

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**HIZLICA KEŞFEDEN RASTGELE AĞAÇ YÖNTEMİ İLE
İNSANSI ROBOT KOLU YÖRÜNGE PLANLAMASI**

YÜKSEK LİSANS TEZİ

Burak BOYACIOĞLU

Makina Mühendisliği Anabilim Dalı

Sistem Dinamiği ve Kontrol Programı

HAZİRAN 2016

**HIZLICA KEŞFEDEN RASTGELE AĞAÇ YÖNTEMİ İLE
İNSANSI ROBOT KOLU YÖRÜNGE PLANLAMASI**

YÜKSEK LİSANS TEZİ

**Burak BOYACIOĞLU
(503131602)**

Makina Mühendisliği Anabilim Dalı

Sistem Dinamiği ve Kontrol Programı

Tez Danışmanı: Prof. Dr. Şeniz ERTUĞRUL

HAZİRAN 2016

İTÜ, Fen Bilimleri Enstitüsü'nün 503131602 numaralı Burak BOYACIOĞLU, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “HIZLICA KEŞFEDEN RASTGELE AĞAÇ YÖNTEMİ İLE İNSANSI ROBOT KOLU YÖRÜNGE PLANLAMASI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Şeniz ERTUĞRUL**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Erdinç ALTUĞ**
İstanbul Teknik Üniversitesi

Yrd. Doç. Dr. Figen ÖZEN
Haliç Üniversitesi

Teslim Tarihi : 2 Mayıs 2016
Savunma Tarihi : 3 Haziran 2016





Aileme,



ÖNSÖZ

“Elinizde tuttuğunuz ya da elektronik ortamda incelediğiniz bu çalışma, tarafımdan kaleme alınmış olsa da çalışmanın, pek kıymetli kişilerin iştirakiyle vücuda geldiğini belirtmeliyim.” cümlesi ile başladığım lisans bitirme çalışmamın tesliminin üzerinden geçen 3 yılda yanımda olan insanlara yenileri eklendi.

Öncelikle Mekatronik Yandal Programı vesilesiyle tanıdığım ve daha ilk dinlediğim dersinde ders anlatımından çok etkilendiğim Prof. Dr. Şeniz ERTUĞRUL ile hem araştırma görevlisi hem de yüksek lisans öğrencisi olarak çalışma fırsatım oldu. Asistanlığa kabulümden yüksek lisans tez savunmama yanımda olan, kararlarımı destekleyerek beni yönlendiren hocama teşekkürlerimi sunmak isterim.

Aileme; çeyrek asır boyunca sevgilerini ve desteklerini hiