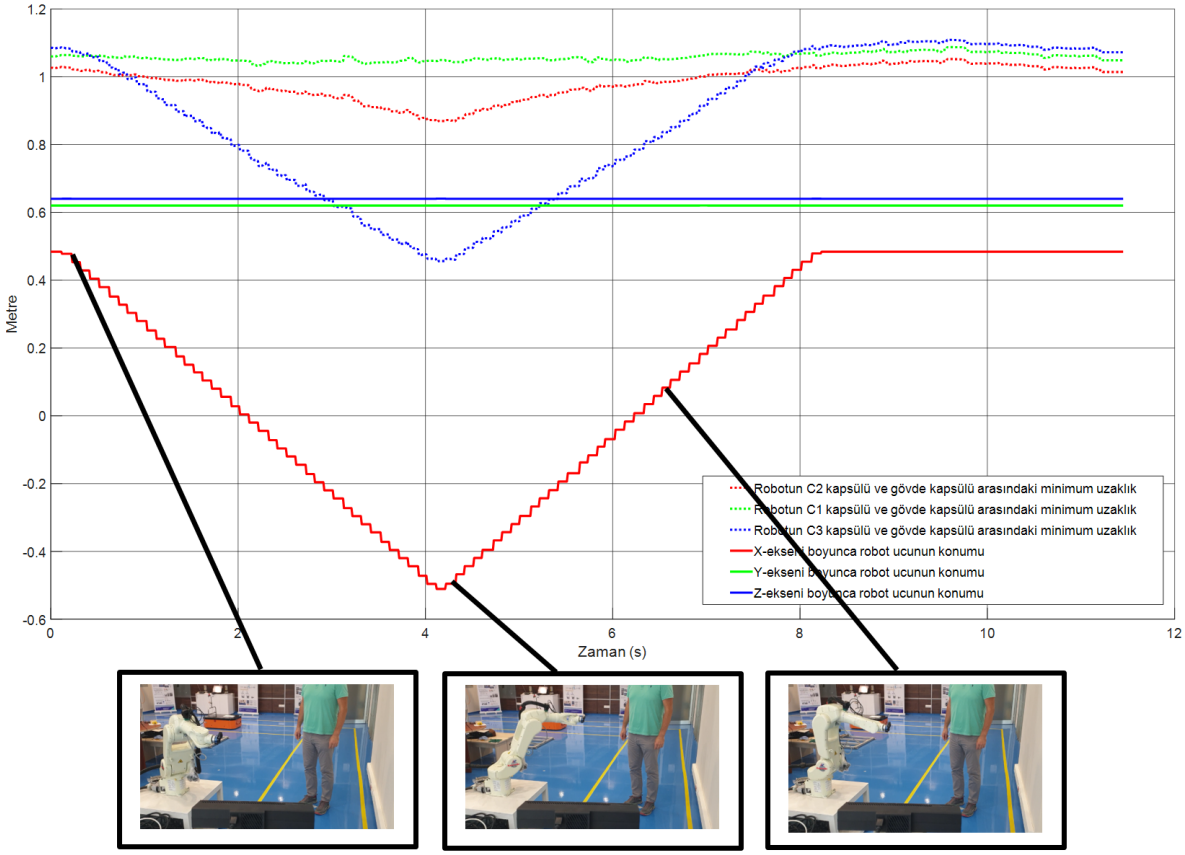
Senaryolardan ilkinde (Senaryo-1), robot Şekil 6.6'da gösterildiği gibi 100mm/s'lik sabit bir doğrusal hızla hareket ederken, insan robottan belirli bir mesafe uzakta durmaktadır. Senaryo-1'deki robot hareketi esnasında sadece uzaklıklar ölçülmektedir. Herhangi bir şekilde SMS türü etkileşim uygulanmamaktadır. Şekil 6.6, aynı zamanda her bir robot kapsülü ile insana ait gövdenin kapsülü arasında ölçülen uzaklığı da göstermektedir. Robot belirtilen hedef konumuna ulaşırken insana yaklaştıkça robot kapsülleri ile insan kapsülü arasındaki ölçülen uzaklığın da azaldığı görülmektedir. Benzer şekilde, robot tekrar başlangıç konumuna dönerken insan ile aralarındaki uzaklık da aynı şekilde artmaktadır. Ölçülen mesafelerden biri, küçük bir zaman aralığında emniyet mesafesi eşiğinin altına düşse de Senaryo-1 için SMS etkileşim türü etkin olmadığından robot hareketi durmamaktadır. Bu senaryo ile gösterilmek istenen, düşük robot hızlarında uzaklık hesaplama performansının etkinliğidir.



Şekil 6.6: Senaryo-1'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 100 mm/sn)

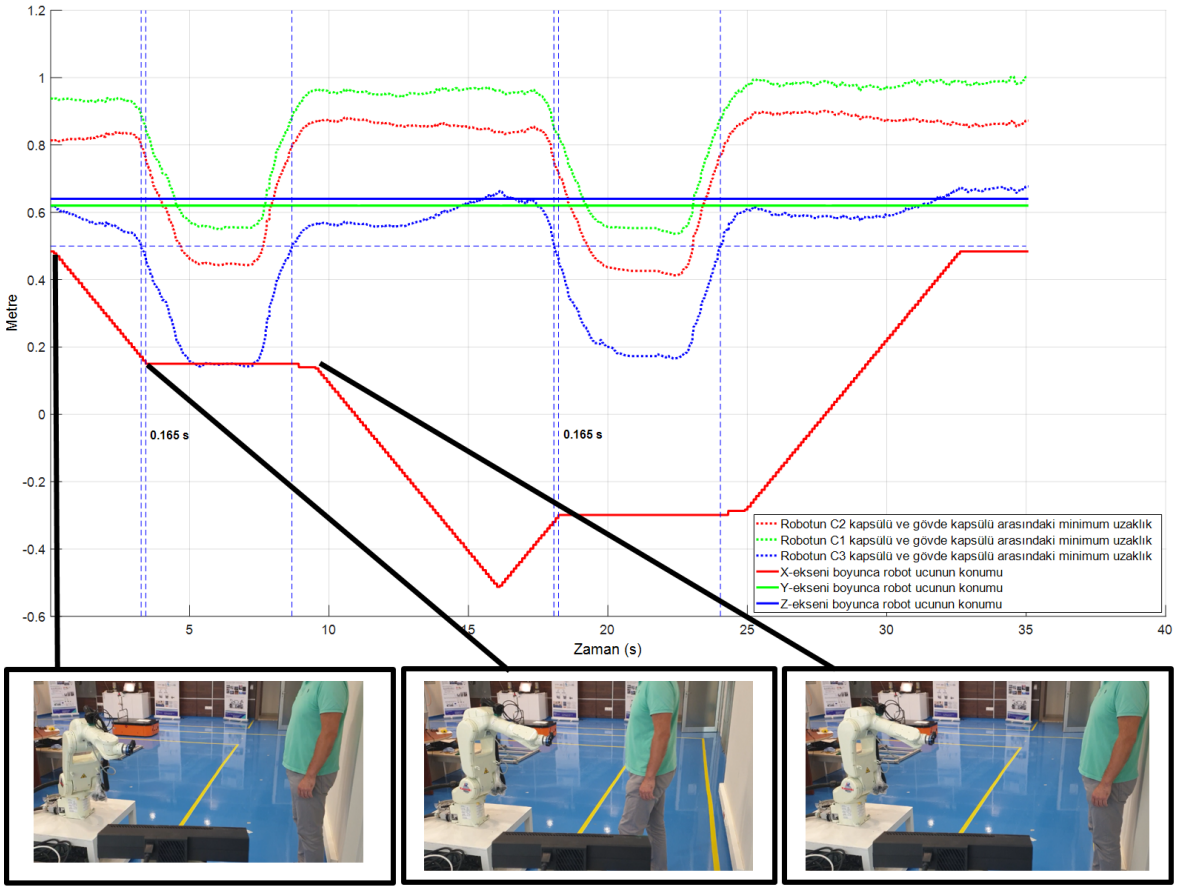
Senaryo-1'in bir sonraki adımında ise robot ucunun sahip olduğu doğrusal hız 250 mm/s'ye çıkarılmaktadır. Senaryoda yer alan bunun dışındaki tüm koşullar ise aynen korunmaktadır. Aynı senaryo koşulları için sadece robot hızının 250 mm/sn'ye yükseltildiği deneye ait sonuçlar Şekil 6.7'de gösterilmektedir. Sonuçlardan da anlaşılacağı üzere robot, hareketini artık daha kısa sürede tamamlamaktadır. Robot ve insan arasında ölçülen uzaklık, yine kısa süreliğine de olsa emniyetli mesafe eşiğinin altına düşmesine rağmen robot hareketi durmamaktadır.

Gerçek ortamdaki robot ile gerçekleştirilen deneyler esnasında diğer insan kapsüllerine olan uzaklıklar da ölçülmektedir. Ancak deney sonuçlarını gösteren şekillerde, görselleştirmeyi sadeleştirmek adına sadece gövde kapsülü ile ilgili ölçümler gösterilmektedir.



Şekil 6.7: Senaryo-1'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 250 mm/sn)

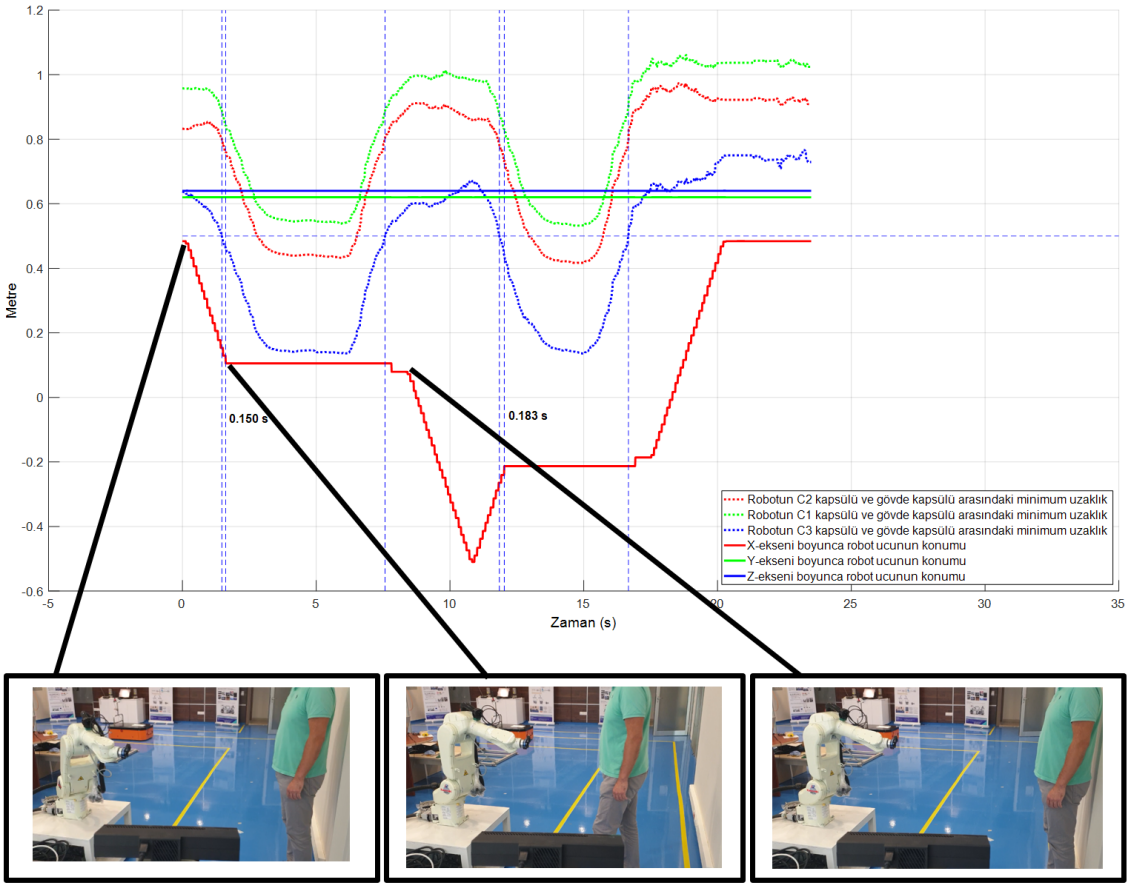
Senaryo-2'de, önceki senaryonun aksine, SMS türünde etkileşim etkinken insan robota yaklaşmaktadır. Bu senaryoya göre, robot ve insan arasında ölçülen uzaklık emniyetli mesafe eşiğinin altına düştüğünde robotun durması beklenmektedir. Şekil 6.8'de yer alan görüntülerde görüldüğü gibi, robot 100mm/s'lik sabit bir doğrusal hızla belirli bir hedef konumuna hareket ederken, insan adım atarak robota yaklaşmaktadır. İnsan robota yeterince yaklaştıktan sonra, bir süre hareketsiz kalmakta ve ardından önceki konumuna dönmek üzere bulunduğu yerden uzaklaşmaktadır. İnsanın uzaklaşmasıyla beraber robot yoluna devam etmekte ve hedef konumuna ulaşmaktadır. Robot hedef konumundan başlangıç konumuna geri dönerken de, insan yine aynı davranışı gerçekleştirmektedir. Şekil 6.8, insan ile robot arasındaki ölçülen uzaklık 500 mm'nin altına düştüğünde, robotun iki seferde de 0.165 saniye içinde durduğunu ve insan uzaklaşana kadar hareket etmediğini göstermektedir.



Şekil 6.8: Senaryo-2'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 100 mm/sn)

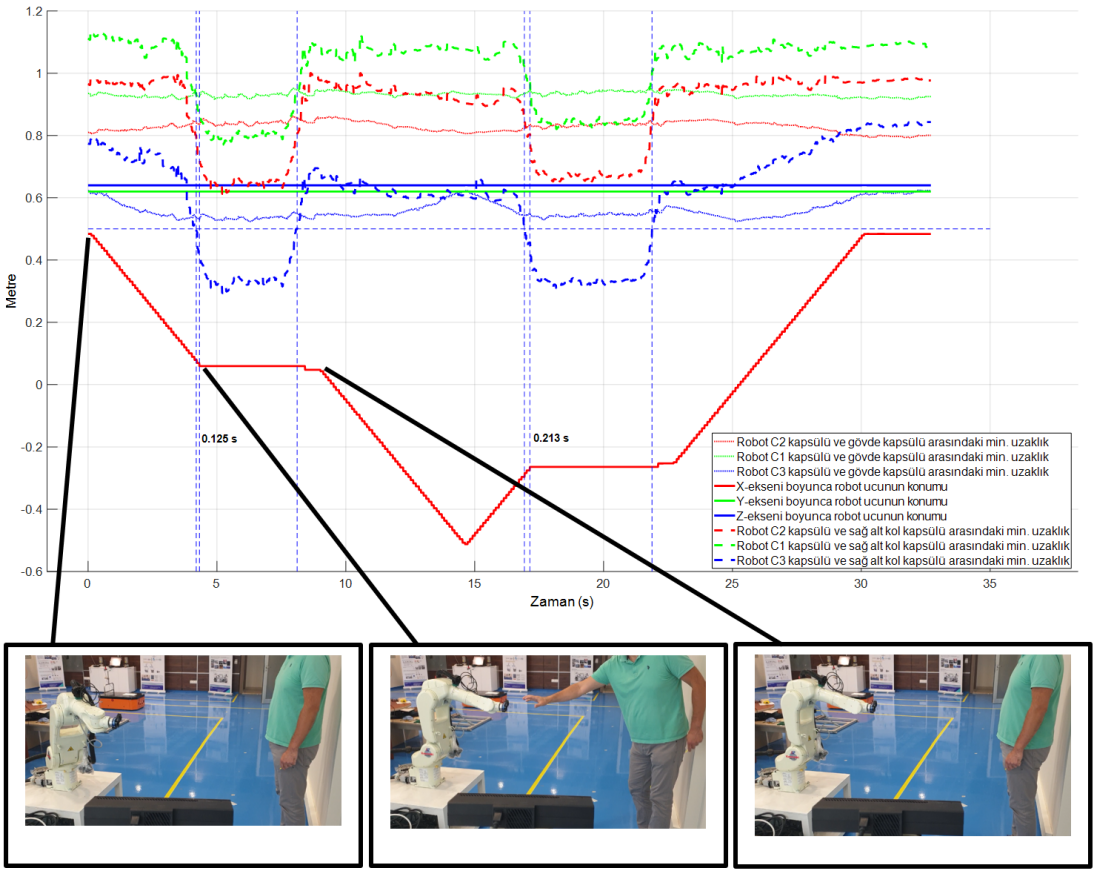
Senaryo-2'nin ikinci kısmında ise robot ucunun sahip olduğu doğrusal hız yine 250 mm/s'ye çıkarılmaktadır. Şekil 6.9, aynı senaryo koşulları için sadece robot hızının 250 mm/sn'ye yükseltildiği sonuçları göstermektedir. Sonuçlar, ölçülen uzaklığın iki kez emniyetli mesafe eşiğinin altına düştüğünü ve robot hareketinin her iki ihlalde de sırasıyla 0.150s ve 0.183s gecikmelerle durduğunu göstermektedir. Ayrıca yine sonuçlardan anlaşılacağı üzere robot, artan hızla beraber tanımlı hareketi daha kısa sürede tamamlamaktadır.

Senaryo-2'ye göre farklı robot hızlarıyla gerçekleştirilen iki deneyden de görüleceği üzere, robotun sahip olduğu hızın tepki süresi üzerinde belirgin bir etkisi olmadığı anlaşılmaktadır. Robot her iki durumda da, durdurma komutuna kısa süre içerisinde tepki vermekte ve insanın robota yaklaşmasından kısa süre sonrasında durmaktadır.



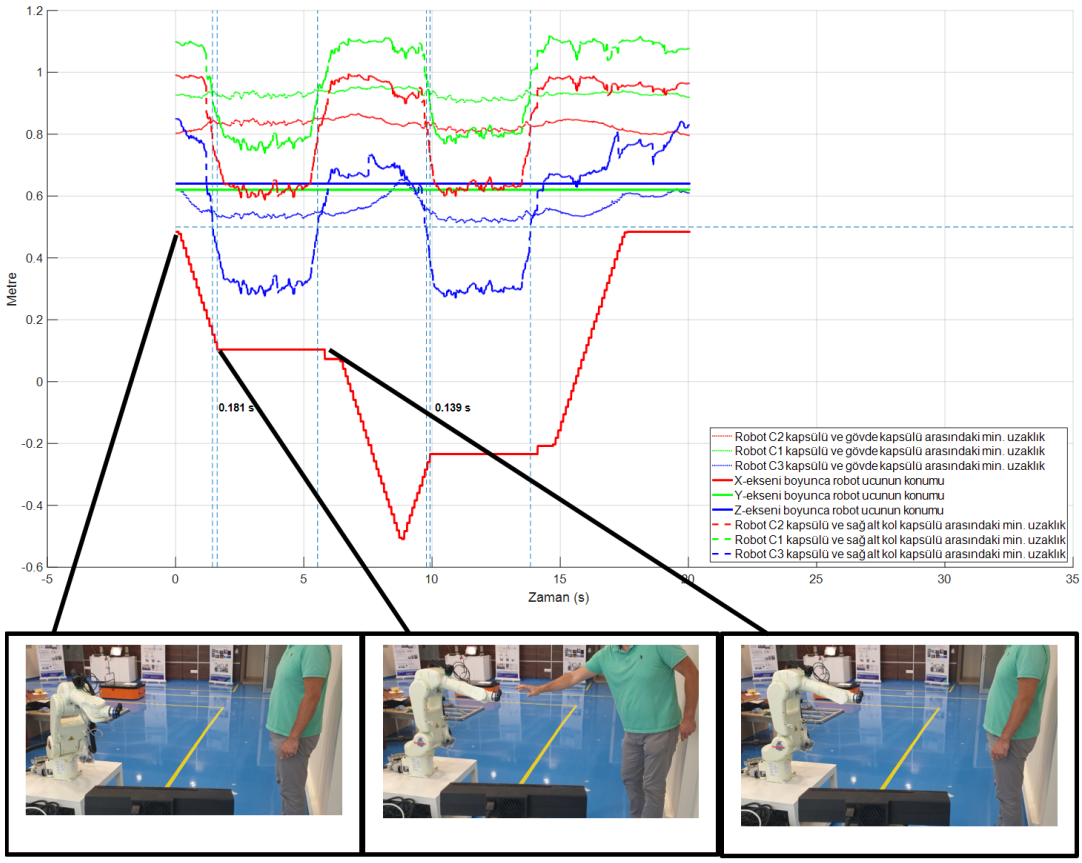
Şekil 6.9: Senaryo-2'de robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 250 mm/sn)

Senaryo-3'te, Senaryo-2'ye göre bu sefer başka bir vücut kısmı robota yaklaşmaktadır. Buna göre, robot hareket ederken insan sağ elini robot operasyon alanına doğru hareket ettirmektedir. Bu senaryoda da, yine SMS türünden etkileşim uygulanmaktadır. Şekil 6.10'da yer alan görüntülerde görüldüğü gibi, robot 100 mm/sn'lik sabit bir doğrusal hızla hareket ederken, insan sağ elini robota yaklaştırmakta ve bir süre sonra geri çekmektedir. Robot başlangıç konumuna dönerken de, insan aynı hareketi tekrarlamaktadır. Bir önceki senaryoya (Senaryo-2) benzer şekilde, emniyetli mesafe eşiği aşıldığında robot sırasıyla 0.125s ve 0.213s gecikmeler içinde hareketini durdurmaktadır. Şekil 6.10'da görüleceği üzere bu deneyde, robot kapsülü ile sağ alt kol kapsülü arasında ölçülen uzaklık emniyetli mesafe eşiğinin altına düşmektedir. Gövde kapsülü için ölçülen uzaklık değeri, robot hareketi boyunca bu eşiğin üstünde yer almaktadır. Bu anlamda robot hareketinin durmasına, robotun operasyon alanına müdahil olan sağ alt kol sebep olmaktadır.



Şekil 6.10: Senaryo-3'te robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 100 mm/sn)

Senaryo-3'ün ikinci kısmında yine robot ucunun sahip olduğu doğrusal hız 250 mm/s'ye çıkarılmaktadır. Şekil 6.11, aynı senaryo koşulları için sadece robot hızının 250 mm/sn'ye yükseltildiği sonuçları göstermektedir. Sonuçlar, ölçülen minimum uzaklığın iki kez emniyetli mesafe eşiğinin altına düştüğünü ve robot hareketinin her iki durumda da sırasıyla 0.181s ve 0.139s gecikmelerle durduğunu göstermektedir. Bu senaryo ile birlikte, farklı vücut kısımları için önerilen sistemin uzaklık hesaplamasındaki etkinliği gösterilmektedir. Senaryo-2'dekine benzer şekilde burada da emniyet ihlali tespit edilir edilmez kısa süre içerisinde robot hareketi durdurulmaktadır.



Şekil 6.11: Senaryo-3'te robot ucunun konumu ve her bir robot kapsülü-gövde kapsülü arasında ölçülen minimum uzaklık (Robot hızı 250 mm/sn)

Sonuç olarak önerilen sistemin fizibilitesini göstermek için, SMS türü iş birliğine dayalı senaryolardan oluşan bir endüstriyel durum tanımlanmıştır. Bu kapsamda, gerçek endüstriyel robot manipülatörü kullanılarak gerçek zamanlı deneyler gerçekleştirilmiştir. Deneysel sonuçlar, insan robotun çalışma alanına girdiğinde, herhangi bir çarpışmayı önlemek için robotun kısa bir zaman aralığında durdurulduğunu göstermektedir. Aynı

şekilde insan ve robot etkileşimi sırasında yapılan uzaklık ölçümleri de tutarlı görünmektedir. Bu sonuçlar, önerilen sistemin emniyetli insan-robot etkileşimi uygulamalarında kullanılabileceğini göstermektedir. Bu sebeple, çarpışmasız yol planlama için gerçekleştirilen deneylerde (Bölüm 6.2) uzaklık hesaplamaları bu sistemin ortaya koyduğu prensiplere dayanmaktadır.

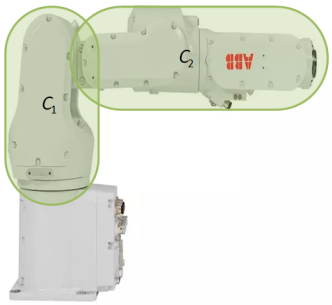
6.2 Çarpışmasız Yol Planlama için Deneysel Çalışmalar

Tez kapsamında önerilen çarpışmasız yol planlama yöntemini test etmek için benzetim ortamında deneyler gerçekleştirilmektedir. Bu bölüm, planlama ortamındaki üç farklı senaryo için gerçekleştirilen benzetimleri açıklamakta ve sonuçlarını ortaya koymaktadır. Benzetimler için MATLAB Robotics System Toolbox™ tarafından sağlanan ABB IRB 120 6-eksenli endüstriyel robot manipülatör modeli kullanılmaktadır. Robotics System Toolbox™, robot modelini kinematik kısıtlar ve dinamik özellikler ile birlikte sağlamakta ve bir eklem-uzay hareket modeli kullanarak robot hareketini simüle etmektedir (MATLAB, 2020). Bu nedenle, bu benzetim ortamı, önerilen yöntemi doğrulamak ve sonuçlarını analiz etmek için bir test ortamı olarak kullanılmaktadır.

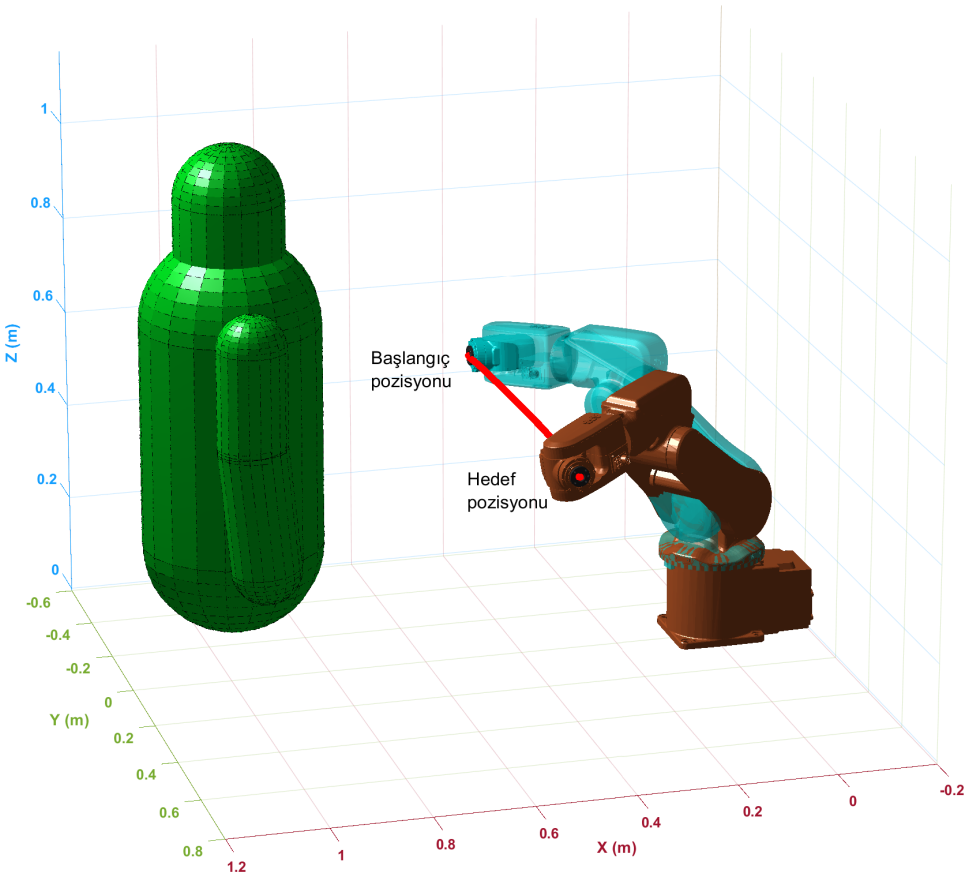
Önerilen algoritma tek engel, çoklu engel ve hareketli engelin yer aldığı üç farklı senaryoda test edilmektedir. Her senaryoda, belirli insan vücut kısımları engel olarak kabul edilmektedir ve bu kısımlar kapsüller ile temsil edilmektedir; tek engel senaryosu için sol alt kol C_3', çoklu engel senaryosu için aynı anda sol alt kol C_3' ve sağ alt kol C_4', hareketli engel senaryoları için de ayrı ayrı gövde C_1' ve sol alt kol C_3' kullanılmaktadır. Ayrıca tekli ve çoklu engel senaryolarında vücut kısımlarının statik olduğu kabul edilmektedir. Benzetim ortamındaki robot modeli de Şekil 6.12'de gösterildiği gibi iki kapsülle temsil edilmektedir; birinci kapsül C_1 robotun birinci ve ikinci uzuvlarını, ikinci kapsül C_2 ise robotun üçüncü, dördüncü, beşinci ve altıncı uzuvlarını kapsamaktadır. Bu iki robot kapsülü ile her bir senaryodaki ilgili vücut kısımlarının kapsülleri arasındaki uzaklık hesaplamaları, Bölüm 5.1.1'de açıklandığı gibi GJK uzaklık algoritması kullanılarak yapılmaktadır.

Benzetim ortamı tasarlanırken, robot için önceden küresel bir yol planının oluşturulduğu ve robotun ucu için hedef pozisyonun halihazırda bilindiği varsayılmaktadır. Önceden oluşturulmuş küresel yol planı, robotun uç noktasının hedefe düz bir çizgi boyunca hareket etmesini sağlamaktır. Bu hareket, malzeme taşıma, tut-yerleştir gibi endüstriyel uygulamalarda yaygın olarak kullanıldığı için tercih edilmektedir. Planlama ortamı, küresel yol planına göre uç noktasının izlediği yol, robotun başlangıç ve hedef pozisyonları, insan

kapsülleri (engel olarak) dahil olmak üzere Şekil 6.13'te gösterilmektedir. İnsan kapsülleri, benzetimler sırasında robotun çalışma alanının önüne yerleştirilmektedir.



Şekil 6.12: ABB IRB 120 6-eksenli endüstriyel robot manipülatörün geometrik temsili (ABB, 2020)



Şekil 6.13: Planlama ortamı: küresel yol planına göre uç noktasının izlediği yol (kırmızı), robotun başlangıç (camgöbeği) ve hedef (turuncu) konfigürasyonları, insan kapsülleri (yeşil)

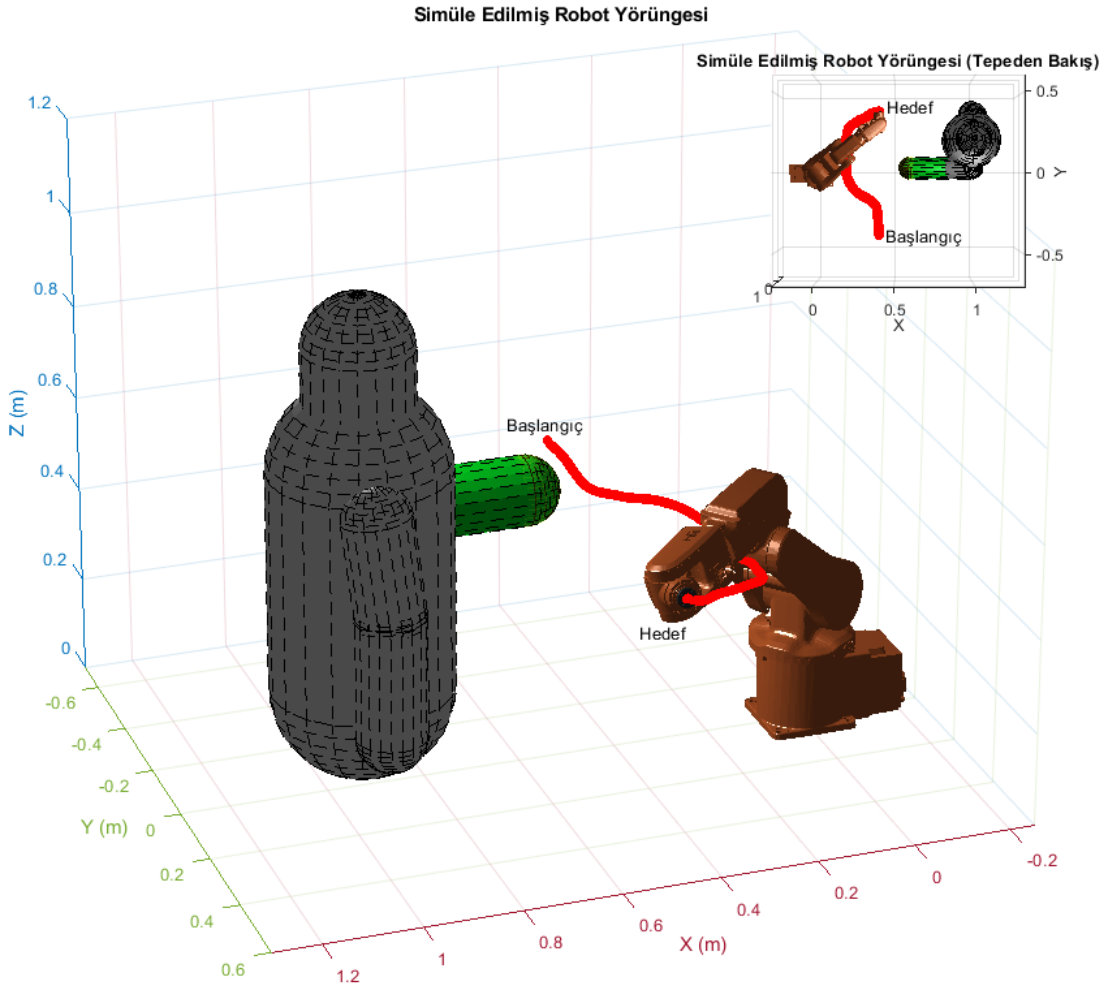
Planlama ortamında, robot ucu için başlangıç ve hedef Kartezyen koordinatları sırasıyla $X_i = \begin{bmatrix} 0.4 & -0.4 & 0.5 \end{bmatrix}^T$ ve $X_g = \begin{bmatrix} 0.4 & 0.4 & 0.5 \end{bmatrix}^T$ olarak belirlenmektedir. Emniyet için mesafe parametreleri ise $L_{safe} = 0.3m$, $L_{crit} = 0.2m$ ve $L_{danger} = 0.1m$ olarak tanımlanmaktadır. Yerel planlayıcının her iterasyonunda robot ucunun yer değiştirmesine de karşılık gelen varsayımsal kürenin yarıçapı, $r_h = 0.04m$ olacak şekilde ayarlanmaktadır. Her iterasyondaki örnek nokta sayısı ise $N_s = 40$ olarak seçilmektedir. Önerilen yöntem, bu planlama ortamında belirtilen koşullar üzerinden test edilmektedir. Robot hareketini simüle etmek için, eklemler üzerinde PD kontrolü sağlayan bir eklem uzayı hareket modeli kullanılmakta ve önerilen yöntemin (yani yerel planlayıcı) çıktıları olan $q_{s_i}^k$ eklem konfigürasyonları da bu hareket modeli için yol noktaları (ing. Waypoints) olarak kullanılmaktadır. Benzetimler için yörünge zaman adımı $T = 0.4s$ olarak belirlenmektedir. Bu süre, aynı zamanda uzaklık ölçümlerinin yapıldığı zaman aralıklarını da belirlemektedir.

6.2.1 Tek engel durumu

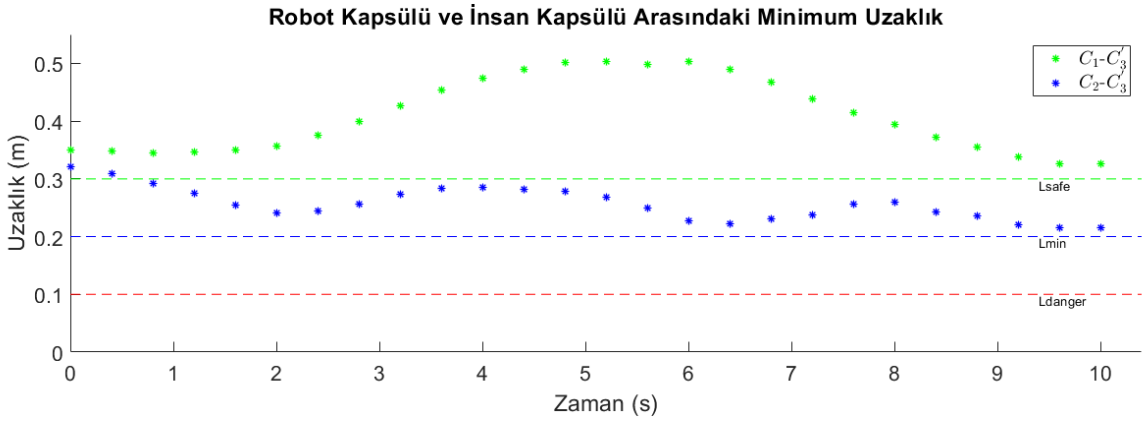
Önerilen yöntem, öncelikle planlama ortamındaki tek bir statik engel için test edilmektedir. Bu senaryoda, sol alt kol engel olarak kullanılmaktadır ve Şekil 6.14'te gösterildiği gibi robot çalışma alanına yakın bir şekilde konumlandırılmaktadır. Engellerden kaçınma ve uzaklık hesaplamaları için insan vücudunun sadece bu kısmı dikkate alınmaktadır. Diğer vücut kısımları için bu senaryoda uzaklık hesaplamaları yapılmamaktadır.

Optimizasyon probleminin ağırlıkları ise deneysel olarak $\alpha = 0.3$, $\beta = 0.4$ ve $\gamma = 0.3$ olacak şekilde belirlenmektedir. Buna göre, her bir iterasyonda örnek noktalar seçilirken, engellerden kaçınma amacı, hedefe ulaşma ve düzgün bir yol izlemeye göre ağır basmaktadır.

Tek engel içeren planlama ortamında gerçekleştirilen benzetim sonucu takip edilen yörünge Şekil 6.14'te gösterilmektedir. Burada, oluşturulan yol planına göre robot uç noktasının takip ettiği yörünge başlangıç noktasından hedef noktasına kırmızı çizgi ile gösterilmektedir. Bu benzetim sırasında ölçülen minimum uzaklık değerleri ise Şekil 6.15'te gösterilmektedir.

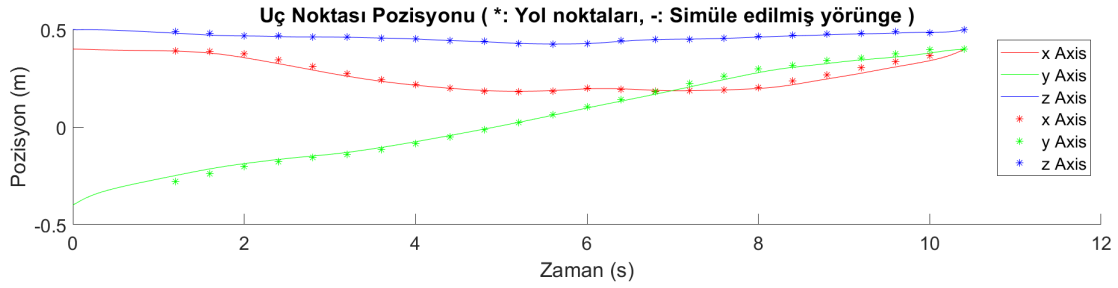


Şekil 6.14: Tek engel senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı

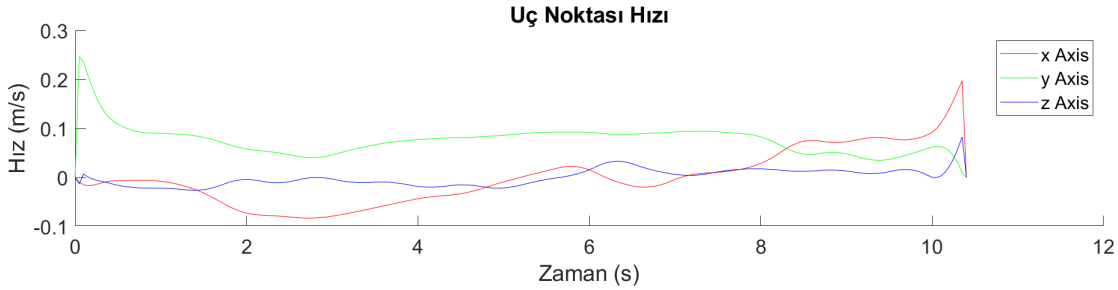


Şekil 6.15: Tek engel senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülü (sol alt kol C_3') arasındaki minimum uzaklık hesaplamaları

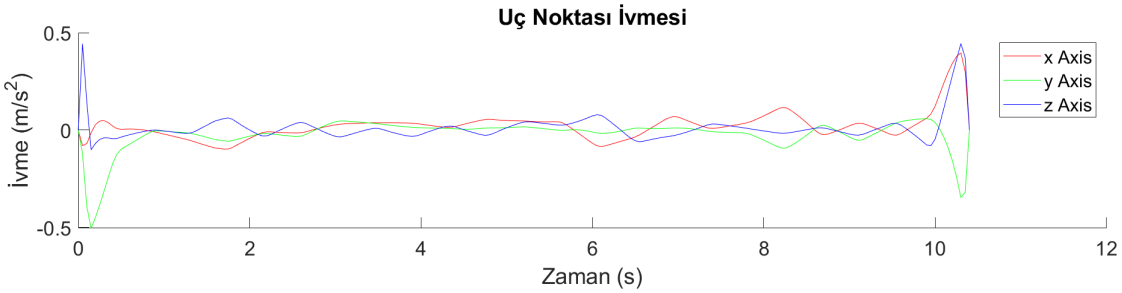
Gerçekleştirilen benzetim sırasında robot uç noktasının pozisyonu, hızı, ivmesi ve sarsımı (ing. Jerk) Şekil 6.16'da gösterilmektedir. Benzer şekilde, eklemlerin sahip olduğu pozisyon, hız, ivme ve sarsımlar ise Şekil 6.17'de gösterilmektedir.



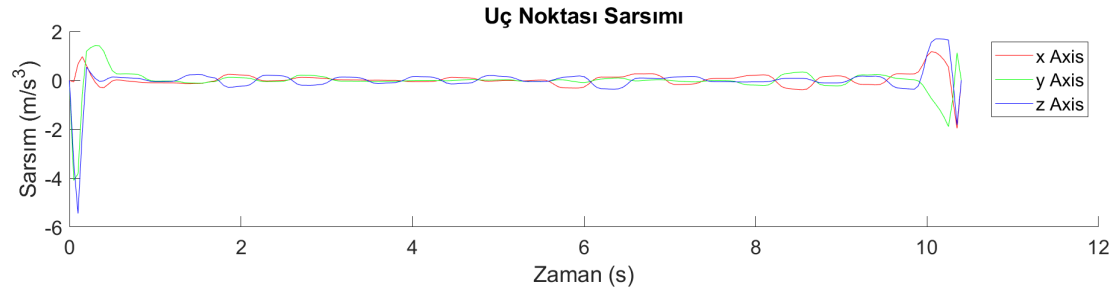
(a)



(b)

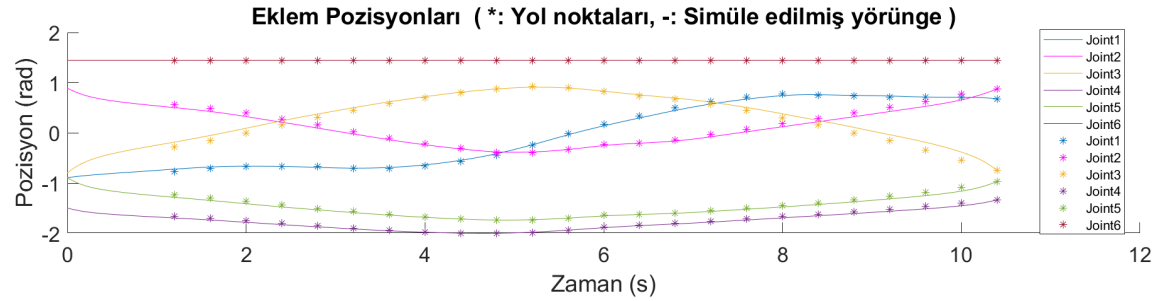


(c)

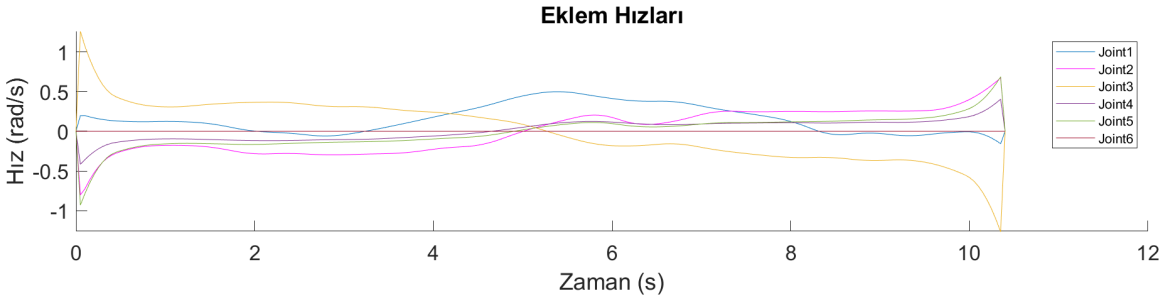


(d)

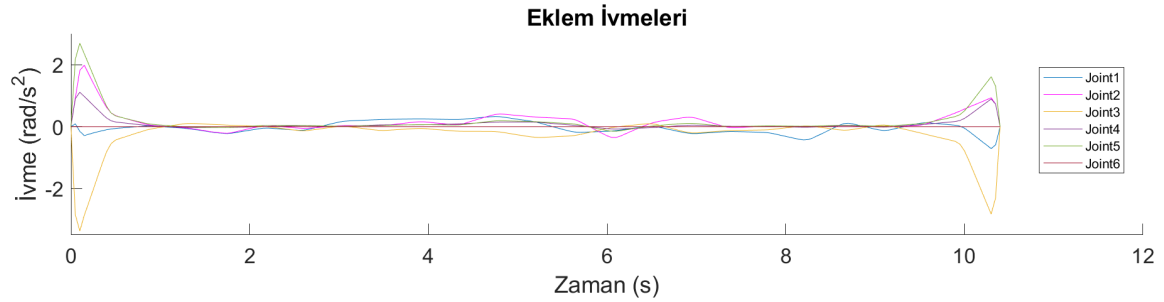
Şekil 6.16: Tek engel senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı



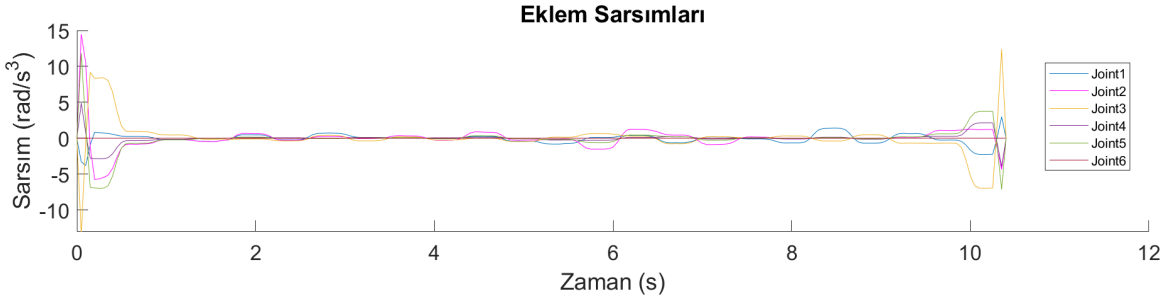
(a)



(b)



(c)



(d)

Şekil 6.17: Tek engel senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı

Benzetim sonuçlarından görüleceği üzere, robot ilk olarak küresel yol planını takip ederek hareketine başlamaktadır. Robot kapsülleri ile insan kapsülü arasında hesaplanan

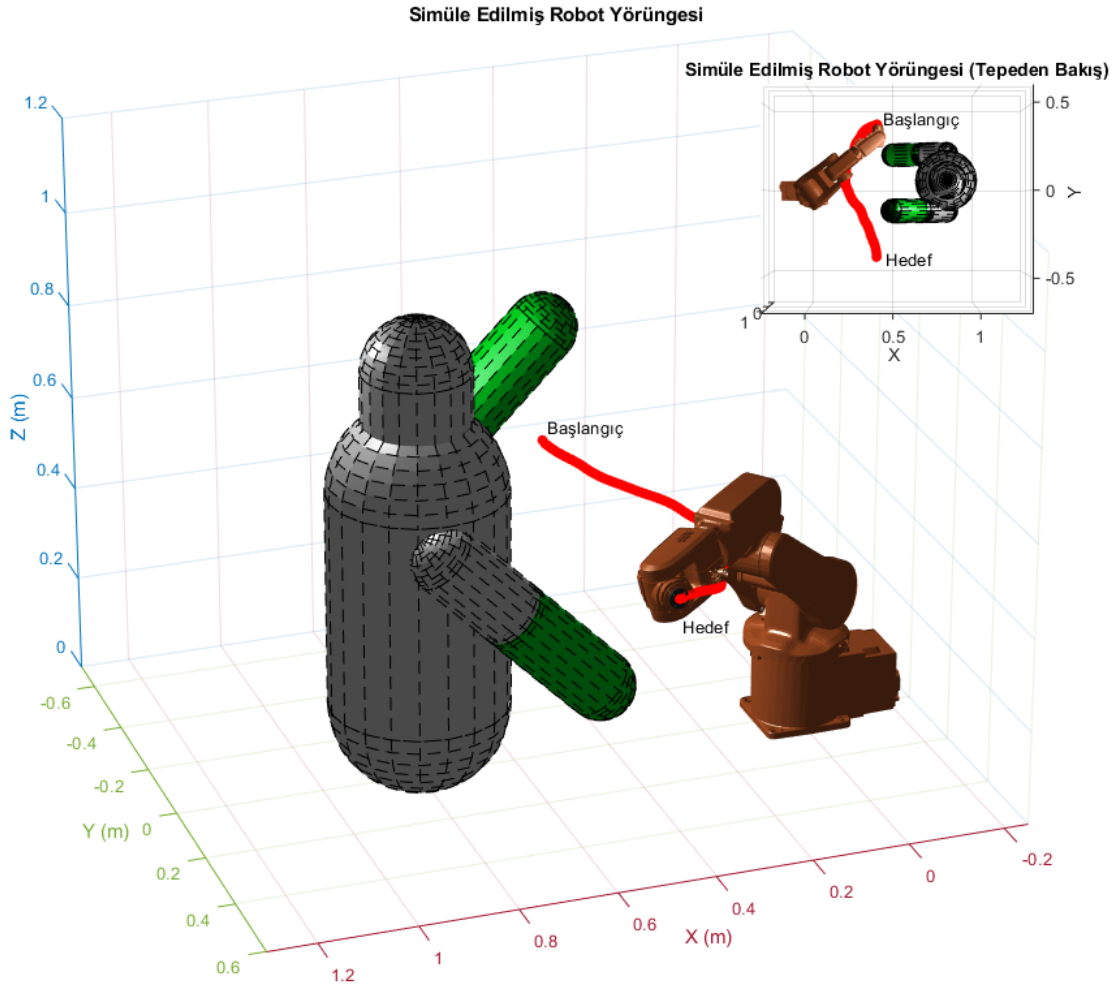
minimum uzaklık, L_{safe}'in altına düştüğünde ise küresel yol planı terkedilmekte ve robot yerel planlayıcı tarafından oluşturulan yol planını takip etmektedir. Elde edilen sonuçlarda da görüldüğü gibi, robot hedef noktasına ulaşırken insana L_{crit}'den daha fazla yaklaşmamaktadır. Bu durum, aynı zamanda robot hareketi boyunca örneklemenin yarım küre yüzeyinde gerçekleştirildiğini göstermektedir. Bir başka deyişle, robot çarpışmadan uzak emniyetli bir mesafede tutulmakta ve aynı zamanda hedefe daha hızlı ulaşma fırsatı sağlanmaktadır. Yörünge sonuçları, ayrıca robotun tek bir statik engelin varlığında engelden uzak durarak ve düzgün bir yol izleyerek hedefine ulaşabileceğini göstermektedir. Sınırlı ivme ve sarsım değerleri de, oluşturulan yol planının uygulanabilirliğini ortaya koymaktadır.

6.2.2 Çoklu engel durumu

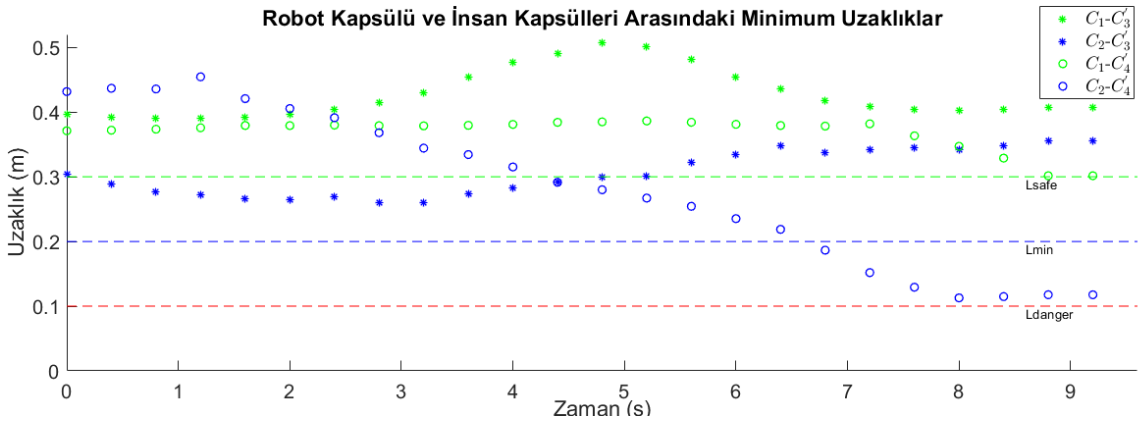
Bu bölümde, planlama ortamında birden fazla engelin varlığında gerçekleştirilen testin sonuçları anlatılmaktadır. Bir önceki senaryoya göre benzer koşullar korunsa da, artık planlama ortamında iki statik engel kullanılmaktadır. Önceki durumda olduğu gibi sol alt kol, bu senaryoda da engellerden biri olarak kabul edilmektedir. Buna ek olarak sağ alt kol da, ikinci engel olarak düşünülmektedir. Planlama ortamındaki kollardan biri, Şekil 6.18'de gösterildiği gibi robot çalışma alanının üstüne, bir diğeri de altına gelecek şekilde konumlandırılmaktadır.

Optimizasyon probleminin ağırlıkları ise bu senaryoda $\alpha = 0.05$, $\beta = 0.9$ ve $\gamma = 0.05$ olacak şekilde belirlenmektedir. Bir önceki senaryoya göre, engellerden kaçınma amacının ağırlığı daha da arttırılmaktadır.

Çoklu engel içeren planlama ortamında gerçekleştirilen benzetim sonucu takip edilen yörünge Şekil 6.18'de gösterilmektedir. Burada, oluşturulan yol planına göre robot uç noktasının takip ettiği yörünge başlangıç noktasından hedef noktasına kırmızı çizgi ile gösterilmektedir. Bu benzetim sırasında ölçülen minimum uzaklık değerleri ise Şekil 6.19'da gösterilmektedir.

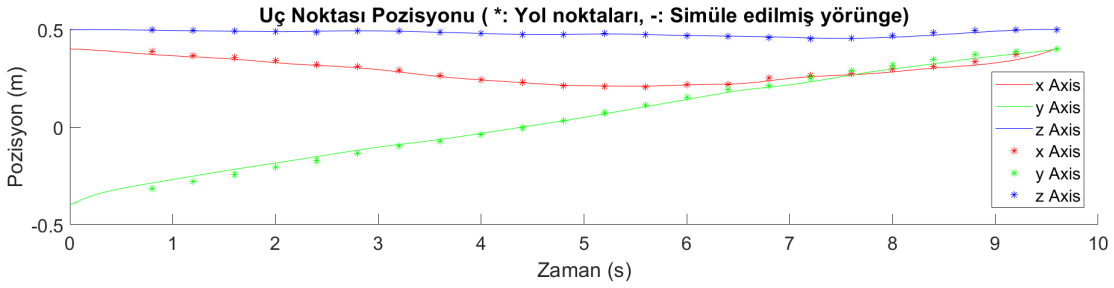


Şekil 6.18: Çoklu engel senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı

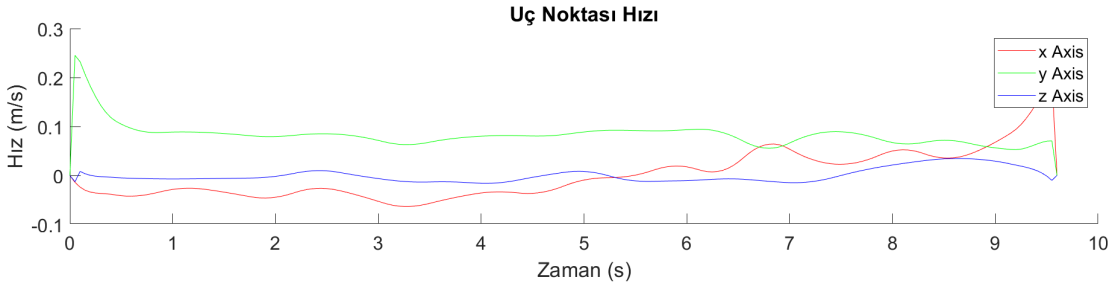


Şekil 6.19: Çoklu engel senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülleri (sol alt kol C_3', sağ alt kol C_4') arasındaki minimum uzaklık hesaplamaları

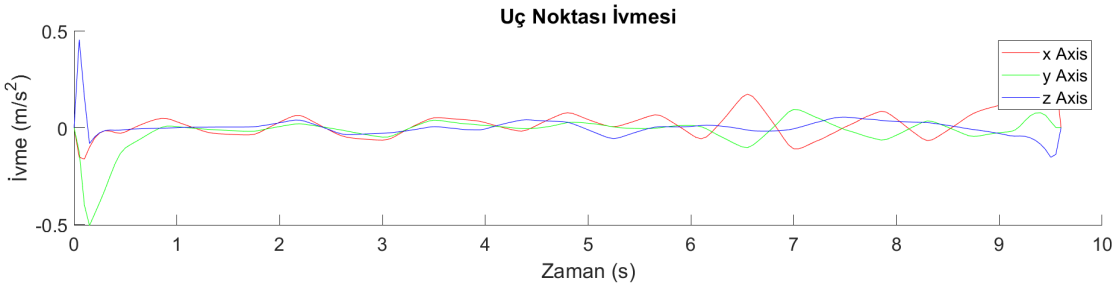
Gerçekleştirilen benzetim sırasında robot uç noktasının pozisyonu, hızı, ivmesi ve sarsımı (ing. Jerk) Şekil 6.20'de gösterilmektedir. Benzer şekilde, eklemlerin sahip olduğu pozisyon, hız, ivme ve sarsımlar ise Şekil 6.21'de gösterilmektedir.



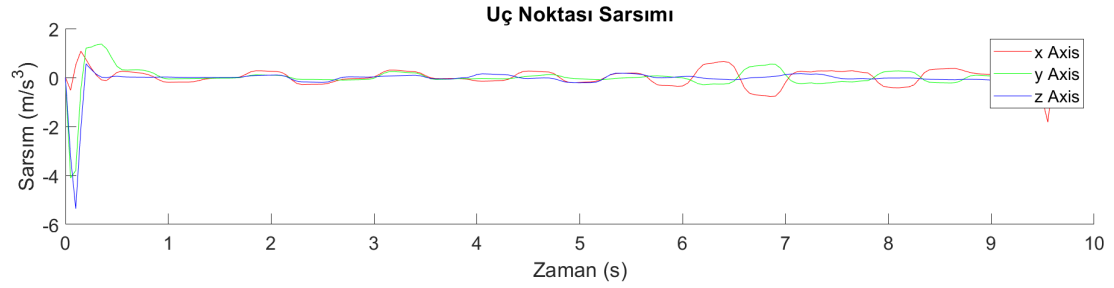
(a)



(b)

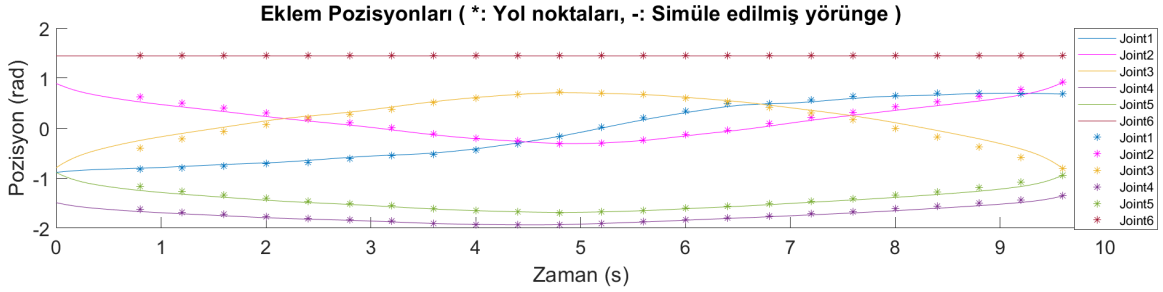


(c)

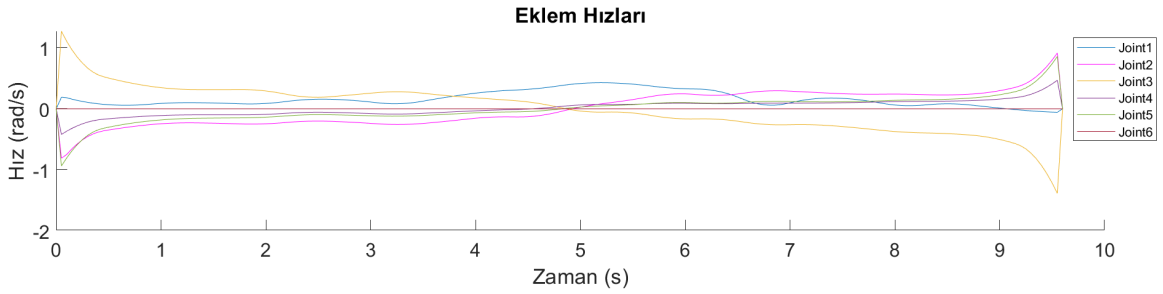


(d)

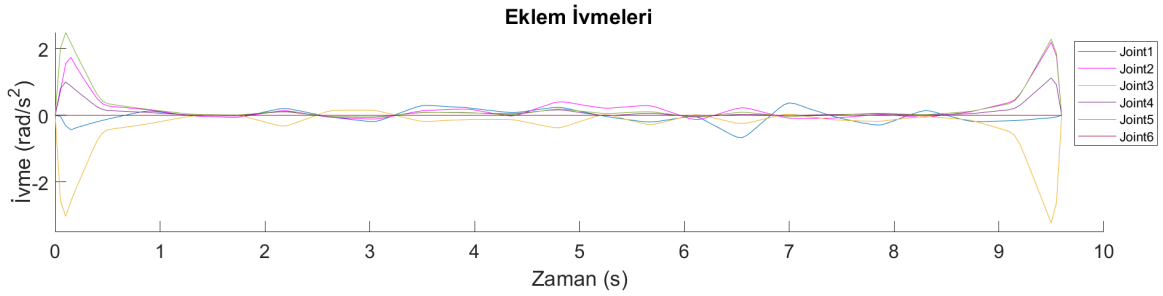
Şekil 6.20: Çoklu engel senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı



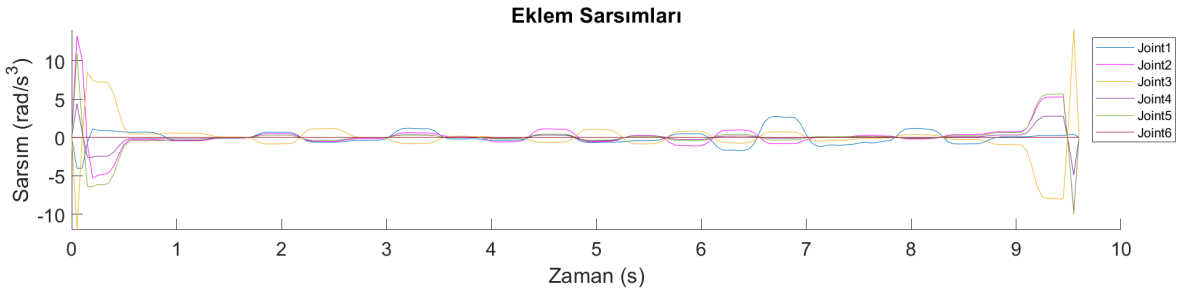
(a)



(b)



(c)

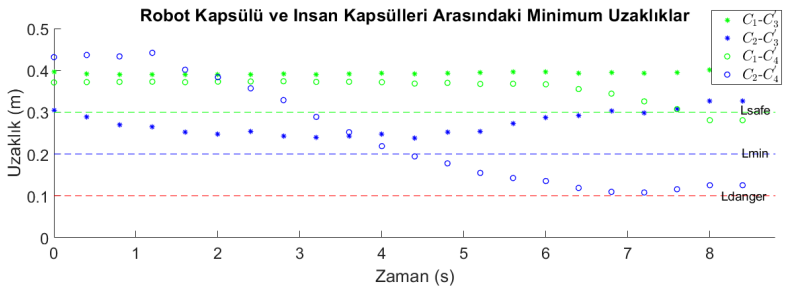


(d)

Şekil 6.21: Çoklu engel senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı

Benzetim sonuçlarında görüldüğü üzere, benzetim başladıktan hemen sonra hesaplanan uzaklıklardan biri L_{safe}'in altına düştüğü için küresel yol planı terk edilmektedir. Robot, simüle edilen hareketin geri kalanını yerel planlayıcı tarafından

oluşturulan plana göre gerçekleştirmektedir. Takip edilen yörünge, robotun planlanan yol noktalarını izleyerek düzgün bir hareket gerçekleştirdiğini göstermektedir. Belirlenen optimizasyon ağırlık parametrelerinin ($\beta = 0.9$) etkisi ile robot, hareketinin uzun bir bölümünde engellerle (en azından L_{crit} sınırının üzerinde) arasındaki emniyetli mesafeyi korumaktadır. Robot, hareketinin sonuna doğru sağ alt kola yaklaşsa da, bu durumun engelin hedef konuma çok yakın olmasından kaynaklandığı anlaşılmaktadır. Aynı koşullar altında gerçekleştirilen ancak $\alpha = 0.9$, $\beta = 0.05$ ve $\gamma = 0.05$ için başka bir benzetimde hesaplanan minimum uzaklık değerleri de Şekil 6.22'de gösterilmektedir. Bu sonuç, ağırlıkların etkisini yorumlamaya yardımcı olmaktadır. Görüldüğü gibi robot, hareketi sırasında engellere bir önceki duruma göre ($\beta = 0.9$) daha fazla yaklaşmakta, ancak hedefine daha hızlı ulaşmaktadır (Bkz. Şekil 6.19). Elbette karşılaştırma yaparken örneklemenin stokastik bir süreç olduğu ve bu nedenle özellikle örnek sayısının yetersiz olduğu durumlarda sonuçlara az da olsa etki edebileceği unutulmamalıdır.

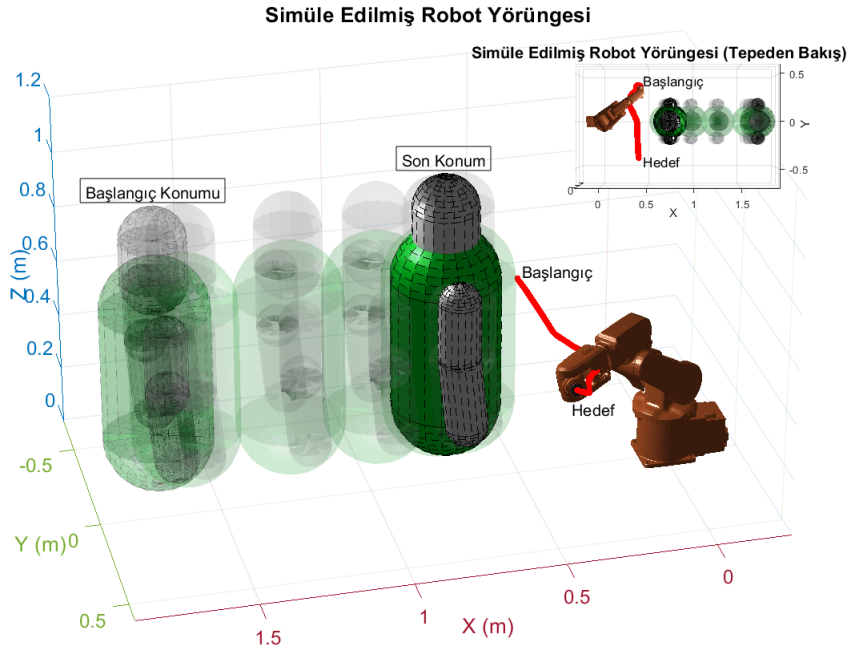


Şekil 6.22: $\alpha = 0.9$, $\beta = 0.05$ ve $\gamma = 0.05$ için gerçekleştirilen benzetimde hesaplanan minimum uzaklıklar

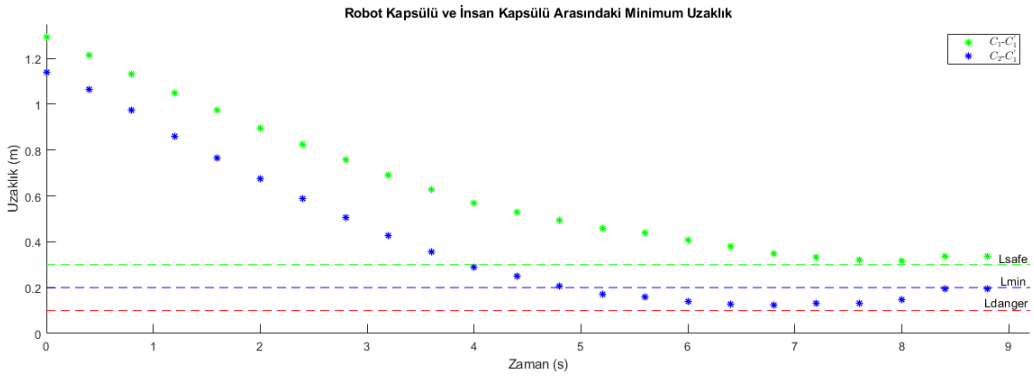
6.2.3 Hareketli engel durumu

Son olarak, önerilen yöntemin performansı, planlama ortamında hareketli bir engelin varlığında test edilmektedir. Bunun için, hareketli engelin farklı vücut kısımları olduğu iki farklı test gerçekleştirilmektedir. İlk testte, insan gövdesinin hareketli engel olduğu kabul edilmektedir. Ortamdaki konumu zamana göre değişecek olan engelin hareketini modellemek için ise sinüzoidal bir fonksiyon kullanılmaktadır. Buna göre, insan gövde kapsülünün merkezine ait Kartezyen koordinatlar deneysel olarak $X_{OC_1'} = \left[1.7 - \left|\sin\left(\frac{\pi}{14}t\right)\right| \quad 0.0 \quad 0.6\right]$ şeklinde belirlenmektedir. Bu şekilde insan modeli, benzetimin başlamasıyla beraber robotun çalışma alanına başlangıç konumundan ($t = 0$) itibaren yaklaşmakta ve belirli bir süre sonra aynı şekilde uzaklaşmaktadır. Benzetim tamamlanır tamamlanmaz ise Şekil 6.23'te gösterildiği gibi son konumuna ulaşmaktadır. Bu senaryoda, optimizasyon ağırlık parametreleri de $\alpha = 0.3$, $\beta = 0.4$ ve $\gamma = 0.3$ olacak şekilde belirlenmektedir. Hareketli gövde kapsülü içeren planlama

ortamında gerçekleştirilen benzetim sonucu takip edilen yörünge Şekil 6.23'te gösterilmektedir. Burada, oluşturulan yol planına göre robot uç noktasının takip ettiği yörünge başlangıç noktasından hedef noktasına kırmızı çizgi ile gösterilmektedir. Bu benzetim sırasında ölçülen minimum uzaklık değerleri ise Şekil 6.24'te gösterilmektedir.

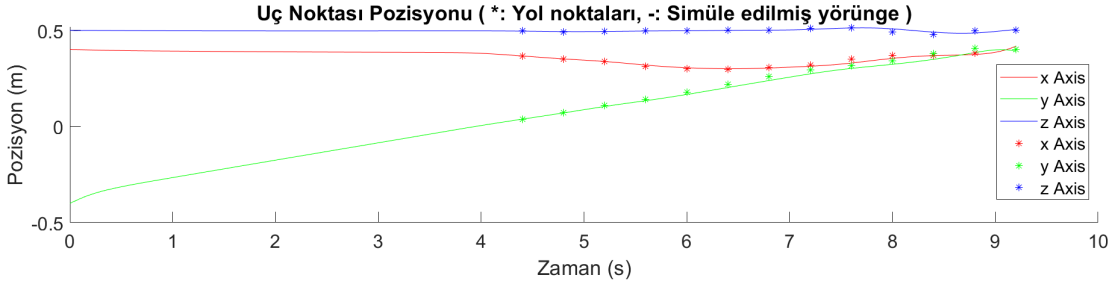


Şekil 6.23: Hareketli engel (gövde) senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı

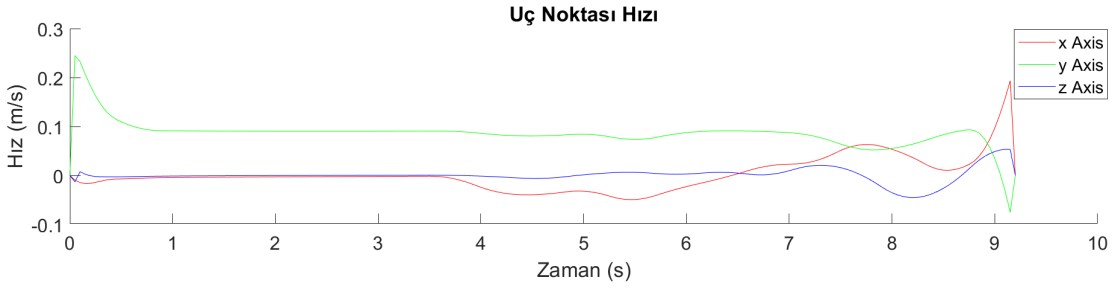


Şekil 6.24: Hareketli engel (gövde) senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülü (gövde C_1') arasındaki minimum uzaklık hesaplamaları

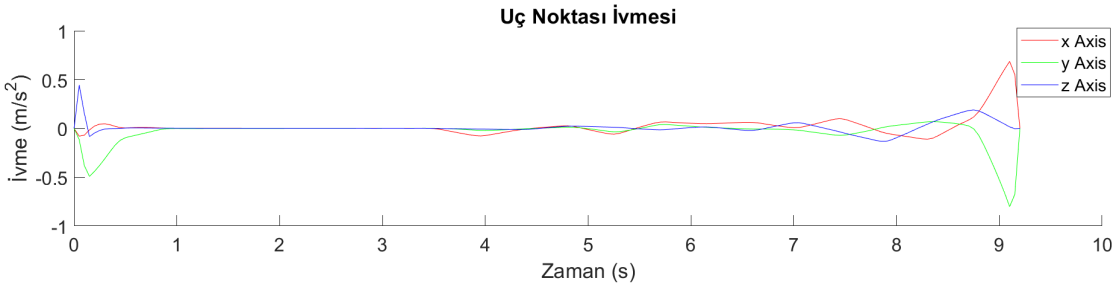
Gerçekleştirilen benzetim sırasında robot uç noktasının pozisyonu, hızı, ivmesi ve sarsımı (ing. Jerk) Şekil 6.25'te gösterilmektedir. Benzer şekilde, eklemlerin sahip olduğu pozisyon, hız, ivme ve sarsımlar ise Şekil 6.26'da gösterilmektedir.



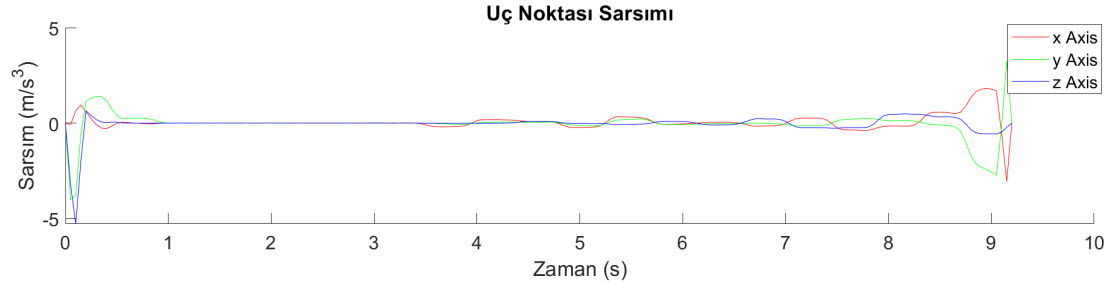
(a)



(b)

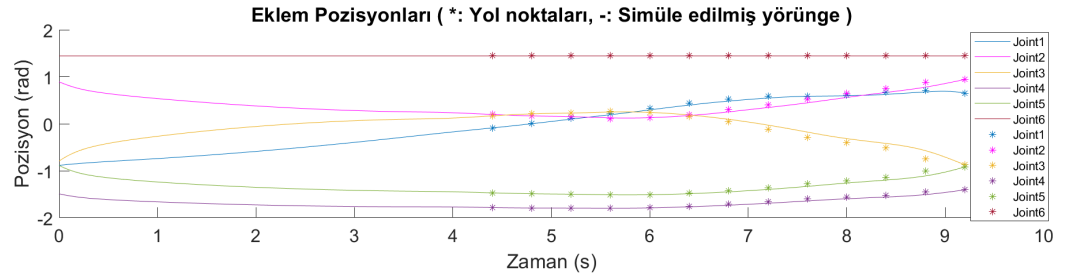


(c)

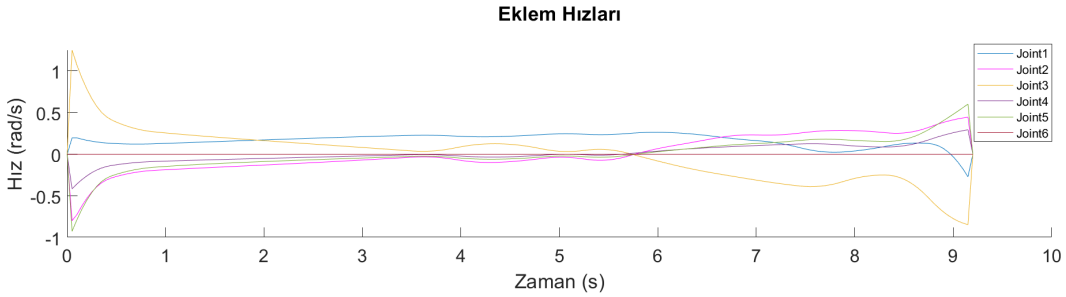


(d)

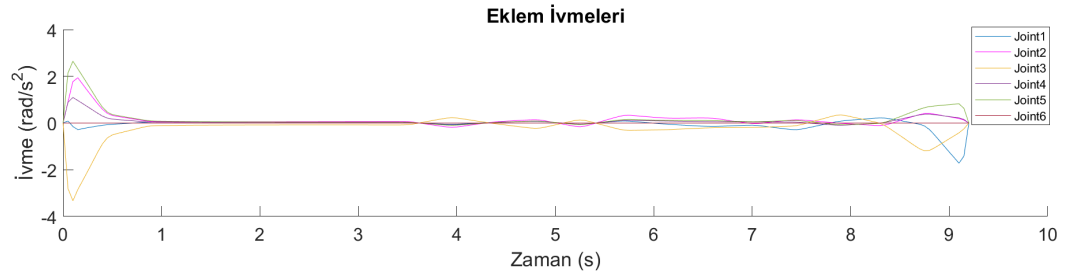
Şekil 6.25: Hareketli engel (gövde) senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı



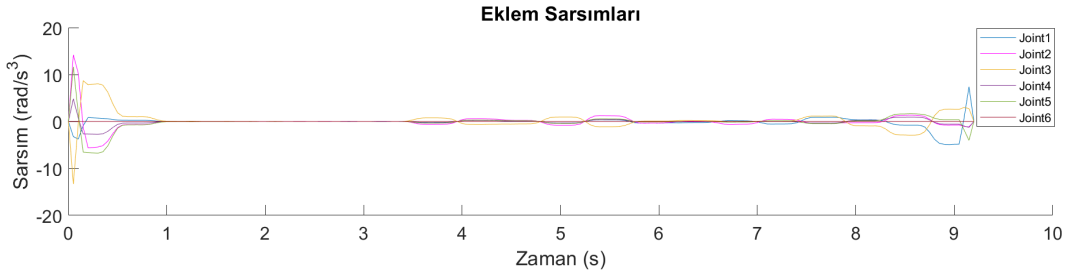
(a)



(b)



(c)

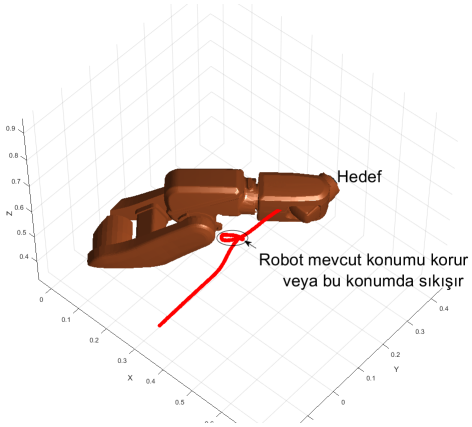


(d)

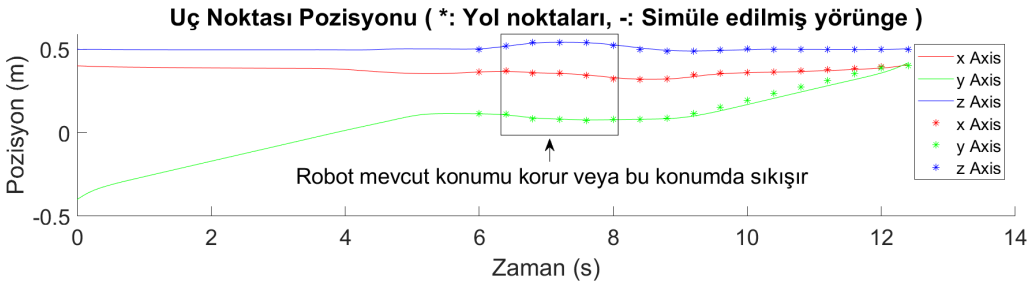
Şekil 6.26: Hareketli engel (gövde) senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı

Sonuçlara göre robot, gövde kapsülü ile arasında hesaplanan uzaklık L_{safe}'in altına düşene kadar küresel yol planını takip etmektedir. Yerel planlayıcı devreye girdikten sonra ve insan yaklaşmaya devam ederken de, mesafeyi L_{danger}'ın üzerinde tutarak hedefine ulaşmaktadır. Yerel planlayıcının aktif olduğu zamanın çoğunda, örnekleme yarım küre

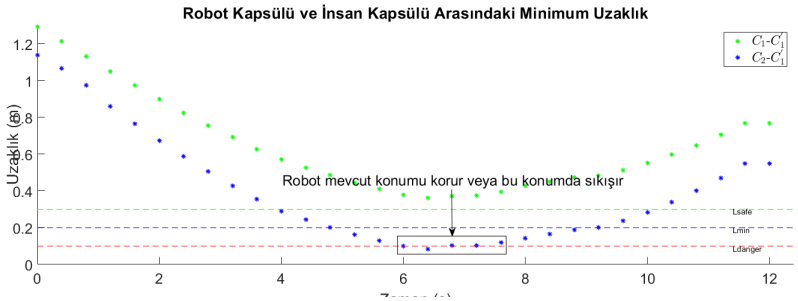
yerine küre üzerinde gerçekleştirilmektedir. Bu durum, örneklemedeki hedefe ulaşma eğilimini ortadan kaldırdığından, robotu engellerden uzak tutmak ve kritik durumlarda çarpışmadan kaçınmak için daha fazla fırsat sağlanmaktadır. Elbette bunun yanında, aynı zamanda hedefe ulaşma imkanı da korunmaktadır. Öyle ki, algoritma aynı koşullar altında ancak sadece yarım küre yüzeyinde örnekleme ile çalıştırıldığında robot, Şekil 6.27'de gösterildiği gibi, ya mevcut konumunu koruyacak ya da engel uzaklaşana kadar bu konumda sıkışıp kalacaktır. Önerilen yöntemde, koşullara bağlı olarak bir küre veya yarım küre yüzeyinde örnekleme yapılması gerektiği, bu şekilde ortaya konmaktadır.



(a)



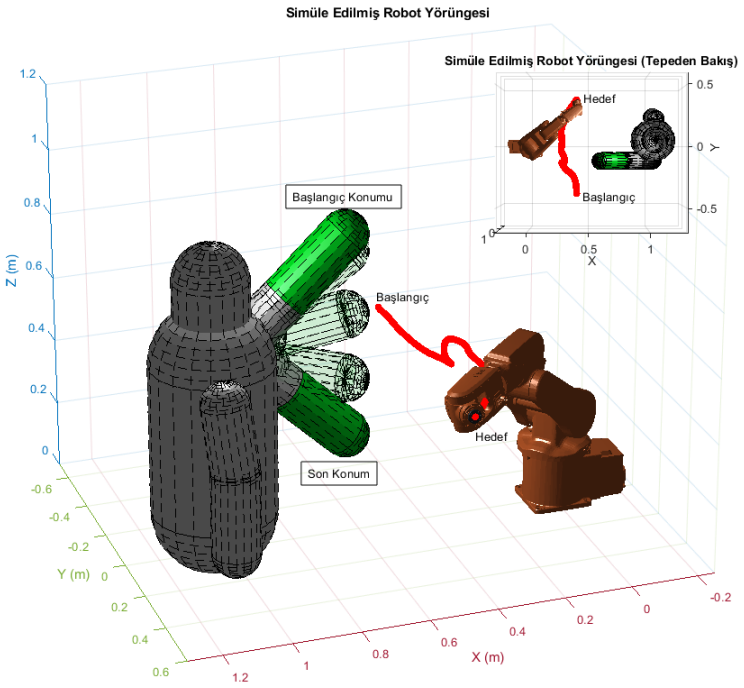
(b)



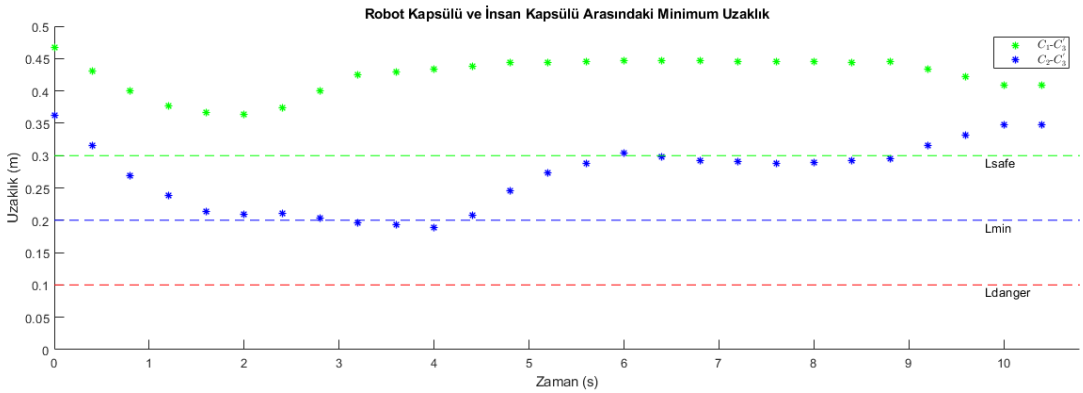
(c)

Şekil 6.27: Önerilen yöntem sadece yarım küre yüzeyinde örnekleme yapılarak kullanıldığında benzetim sonuçları: (a) Simüle edilen yörünge, (b) Robot ucunun konumu, (c) Minimum uzaklık hesaplamaları

Hareketli engel senaryosu için aynı koşullar altında sol alt kolun hareketli engel olarak kabul edildiği başka bir test daha yapılmaktadır. Çoklu engel senaryosundan farklı olarak sol alt kol, bu sefer Şekil 6.28'de gösterildiği gibi robotun çalışma alanında başlangıç konumundan (–z yönünde) aşağı doğru hareket etmektedir. Benzetim sırasında robot kapsülleri ile sol alt kol arasındaki hesaplanan uzaklıklar Şekil 6.29'da gösterilmektedir.

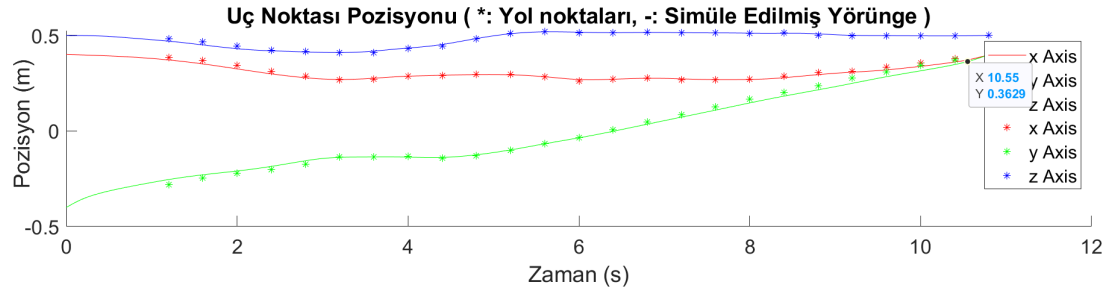


Şekil 6.28: Hareketli engel (sol alt kol) senaryosu için planlama ortamında benzetimi gerçekleştirilmiş yol planı

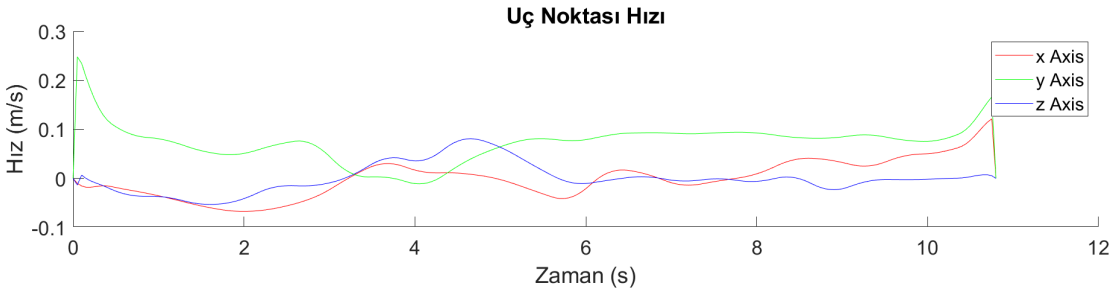


Şekil 6.29: Hareketli engel (sol alt kol) senaryosu için iki robot kapsülü (C_1 ve C_2) ile insan kapsülü (gövde C_3') arasındaki minimum uzaklık hesaplamaları

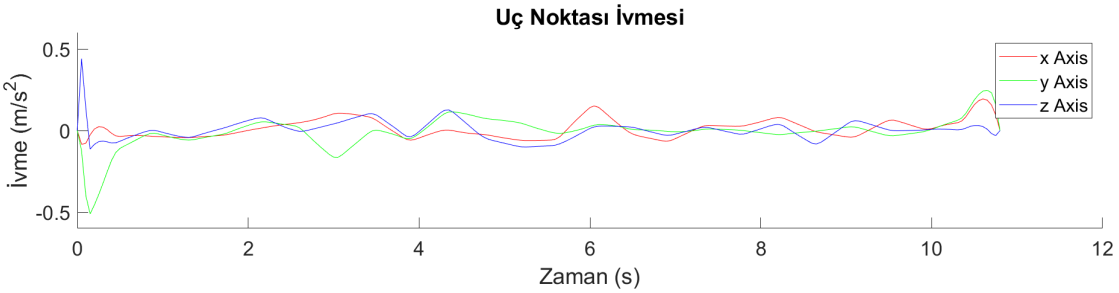
Gerçekleştirilen benzetim sırasında robot uç noktasının pozisyonu, hızı, ivmesi ve sarsımı (ing. Jerk) Şekil 6.30'da gösterilmektedir. Benzer şekilde, eklemlerin sahip olduğu pozisyon, hız, ivme ve sarsımlar ise Şekil 6.31'de gösterilmektedir.



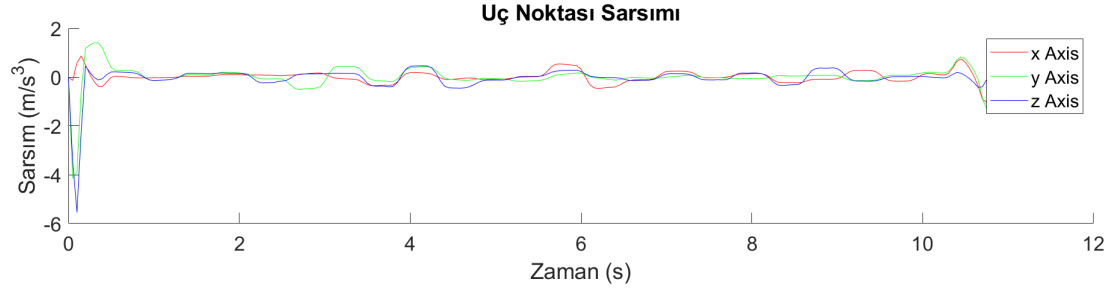
(a)



(b)

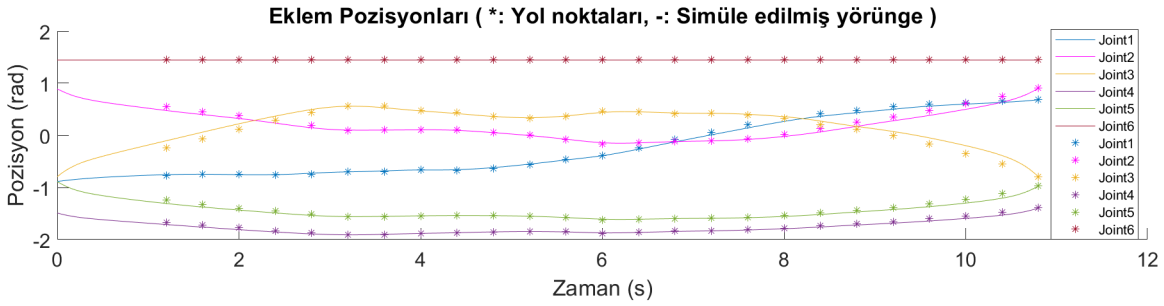


(c)

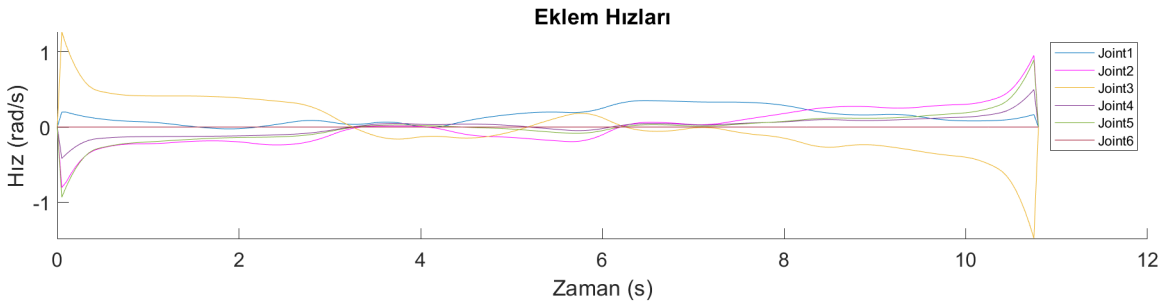


(d)

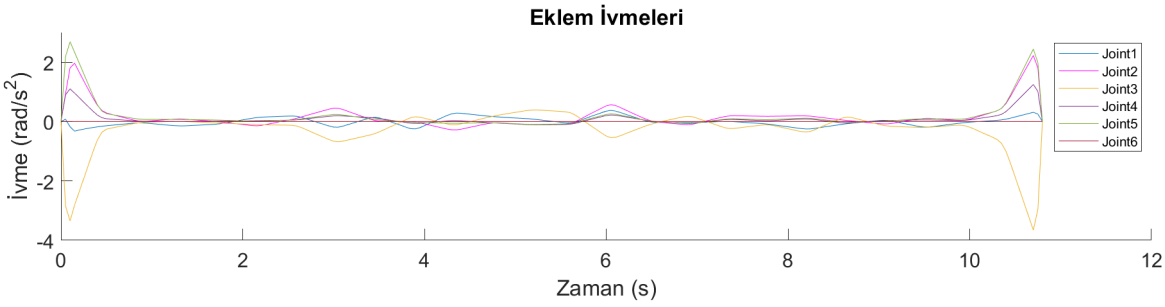
Şekil 6.30: Hareketli engel (sol alt kol) senaryosu için robot uç noktasının: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı



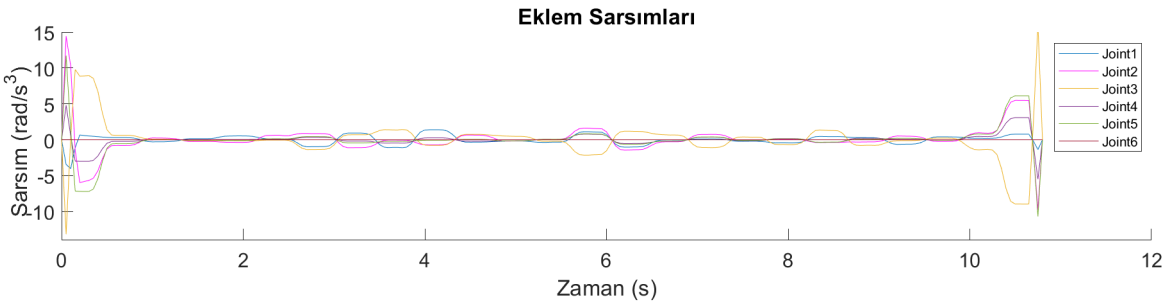
(a)



(b)



(c)



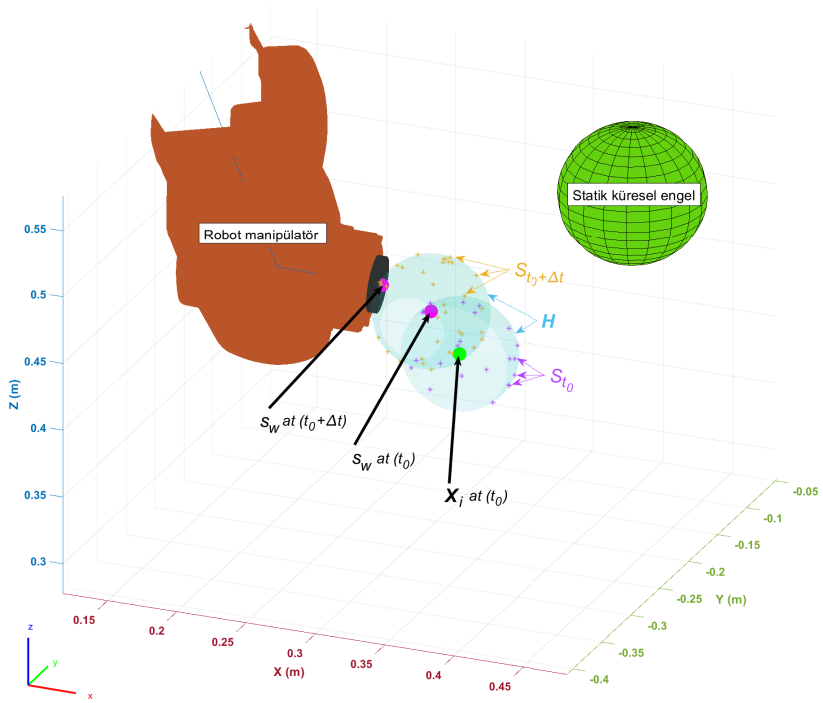
(d)

Şekil 6.31: Hareketli engel (sol alt kol) senaryosu için eklemlerin: (a) pozisyonu, (b) hızı, (c) ivmesi, (d) sarsımı

6.2.4 Ağırlık parametrelerinin analizi

Bu tez çalışmasında ayrıca ağırlıkların optimizasyon problemine etkisini daha iyi anlamak için çeşitli testler de yapılmaktadır. Bunu göstermek için Şekil 6.32'deki gibi iki iterasyon adımından oluşan bir planlama şeması tanımlanmakta ve ağırlık parametre kümesindeki $\{\alpha, \beta, \gamma\}$ değerleri için ikinci iterasyon adımında hangi örnek noktanın seçildiği incelenmektedir. Aynı planlama şeması aynı parametre değerleri için en az 50 kez tekrarlanmakta ve daha sonra kümedeki ağırlıklar değiştirilmektedir (n arttıkça). Üç parametrenin her birinin etkilerini ayrı ayrı gözlemleyebilmek için toplamda üç farklı deneme yapılmaktadır. Üç denemeye ait ağırlık parametre kümeleri (sırasıyla T_α, T_β ve T_γ), $n = 1, \ldots, 19$ için aşağıdaki gibi belirlenmektedir:

- $T_\alpha(n) = \left\{\alpha = \sum_1^n 0.05, \quad \beta = \dfrac{1-\alpha}{2}, \quad \gamma = \dfrac{1-\alpha}{2}\right\}$
- $T_\beta(n) = \left\{\alpha = \dfrac{1-\beta}{2}, \quad \beta = \sum_1^n 0.05, \quad \gamma = \dfrac{1-\beta}{2}\right\}$
- $T_\gamma(n) = \left\{\alpha = \dfrac{1-\gamma}{2}, \quad \beta = \dfrac{1-\gamma}{2}, \quad \gamma = \sum_1^n 0.05\right\}$

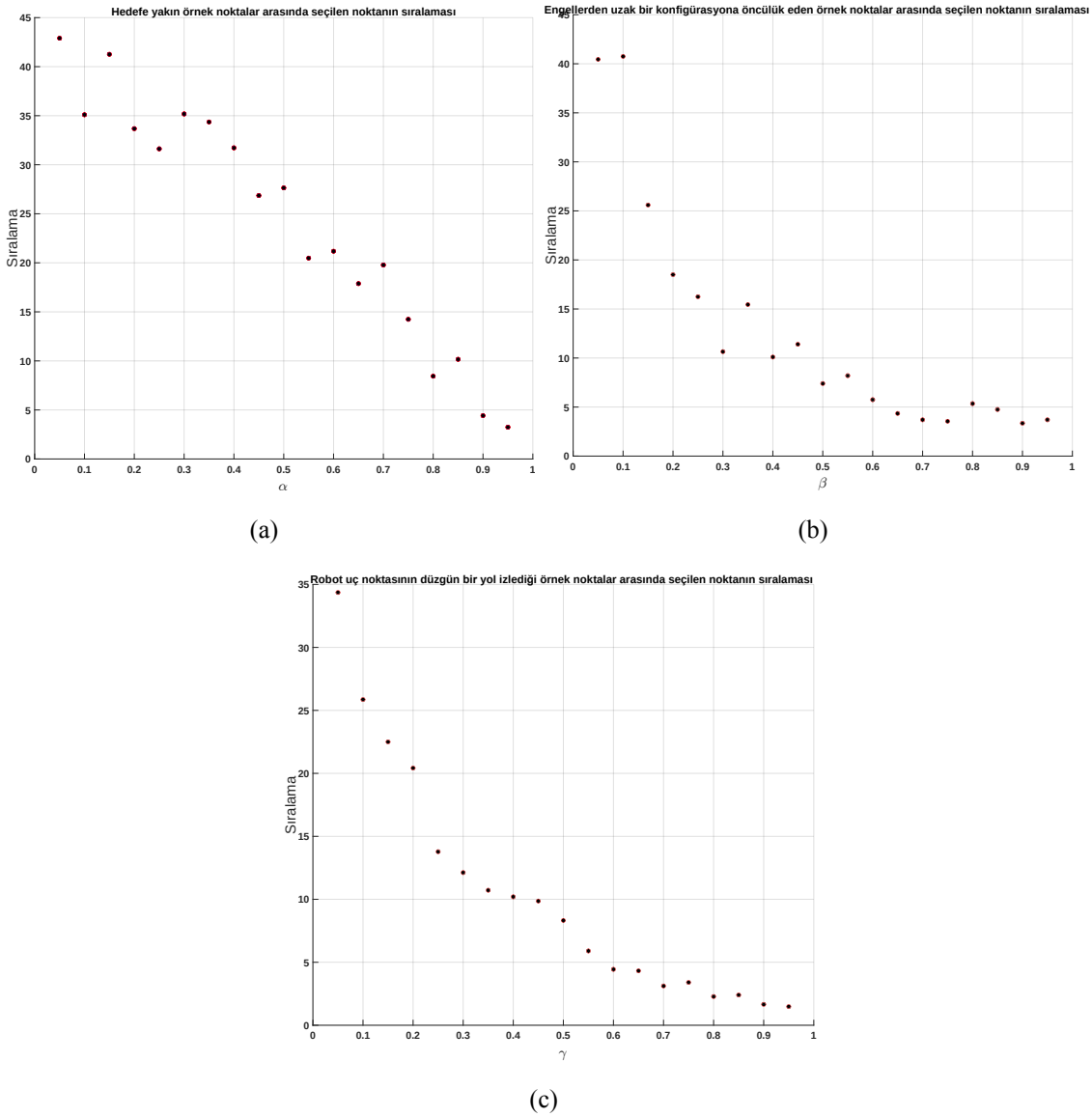


Şekil 6.32: Ağırlık parametrelerinin analizi sırasında planlama şeması

Buna göre, denemelerin ilkinde α'nın, ikincisinde β'nın ve üçüncüsünde γ'nın etkisi incelenmektedir. Planlama şemasında, uç noktasının başlangıç konumu X_i, birinci iterasyon adımında (yani t_0 anında) seçilen yol noktası s_w ve robota yakın statik küresel bir engel, Şekil 6.32'de gösterildiği gibi önceden tanımlanmaktadır. Bundan sonra, ikinci iterasyon adımında (yani $t_0 + \delta t$ anında), örnek noktalar kümesi S_t'deki hangi örnek noktanın yol noktası s_w olarak seçildiğini incelemek için algoritma birçok kez çalıştırılmaktadır.

Bu denemeleri gerçekleştirmenin arkasındaki amaç, optimizasyon sırasında bir ağırlık parametresinin seçilen yol noktası üzerinde ne kadar etkisi olduğunu görmektir. Örneğin α değeri arttığında, seçilen örnek noktanın robot uç noktasını hedefe daha fazla yaklaştıracak noktalardan biri olması beklenir. Aynı şekilde β değeri arttığında, seçilen örnek noktanın engellerden daha uzak olan robot konfigürasyonlarından birine yönlendirmesi beklenir. Benzer şekilde γ değeri arttığında, seçilen örnek noktanın robot uç noktasının daha düzgün bir yol izleyebileceği konumlardan biri olması beklenir. Bunu değerlendirmek için yapılan denemelere ait sonuçlar Şekil 6.33'te gösterilmektedir.

Denemeler sırasında, örnek noktası sayısı $N_s = 100$ olarak ayarlanmaktadır. Bir ağırlık parametre kümesinde belirlenen değerler için de algoritma, 50 defa çalıştırılmakta ve seçilen noktanın sıralaması bunların ortalaması şeklinde ifade edilmektedir. Örneğin ilk denemede, $n = 1$ için ağırlık parametre değerleri $\alpha = 0.05$, $\beta = 0.475$ ve $\gamma = 0.475$ olmaktadır ve algoritma 50 defa koşulmaktadır. Bu denemelerin her biri sırasıyla $n = \{1, \ldots, 19\}$ için gerçekleştirilmektedir. Şekil 6.33(a), değişen α değeri için, hedefe en yakın örnek noktalar arasında ikinci iterasyon adımında seçilen noktanın sıralamasını göstermektedir. Şekil 6.33(b), değişen β değeri için, ikinci iterasyon adımında engellerden uzakta bir robot konfigürasyonuna öncülük eden örnek noktalar arasında seçilen noktanın sıralamasını göstermektedir. Şekil 6.33(c) ise, γ değeri ile ikinci iterasyon adımında seçilen noktanın, uç noktasını en düzgün yolu izlediği örnek noktalar arasındaki sıralaması arasındaki ilişkiyi göstermektedir. Sonuçlar, ilgili ağırlık parametresinin değeri arttıkça onunla ilgili fonksiyonun önceliğinin de arttığını göstermektedir.

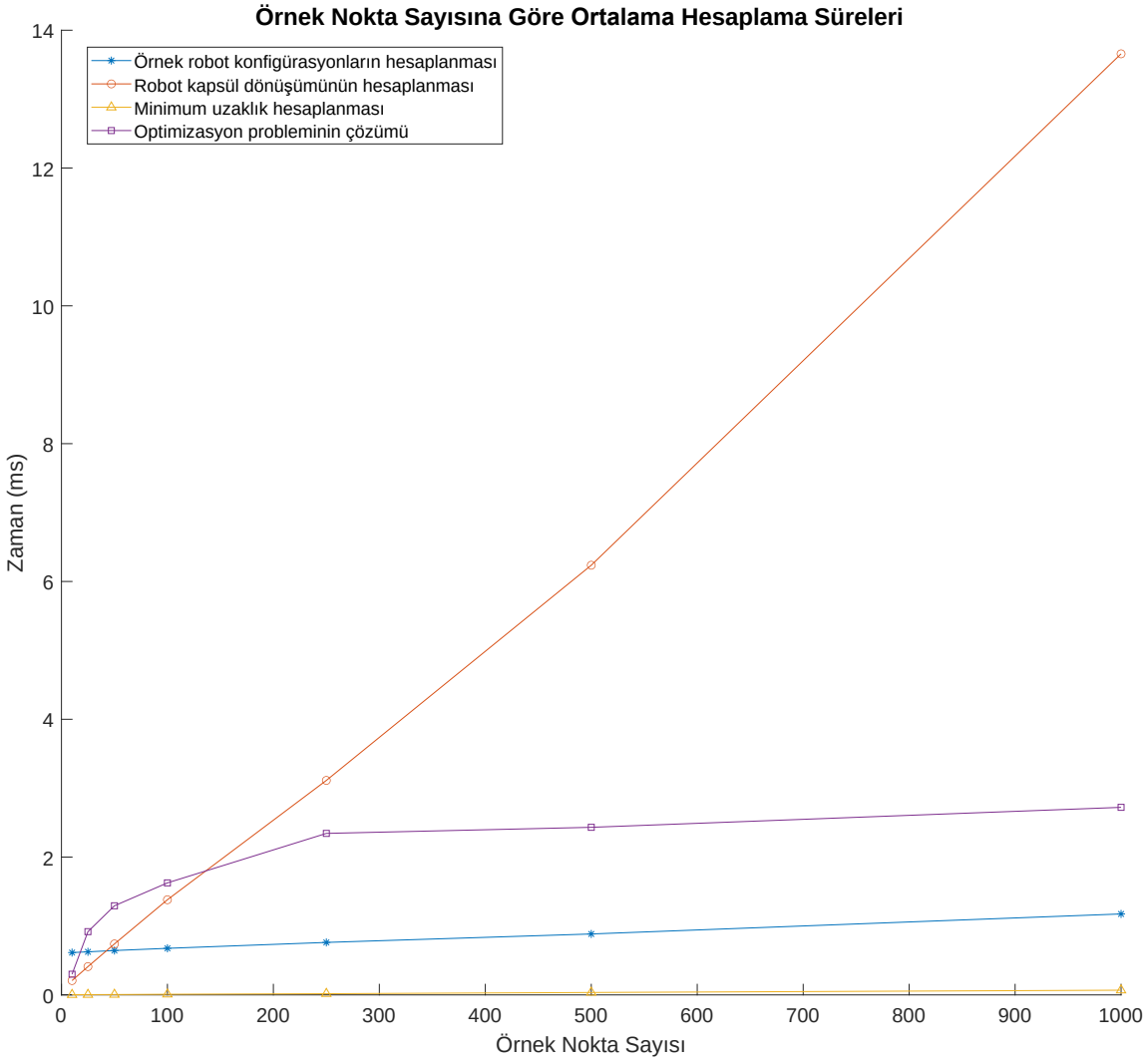


Şekil 6.33: Ağırlık parametrelerindeki değişime göre seçilen yol noktasının sıralaması: (a) hedefe en yakın yol noktaları arasındaki sıralama, (b) engellerden uzak bir konfigürasyona taşıyan yol noktaları arasındaki sıralama, (c) düzgün bir yol izlenmesine öncülük eden yol noktaları arasındaki sıralama

6.2.5 Hesaplama sürelerinin analizi

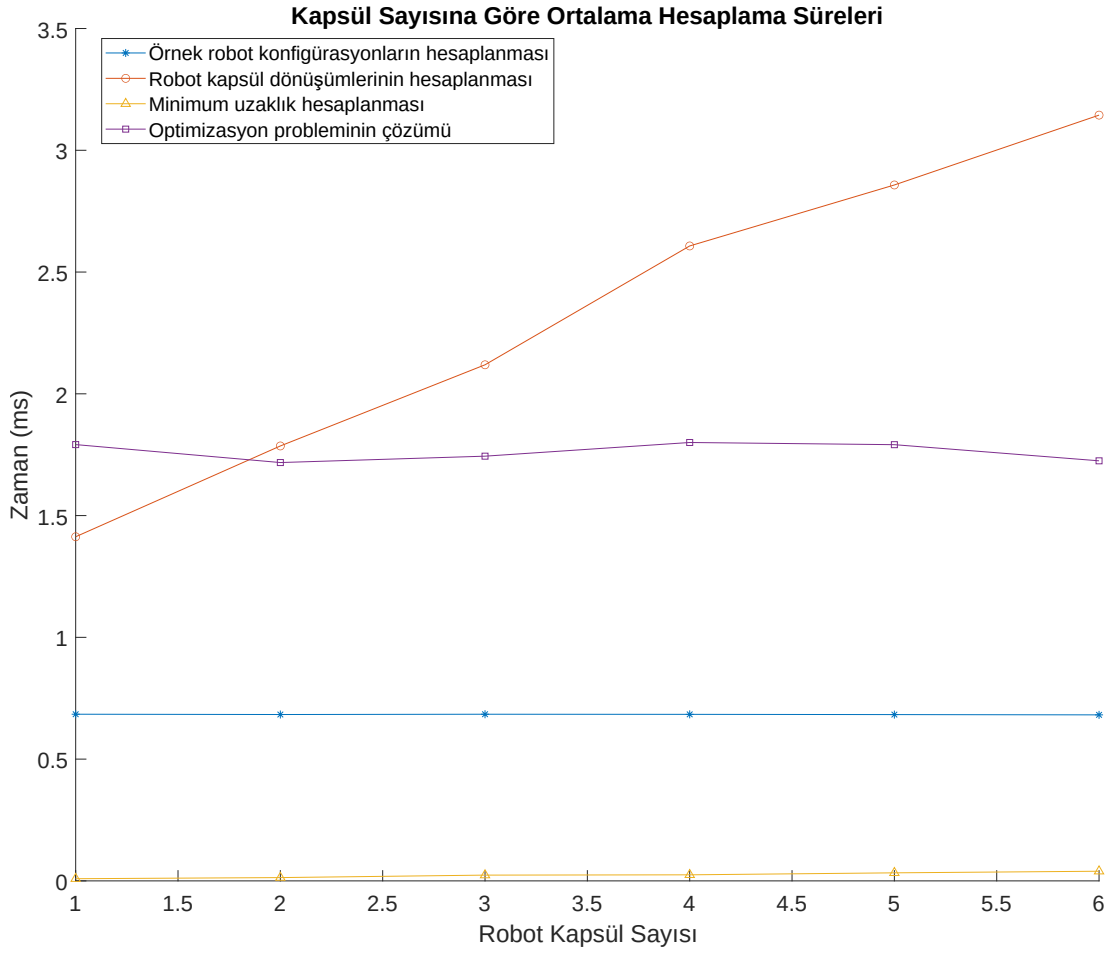
Önerilen yöntem, aynı zamanda hesaplama süreleri açısından da incelenmektedir. Hesaplama sürelerine etki eden en önemli faktörlerin örnek nokta sayısı ve kapsül sayısı olduğu göz önünde bulundurulduğunda, gerçekleştirilen deneylerle bu faktörlerin hesaplama süreleri üzerindeki etkisi gösterilmektedir. Söz konusu hesaplamalar, örnek robot konfigürasyonlarının hesaplanması, robot kapsül dönüşümünün hesaplanması, minimum uzaklığın hesaplanması ve optimizasyon probleminin çözümünden oluşmaktadır.

Bu deneylerden ilkinde, farklı nokta sayılarının hesaplama süreleri üzerindeki etkisi incelenmektedir. Bunun için, planlama ortamı bir insan kapsülü ve bir robot kapsülü içerecek şekilde oluşturulmaktadır. Minimum uzaklık hesaplamaları da, bu iki kapsül arasında FCL 0.7.0 kütüphanesindeki GJK implementasyonu kullanılarak gerçekleştirilmektedir (FCL, 2020a). Örnek nokta sayısı sırasıyla $N_s = 10, 25, 50, 100, 250, 500, 1000$ olacak şekilde, algoritma birden fazla kez çalıştırılmakta ve ölçülen hesaplama sürelerinin ortalaması alınmaktadır. Elde edilen sonuçlar Şekil 6.34'te gösterilmektedir. Sonuçlarda da görüleceği üzere örnek nokta sayısındaki artış, tüm hesaplama süreçlerine ait ortalama sürelerde artışa neden olmaktadır. En büyük artış, kapsül dönüşümlerinin hesaplanmasında yaşanmaktadır. $N_s = 1000$ için ortalama toplam süre ise 17.6231 ms olarak görülmektedir. Bu durum, önerilen yöntemin yüksek örnek nokta sayılarında dahi gerçek zamanlı uygulamalar için uygun hesaplama süreleri sağladığını göstermektedir.



Şekil 6.34: Örnek nokta sayısına göre ortalama hesaplama süreleri

Kapsül sayılarının hesaplama süreleri üzerindeki etkisi ise bir başka deneyle gösterilmektedir. Bu deney için, planlama ortamındaki örnek nokta sayısı $N_s = 100$ ve insan kapsül sayısı $n_h = 1$ olacak şekilde belirlenmektedir. Robot kapsül sayıları sırasıyla $n_r = 1, 2, 3, 4, 5, 6$ olacak şekilde algoritma birden fazla kez çalıştırılmakta ve ölçülen hesaplama sürelerinin ortalaması alınmaktadır. Elde edilen sonuçlar Şekil 6.35'te gösterilmektedir. Sonuçlarda da görüleceği üzere robot kapsül sayısındaki artış, sadece robot kapsül dönüşümlerinin ve minimum uzaklıkların hesaplanmasına ait ortalama sürelerde belirgin bir artışa neden olmaktadır. Örnek konfigürasyonların hesaplanma süresinde bir değişim olmamakla birlikte, optimizasyon probleminin çözümü için de belirgin bir değişim yaşanmamaktadır. $n_r = 6$ için ortalama toplam süre ise 5.5906 ms olarak görülmektedir. Bu sonuçlar, kapsül sayısındaki bu denli artışların gerçek zamanlı uygulamaların gerçekleştirilmesinde engel teşkil etmeyeceği şeklinde yorumlanmaktadır.



Şekil 6.35: Kapsül sayısına göre ortalama hesaplama süreleri

7. SONUÇ VE ÖNERİLER

Bu tez çalışmasında emniyetli insan-robot etkileşimi için minimum uzaklık hesaplanmasına dayanan bir sistem tasarımı önerilmekte ve çarpışmasız robot hareketi için optimizasyona dayanan bir yerel yol planlama yöntemi geliştirilmektedir. Minimum uzaklık hesaplamaları için ortamın modellenmesi, çalışma ortamında yer alan insanın ve robotun kapsüllerle temsil edilmesi sayesinde gerçekleştirilmektedir. Yol planlama esnasında çarpışmayı önlemek için de, uzaklık hesaplama sonuçları kullanılmaktadır.

Emniyetli insan-robot etkileşimi uygulamaları gerçekleştirebilmek için etkileşime konu olan ortamda yer alan unsurların takip edilmesi ve buna göre önlem alınması gerekmektedir. Etkileşime dahil olan unsurların birbirlerine göre olan durumu, alınacak önlemlerde belirleyici olmaktadır. Durum bilgisi ise genellikle konum ve/veya hız şeklinde ele alınmaktadır. Bunun için, bazen gerçek zamanlı algılayıcılar aracılığıyla gerçek zamanlı konum/hız bilgisi kullanılmakta, bazen de yine algılayıcılar aracılığıyla elde edilen bilgilere göre tahmin gerçekleştirilmektedir. Endüstriyel ortamdaki robot manipülatörlerin insanlarla etkileşiminde emniyeti sağlamak için ise çoğunlukla robotun durumu ve insanın robota göre olan konumu göz önünde bulundurulmaktadır. Bunu sağlayabilmenin yollarından biri de, bu iki unsur arasındaki uzaklığın tam anlamıyla hesaplanması ve böylece göreceli durumun bilinmesi şeklinde olmaktadır. Bu anlamda, minimum uzaklık bilgisi insan-robot işbirliği uygulamaları için elverişli ve etkin bir rol oynamaktadır.

Minimum uzaklık hesaplamasının gerçekleştirildiği sisteme göre insan ve robot, geometrik ilkellerden biri olan kapsüllerle temsil edilmektedir. Kapsüller sayesinde, çalışma ortamında yer alan bu unsurlar modellenmektedir. Modelleme için, robota ait veriler doğrudan robot denetleyicisi aracılığıyla elde edilmekteyken, insan için ise iskelet takibi yapılmaktadır. İskelet takibi için, giyilebilir aygıtlar, çalışma ortamına döşenen özel algılayıcı zeminler literatürdeki mevcut çözümler arasında yer almaktadır. Ancak bu çözümler genel olarak yüksek maliyete sahip olmakla birlikte, bazen de hassasiyet konusunda sıkıntı yaratmaktadır. Önerilen sistem, insanların özel algılayıcılar kuşanmasına gerek kalmadan insan hareketini takip etmek için bir RGB-D algılayıcı ve iskelet takip yazılımı kullanmaktadır. Algılayıcı, el-göz kalibrasyon prosedürü ile kalibre edilmekte ve bu sayede iskelet verileri robot koordinat sistemine göre ifade edilmektedir. Gerekli kurulumun yapılmasından sonra, dışbükey geometriye sahip kapsüller arasındaki minimum uzaklıkları hesaplamak için GJK uzaklık algoritması kullanılmaktadır. GJK uzaklık algoritması, dışbükey geometriler arasındaki uzaklık hesaplama ve çarpışma denetleme

uygulamaları için sıklıkla tercih edilen hızlı ve basit bir yöntemdir. Bu tez çalışmasında da GJK algoritmasının kullanılmasının sebebi, kapsüllerin dışbükey yapıya sahip olması ve gerçek zamanlı olarak emniyeti sağlamak için hızlı bir çözüme ihtiyaç duyulmasıdır.

Önerilen sistemin performansı, bu çalışmada ilk olarak bir insanın ve bir robot manipülatörünün yer aldığı çalışma ortamında test edilmektedir. Robota ait veriler, robot denetleyicisi yerine benzetim ortamındaki robot modelinden sağlanmaktadır. Bunun için, insanın endüstriyel robotun çalışma ortamına dahil olduğu birkaç deney gerçekleştirilmekte ve deneyler aracılığıyla sistemin nicel değerlendirmesi yapılmaktadır. Elde edilen sonuçlar, özellikle hesaplama süreleri açısından önerilen yöntemin gerçek zamanlı ve emniyetli etkileşim uygulamalarında kullanılabileceğini göstermektedir. Özellikle birden fazla insan vücut kısmının düşük hesaplama maliyetiyle ayrı ayrı takip edilebilmesi, önerilen yöntemin literatürdeki benzer yaklaşımlara göre daha fazla doğruluk sağlaması şeklinde yorumlanmaktadır.

Minimum uzaklık hesaplanmasına dayanan bu sistem, aynı zamanda gerçek zamanlı bir uygulama ile doğrulanmaktadır. Bunun için, SMS türünden iş birlikçi etkileşimin görüldüğü bir dizi senaryo oluşturulmaktadır. Her bir senaryoda, insana ait davranışlar ve robot hızları farklılık göstermektedir. Uluslararası standartlarda da tarif edildiği üzere, SMS türünden iş birlikçi senaryoda insan robota izin verilenden fazla yaklaştığında, robot hareketi duraklamaktadır. İnsan ile robot arasındaki uzaklık, emniyet zafiyetine sebep vermeyecek duruma geldiğinde ise robot operasyonu devam etmektedir. Buna göre gerçekleştirilen senaryoların her birinde, önerilen minimum uzaklık hesaplama sistemi kullanılmaktadır. Sonuçların gösterdiği üzere, insan ile robot arasındaki uzaklık izin verilen değerin altına düştüğünde robot hareketi kısa süre içerisinde duraklatılmaktadır. İnsan, robot çalışma alanını terk ettiğinde ise robot daha önceden tanımlanmış hareketine devam etmektedir. Bu senaryoların sonuçları değerlendirildiğinde, önerilen sistemin gerçek zamanlı emniyet uygulamalarında kullanılabileceği öngörülmektedir.

Hem benzetim ortamındaki robot modelinin, hem de gerçek bir robot manipülatörün kullanıldığı senaryoların her biri, algılayıcının görüş açısının uygun bir şekilde ayarlandığı uygulamalar içermektedir. Bu sayede, insanın çalışma ortamında algılanması açısından herhangi bir sıkıntıyla karşılaşılmamaktadır. Ancak karmaşık ve yapılandırılmamış çalışma ortamlarında, her zaman böylesine elverişli sonuçlar elde etmek mümkün olmayabilir. Özellikle robot hareketi sebebiyle algılayıcı görüşü engellenebilir. Böyle bir durumda da, çalışma ortamını izlemek için birden fazla algılayıcıya ihtiyaç duyulabilir. Bunun için, gelecek çalışmaların birden fazla algılayıcı içeren sistem ve çoklu-algılayıcı füzyonu üzerine olması öngörülmektedir. Benzer şekilde endüstride kullanımına başvurulan çözümler, ağırlıklı olarak güvenli olarak sınıflandırılmış algılayıcılar (ing. Safety-rated

sensors) içermektedir. Bu algılayıcıların, her ne kadar maliyeti arttırsa da, hassas ve güvenilir sonuçlar sağlaması açısından bu tarz sistemlerin geliştirilmesinde kullanılabileceği öngörülmektedir.

Emniyetli insan-robot etkileşimini sağlamak için robot üzerinden gerçekleştirilecek önlem ise robot yol planlamasını içermektedir. Buna göre, bu tez çalışmasında optimizasyona dayanan bir yerel yol planlama yöntemi önerilmektedir. Normal koşullarda küresel bir yol planını takip eden robot, bu yönteme göre emniyet zafiyeti oluştuğunda yerel planlayıcı tarafından iteratif olarak oluşturulan yol planını takip etmektedir. Yerel planlayıcı tarafından oluşturulan yol noktaları ise her bir iterasyonda optimizasyon probleminin çözümüne göre belirlenmektedir. Her bir iterasyonda, robot uç noktasının takip edeceği yol noktasının dışında, aynı zamanda robotun sahip olacağı konfigürasyon da belirlenmektedir. Bu sayede robot, takip ettiği yol boyunca kendisini insandan uzakta tutacak konfigürasyonlara sahip olmaktadır.

Önerilen yöntem, tez çalışması kapsamında benzetim ortamında gerçekleştirilen deneylerle test edilmektedir. Bunun için, statik veya dinamik engeller içeren planlama ortamları oluşturulmaktadır. Bu deneyler sayesinde, hem önerilen yöntemin uygulanabilirliği test edilmekte, hem de optimizasyon parametrelerinin sonuca olan etkileri gösterilmektedir.

Önerilen yerel yol planlama yöntemine göre, robot uç noktasının takip edeceği olası yol noktaları bir küre veya yarım küre yüzeyinde tekdüze dağılımlı örnekleme ile belirlenmektedir. Bu örnekleme, robotun görev uzayında tanımlı hedefe düzgün bir yol izleyerek ulaşması için imkan sağlanmaktadır. Literatürde uç noktasının takip edeceği yol noktalarını belirlemek için buna benzer yaklaşımlar olsa da, tekdüze dağılımlı örneklemenin kullanıldığı başka bir yaklaşım bulunmamaktadır. Bu yaklaşımın sonuçlarını değerlendirmek için benzetim ortamında gerçekleştirilen deneyler kullanılmaktadır. Gerçekleştirilen deneylerin her birinde robot, optimizasyon kriterlerine de bağlı olmak kaydıyla, görev uzayında tanımlanmış hedefe ulaşabilmektedir.

Robotun emniyetli bir şekilde etkileşime girebilmesi için önerilen yöntemde, üç farklı mesafe parametresi tanımlanmaktadır. Bu parametreler, literatürdeki birçok yaklaşıma benzer şekilde, emniyetli bölgenin, kritik bölgenin ve tehlikeli bölgenin sınırlarını belirlemektedir. Bu parametreler sayesinde de, yerel planlayıcıya geçişe ve örneklemenin ne şekilde yapılacağına karar verilmektedir. Öyle ki, kritik bölgede yarım küre üzerinde örnekleme yapılmaktayken, tehlikeli bölgede küre yüzeyinde örnekleme yapılmaktadır. Böylece kritik bölgede daha fazla hedefe ulaşma eğilimi sağlanırken, tehlikeli bölgede hedefe uzak örneklerin de devreye girmesiyle bu eğilim ortadan

kaldırılmaktadır. Sadece yarım küre yüzeyinde örnekleme yapılarak gerçekleştirilen deneyin sonuçları, bu farkın yarattığı etkiyi açık bir şekilde göstermektedir.

Önerilen yönteme göre, yol planı boyunca takip edilecek örnek yol noktalarını belirleyebilmek için optimizasyon problemi çözülmektedir. Optimizasyon kriterleri ise hedefe ulaşma, engellerden sakınma ve robot uç noktası için düzgün bir yol takip etme amaçlarını içermektedir. Her amacın ağırlığı da, maliyet fonksiyonundaki ağırlık parametreleri sayesinde belirlenmektedir. Aynı planlama koşulları altında ancak farklı parametre değerleri için gerçekleştirilen deneysel çalışma, bu parametrelerin oluşturulan yol planına etkisini ortaya koymaktadır. Deneysel sonuçlarda da görüldüğü üzere parametrelerdeki değişim, amacın da beklendiği gibi değişmesine sebep olmaktadır. Oluşturulan iki farklı yol planında, parametrelerdeki değişimin etkisi açıkça görülmektedir. Benzer şekilde, parametrelerin etkisini görmek için gerçekleştirilen analiz de bu sonucu desteklemektedir. Buna göre farklı türden amaçlar için, örneğin eklem pozisyonlarının değişimi için, optimizasyon kriterlerine yeni hedeflerin eklenmesi söz konusu olabilir. Bu sebeple gelecek çalışmalarda, bu türden amaçların da optimizasyon problemine dahil edilebileceği öngörülmektedir.

Sonuç olarak önerilen yol planlama yönteminin test edildiği bütün benzetim senaryolarında, optimizasyon parametrelerine bağlı olarak, robot hedefine ulaşmakta ve bu sırada da tehlikeli bölgenin dışında kalmaktadır. Benzetim sırasında robot eklemlerinin sahip olduğu hız, ivme ve sarsım değerleri de üretilen yol noktalarının uygulanabilir olduğunu göstermektedir. Özellikle eklem hızlarının robotun sahip olduğu limitlerin altında olması ve düşük ivme/sarsım değerleri bu görüşü desteklemektedir. Bundan sonra gerçekleştirilecek çalışmalarda, bu sonuçları gerçekçi koşullar altında görebilmek için yöntemin gerçek zamanlı olarak uygulanması da hedeflenmektedir.

KAYNAKLAR DİZİNİ

ABB (2020), *IRB 120 | ABB Robotics - Industrial Robots | ABB Robotics*. URL: https://new.abb.com/products/robotics/industrial-robots/irb-120 (Retrieved 10/10/2021).

Aaltonen, I., Salmi, T. ve Marstio, I. (2018), Refining levels of collaboration to support the design and evaluation of human-robot interaction in the manufacturing industry, *Procedia CIRP* 72, 93–98.

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K. ve Khatib, O. (2018), Progress and prospects of the human–robot collaboration, *Autonomous Robots* 42.5, 957–975.

Aslan, M. F. (2018), İnsan makine etkileşimine yönelik akıllı algılama ve takip etme uygulamaları, Doktora tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, 108 s.

Balan, L. ve Bone, G. M. (2006), Real-time 3D collision avoidance method for safe human and robot coexistence, *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 276–282.

Bdiwi, M., Pfeifer, M. ve Sterzing, A. (2017), A new strategy for ensuring human safety during various levels of interaction with industrial robots, *CIRP Annals* 66.1, 453–456. ISSN: 0007-8506. DOI: https://doi.org/10.1016/j.cirp.2017.04.009.

Behrens, R, Saenz, J, Vogel, C ve Elkmann, N (2015), Upcoming technologies and fundamentals for safeguarding all forms of human-robot collaboration, *8th International Conference Safety of Industrial Automated Systems (SIAS 2015), Königswinter, Germany*, 18–20.

Bergen, G. V. den (1999), A Fast and Robust GJK Implementation for Collision Detection of Convex Objects, *Journal of Graphics Tools* 4.2, 7–25. DOI: 10.1080/10867651.1999.10487502.

Bernabeu, E. J. ve Tornero, J. (2002), Hough transform for distance computation and collision avoidance, *IEEE Transactions on Robotics and Automation* 18.3, 393–398.

KAYNAKLAR DİZİNİ (devam)

Bingol, M. C. ve Aydogmus, O. (2019), İnsan-Robot Etkileşiminde İnsan Güvenliği için Çok Kanallı İletişim Kullanarak Evrişimli Sinir Ağı Tabanlı Bir Yazılımının Geliştirilmesi ve Uygulaması, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi* 31, 489 –495.

Boyacıoğlu, B. (2016), Hızlıca keşfeden rastgele ağaç yöntemi ile insansı robot kolu yörünge planlaması, Yüksek lisans tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, 89 s.

Brito, T., Lima, J., Costa, P. ve Piardi, L. (2017), Dynamic collision avoidance system for a manipulator based on RGB-D data, *Iberian Robotics conference*. Springer, 643–654.

Cefalo, M., Magrini, E. ve Oriolo, G. (2018), Sensor-based task-constrained motion planning using model predictive control, *IFAC-PapersOnLine* 51.22, 220–225.

Cherubini, A., Passama, R., Crosnier, A., Lasnier, A. ve Fraisse, P. (2016), Collaborative manufacturing with physical human–robot interaction, *Robotics and Computer-Integrated Manufacturing* 40, 1–13. ISSN: 0736-5845. DOI: `https://doi.org/10.1016/j.rcim.2015.12.007`.

Cherubini, A., Passama, R., Navarro, B., Sorour, M., Khelloufi, A., Mazhar, O., Tarbouriech, S., Zhu, J., Tempier, O., Crosnier, A., vd. (2019), A collaborative robot for the factory of the future: Bazar, *The International Journal of Advanced Manufacturing Technology* 105.9, 3643–3659.

Cubemos (2020), *Skeleton Tracking SDK*. URL: `https://www.cubemos.com/skeleton-tracking-sdk` (Retrieved 10/10/2021).

Daniilidis, K. (1999), Hand-Eye Calibration Using Dual Quaternions, *The International Journal of Robotics Research* 18.3, 286–298. DOI: `10.1177/02783649922066213`.

Dağlı, M. (2019), İnsan-Robot İşbirliği: Kolaboratif Robotlar, *Mühendis ve Makine*, 28–33.

Dilibal, S. ve Şahin, H. (2018), İşbirlikçi Endüstriyel Robotlar ve Dijital Endüstri, *International Journal of 3D Printing Technologies and Digital Industry* 2, 86–96.

KAYNAKLAR DİZİNİ (devam)

Duman, F. ve Düven, E. (2019), Bir Endüstriyel Robotun Kafessiz Çalışmasını Sağlayacak Görüntü Tabanlı Güvenlik Sisteminin Geliştirilmesi, *Uludağ University Journal of The Faculty of Engineering*, 355 –372.

FCL (2020a), *Flexible Collision Library*. URL: `https://github.com/flexible-collision-library/fcl` (Retrieved 04/12/2021).

— (2020b), *fcl::detail::GJKSolver_indep< S_ > Struct Template Reference*. URL: `https://flexible-collision-library.github.io/dd/dd2/structfcl_1_1detail_1_1GJKSolver__indep.html` (Retrieved 05/03/2020).

Fischer, M. ve Henrich, D. (2009), Surveillance of robots using multiple colour or depth cameras with distributed processing, *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. IEEE, 1–8.

Flacco, F., Kröger, T., De Luca, A. ve Khatib, O. (2012), A depth space approach to human-robot collision avoidance, *2012 IEEE International Conference on Robotics and Automation*. IEEE, 338–345.

Gao, M., Chen, D., Yang, Y. ve He, Z. (2015), A fixed-distance planning algorithm for 6-DOF manipulators, *Industrial Robot: An International Journal*.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J. ve Marín-Jiménez, M. J. (2014), Automatic generation and detection of highly reliable fiducial markers under occlusion, *Pattern Recognition* 47.6, 2280–2292.

Garrido-Jurado, S., Munoz-Salinas, R., Madrid-Cuevas, F. J. ve Medina-Carnicer, R. (2016), Generation of fiducial marker dictionaries using mixed integer linear programming, *Pattern Recognition* 51, 481–491.

Gilbert, E. ve Foo, C.-P. (1990), Computing the distance between general convex objects in three-dimensional space, *IEEE Transactions on Robotics and Automation* 6.1, 53–61. DOI: `10.1109/70.88117`.

KAYNAKLAR DİZİNİ (devam)

Gilbert, E., Johnson, D. ve Keerthi, S. (1988), A fast procedure for computing the distance between complex objects in three-dimensional space, *IEEE Journal on Robotics and Automation* 4.2, 193–203. DOI: 10.1109/56.2083.

Han, D., Nie, H., Chen, J. ve Chen, M. (2018), Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection, *Robotics and Computer-Integrated Manufacturing* 49, 98–104.

Hentout, A., Aouache, M., Maoudj, A. ve Akli, I. (2019), Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017, *Advanced Robotics* 33.15-16, 764–799. DOI: 10.1080/01691864.2019.1636714.

IFR (2020a), *Executive Summary World Robotics 2020 Industrial Robots*. International Federation of Robotics, 4 p.

— (2020b), *Top trends robotics 2020*. URL: https://ifr.org/news/top-trends-robotics-2020/ (Retrieved 10/10/2021).

ISO (2011), *ISO 10218-1:2011: Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots*. International Organization for Standardization.

— (2012), *ISO 8373:2012: Robots and robotic devices - Vocabulary*. International Organization for Standardization.

— (2016), *ISO/TS 15066:2016: Robots and robotic devices — Collaborative robots*. International Organization for Standardization.

KUKA (2020), *LBR iiwa | KUKA AG*. URL: https://www.kuka.com/en-gb/products/robotics-systems/industrial-robots/lbr-iiwa (Retrieved 10/10/2021).

Karacığan, A. O. (2021), İş birlikçi endüstriyel robotlarda iş güvenliği ve risk analizi yaklaşımları, Yüksek lisans tezi, Üsküdar Üniversitesi Sağlık Bilimleri Enstitüsü, 197 s.

KAYNAKLAR DİZİNİ (devam)

Kawasaki (2020), *RS005L Robot | R Series | Small-Medium Payload | Kawasaki Robotics*. URL: https://robotics.kawasaki.com/en1/products/robots/small-medium-payloads/RS005L/ (Retrieved 10/10/2021).

Kelemen, M., Virgala, I., Lipták, T., Miková, L., Filakovskỳ, F. ve Bulej, V. (2018), A novel approach for a inverse kinematics solution of a redundant manipulator, *Applied Sciences* 8.11, 2229.

Khan, A. H., Li, S. ve Luo, X. (2019), Obstacle avoidance and tracking control of redundant robotic manipulator: an RNN-based metaheuristic approach, *IEEE transactions on industrial informatics* 16.7, 4670–4680.

Khatib, O. (1986), Real-time obstacle avoidance for manipulators and mobile robots, *Autonomous robot vehicles*. Springer, 396–404.

Khungrun, P. (2015), Probability Density Under Transformation, *Cornell University*. URL: https://www.cs.cornell.edu/courses/cs6630/2015fa/notes/pdf-transform.pdf (Retrieved 10/10/2021).

Kivelä, T., Mattila, J., Puura, J. ve Launis, S. (2018), Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance, *Advances in Service and Industrial Robotics*. Ed. C. Ferraresi ve G. Quaglia. Springer International Publishing, 208–216. ISBN: 978-3-319-61276-8.

Koenig, N. P. ve Howard, A. (2004), Design and use paradigms for Gazebo, an open-source multi-robot simulator, *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 3, 2149–2154.

Kong, M. ve Yu, G. (2014), Collision detection algorithm for dual-robot system, *2014 IEEE International Conference on Mechatronics and Automation*. IEEE, 2083–2088.

LIPS (2020), *3D Body Skeleton | LIPS Corporation*. URL: https://www.lips-hci.com/3d-body-skeleton (Retrieved 10/10/2021).

LaValle, S. M. (2006), *Planning algorithms*. Cambridge university press.

KAYNAKLAR DİZİNİ (devam)

Lasota, P. A., Rossano, G. F. ve Shah, J. A. (2014), Toward safe close-proximity human-robot interaction with standard industrial robots, *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 339–344.

Lee, K.-K. ve Buss, M. (2007), Obstacle avoidance for redundant robots using Jacobian transpose method, *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3509–3514.

Lillo, P. D., Chiaverini, S. ve Antonelli, G. (2019), Handling robot constraints within a Set-Based Multi-Task Priority Inverse Kinematics Framework, *2019 International Conference on Robotics and Automation (ICRA)*, 7477–7483. DOI: 10.1109/ICRA.2019.8793625.

Lin, H.-C., Fan, Y., Tang, T. ve Tomizuka, M. (2016), Human guidance programming on a 6-dof robot with collision avoidance, *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2676–2681.

Liu, J., Tong, Y., Ju, Z. ve Liu, Y. (2020), Novel Method of Obstacle Avoidance Planning for Redundant Sliding Manipulators, *IEEE Access* 8, 78608–78621. DOI: 10.1109/ACCESS.2020.2990555.

Long, P., Chevallereau, C., Chablat, D. ve Girin, A. (2018), An industrial security system for human-robot coexistence, *Industrial Robot: An International Journal*.

Lozano-Perez, T. (1990), Spatial planning: A configuration space approach, *Autonomous robot vehicles*. Springer, 259–271.

MATLAB (2020), *Robotics System Toolbox™ Documentation*. URL: https://www.mathworks.com/help/robotics/index.html (Retrieved 10/10/2021).

Magrini, E., Ferraguti, F., Ronga, A. J., Pini, F., De Luca, A. ve Leali, F. (2020), Human-robot coexistence and interaction in open industrial cells, *Robotics and Computer-Integrated Manufacturing* 61, 101846. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2019.101846.

KAYNAKLAR DİZİNİ (devam)

Mao, H. ve Xiao, J. (2019), Real-time conflict resolution of task-constrained manipulator motion in unforeseen dynamic environments, *IEEE Transactions on Robotics* 35.5, 1276–1283.

Marvel, J. A. ve Norcross, R. (2017), Implementing speed and separation monitoring in collaborative robot workcells, *Robotics and computer-integrated manufacturing* 44, 144–155.

Matthias, B., Byner, C. ve Ding, H. (2019), Dynamic speed and separation monitoring for collaborative robot applications – Concepts and performance, *Robotics and Computer-Integrated Manufacturing* 58, 239–252. DOI: 10.1016/j.rcim.2018.11.002.

Maurtua, I., Fernandez, I., Tellaeche, A., Kildal, J., Susperregi, L., Ibarguren, A. ve Sierra, B. (2017a), Natural multimodal communication for human-robot collaboration, *International Journal of Advanced Robotic Systems* 14, 1–12. DOI: 10.1177/1729881417716043.

Maurtua, I., Ibarguren, A., Kildal, J., Susperregi, L. ve Sierra, B. (2017b), Human–robot collaboration in industrial applications: Safety, interaction and trust, *International Journal of Advanced Robotic Systems* 14.4. DOI: 10.1177/1729881417716010.

Meziane, R., Li, P., Otis, M. J.-D., Ezzaidi, H. ve Cardou, P. (2014), Safer hybrid workspace using human-robot interaction while sharing production activities, *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*. IEEE, 37–42.

Michalos, G., Makris, S., Spiliotopoulos, J., Misios, I., Tsarouchi, P. ve Chryssolouris, G. (2014), ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future, *Procedia CIRP* 23, 71–76.

Michalos, G., Makris, S., Tsarouchi, P., Guasch, T., Kontovrakis, D. ve Chryssolouris, G. (2015), Design considerations for safe human-robot collaborative workplaces, *Procedia CIrP* 37, 248–253.

KAYNAKLAR DİZİNİ (devam)

Mohammed, A., Schmidt, B. ve Wang, L. (2017), Active collision avoidance for human–robot collaboration driven by vision sensors, *International Journal of Computer Integrated Manufacturing* 30.9, 970–980.

MoveIt (2020), *Hand-Eye Calibration - moveit_tutorials Noetic documentation*. URL: `https://ros-planning.github.io/moveit_tutorials/doc/hand_eye_calibration/hand_eye_calibration_tutorial.html` (Retrieved 10/10/2021).

Nikolakis, N., Maratos, V. ve Makris, S. (2019), A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace, *Robotics and Computer-Integrated Manufacturing* 56, 233–243.

Nuitrack (2020), *Nuitrack Full Body Skeletal Tracking Software*. URL: `https://nuitrack.com/` (Retrieved 10/10/2021).

OpenCV (2020), *OpenCV: Detection of ArUco Markers - OpenCV 4.2.0-dev documentation*. URL: `https://docs.opencv.org/trunk/d5/dae/tutorial_aruco_detection.html` (Retrieved 10/10/2021).

OpenNI2 (2020), *OpenNI2*. URL: `https://github.com/OpenNI/OpenNI2` (Retrieved 10/10/2021).

Papanastasiou, S., Kousi, N., Karagiannis, P., Gkournelos, C., Papavasileiou, A., Dimoulas, K., Baris, K., Koukas, S., Michalos, G. ve Makris, S. (2019), Towards seamless human robot collaboration: integrating multimodal interaction, *The International Journal of Advanced Manufacturing Technology* 105.9, 3881–3897.

Park, F. C. ve Martin, B. J. (1994), Robot sensor calibration: solving {AX} = {XB} on the {Euclidean} group, *IEEE Transactions on Robotics and Automation* 10.5, 717–721.

Pellegrinelli, S., Orlandini, A., Pedrocchi, N., Umbrico, A. ve Tolio, T. (2017), Motion planning and scheduling for human and industrial-robot collaboration, *CIRP Annals* 66.1, 1–4.

KAYNAKLAR DİZİNİ (devam)

Qiu, Q. ve Cao, Q. (2018), Task constrained motion planning for 7-degree of freedom manipulators with parameterized submanifolds, *Industrial Robot: An International Journal*.

ROS (2020), *ROS.org | Powering the world's robots*. URL: https://www.ros.org/ (Retrieved 10/10/2021).

Robla-Gómez, S., Becerra, V. M., Llata, J. R., González-Sarabia, E., Torre-Ferrero, C. ve Pérez-Oria, J. (2017), Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments, *IEEE Access* 5, 26754–26773. DOI: 10.1109/ACCESS.2017.2773127.

Safeea, M., Béarée, R. ve Neto, P. (2020), Collision Avoidance of Redundant Robotic Manipulators Using Newton's Method, *Journal of Intelligent and Robotic Systems*. DOI: 10.1007/s10846-020-01159-3.

Safeea, M., Mendes, N. ve Neto, P. (2017), Minimum distance calculation for safe human robot interaction, *Procedia Manufacturing* 11, 99–106.

Safeea, M. ve Neto, P. (2019), Minimum distance calculation using laser scanner and IMUs for safe human-robot interaction, *Robotics and Computer-Integrated Manufacturing* 58, 33–42.

Safeea, M., Neto, P. ve Bearee, R. (2019), On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case, *Robotics and Autonomous Systems* 119, 278–288.

Salmi, T., Marstio, I., Malm, T. ve Montonen, J. (2016), Advanced safety solutions for human-robot-cooperation, *Proceedings of ISR 2016: 47st International Symposium on Robotics*. VDE, 1–6.

Schmidt, B. ve Wang, L. (2014), Depth camera based collision avoidance via active robot control, *Journal of manufacturing systems* 33.4, 711–718.

Secil, S. ve Ozkan, M. (2022), Minimum distance calculation using skeletal tracking for safe human-robot interaction, *Robotics and Computer-Integrated Manufacturing* 73, 102253.

KAYNAKLAR DİZİNİ (devam)

Shiu, Y. C. ve Ahmad, S. (1989), Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX =XB, *IEEE Trans. Robotics Autom.* 5, 16–29.

Siciliano, B. (1990), Kinematic control of redundant robot manipulators: A tutorial, *Journal of intelligent and robotic systems* 3.3, 201–212.

Stentz, A. (1997), Optimal and efficient path planning for partially known environments, *Intelligent unmanned ground vehicles*. Springer, 203–220.

Teulieres, M., Tilley, J., Bolz, L., Ludwig-Dehm, P. M. ve Wägner, S. (2019), *Industrial robotics - Insights into the sector's future growth dynamics*. McKinsey & Company, 33 p.

Tsai, C.-S. (2014), *Online trajectory generation for robot manipulators in dynamic environment—An optimization-based approach*. University of California, Berkeley.

Tsai, R. ve Lenz, R. (1989), A new technique for fully autonomous and efficient 3D robotics hand/eye calibration, *IEEE Transactions on Robotics and Automation* 5.3, 345–358. DOI: 10.1109/70.34770.

Tsarouchi, P., Makris, S. ve Chryssolouris, G. (2016), Human–robot interaction review and challenges on task planning and programming, *International Journal of Computer Integrated Manufacturing* 29.8, 916–931. DOI: 10.1080/0951192X.2015.1130251.

UniversalRobots (2020), *UR5 collaborative robot arm | Flexible and lightweight cobot*. URL: https://www.universal-robots.com/products/ur5-robot/ (Retrieved 10/10/2021).

Unver, B. (2004), Artık robot kollarının ters kinematik problemlerinin çözümü ve gerçeklenmesi, Yüksek lisans tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, 123 s.

Villani, V., Pini, F., Leali, F. ve Secchi, C. (2018), Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications, *Mechatronics* 55, 248–266. DOI: https://doi.org/10.1016/j.mechatronics.2018.02.009.

KAYNAKLAR DİZİNİ (devam)

Wang, W., Zhu, M., Wang, X., He, S., He, J. ve Xu, Z. (2018), An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators, *International Journal of Advanced Robotic Systems* 15.5.

Wang, X. V., Kemény, Z., Váncza, J. ve Wang, L. (2017a), Human–robot collaborative assembly in cyber-physical production: Classification framework and implementation, *CIRP annals* 66.1, 5–8.

Wang, Y., Sheng, Y., Wang, J. ve Zhang, W. (2017b), Optimal collision-free robot trajectory generation based on time series prediction of human motion, *IEEE Robotics and Automation Letters* 3.1, 226–233.

Wilhelm, B., Manfred, B., Braun, M., Rally, P. ve Scholtz, O. (2016), *Lightweight robots in manual assembly – best to start simply! Examining companies' initial experiences with lightweight robots*. Fraunhofer IAO, 63 p.

Xiang, L., Echtler, F., Kerl, C., Wiedemeyer, T., Lars, Hanyazou, Gordon, R., Facioni, F., laborer2008, Wareham, R., Goldhoorn, M., alberth, gaborpapp, Fuchs, S., jmtatsch, Blake, J., Federico, Jungkurth, H., Mingze, Y., vinouz, Coleman, D., Burns, B., Rawat, R., Mokhov, S., Reynolds, P., Viau, P. E., Fraissinet-Tachet, M., Ludique, Billingham, J. ve Alistair (2016), *libfreenect2: Release 0.2*. URL: `https://doi.org/10.5281/zenodo.50641` (Retrieved 10/10/2021).

Zabalza, J., Fei, Z., Wong, C., Yan, Y., Mineo, C., Yang, E., Rodden, T., Mehnen, J., Pham, Q.-C. ve Ren, J. (2019), Smart sensing and adaptive reasoning for enabling industrial robots with interactive human-robot capabilities in dynamic environments—A case study, *Sensors* 19.6, 1354.

Zavlangas, P. G. ve Tzafestas, S. G. (2000), Industrial robot navigation and obstacle avoidance employing fuzzy logic, *Journal of Intelligent and robotic Systems* 27.1, 85–97.

Zhang, J., Li, P., Zhu, T., Zhang, W.-A. ve Liu, S. (2020), Human Motion Capture Based on Kinect and IMUs and Its Application to Human-Robot Collaboration, *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 392–397.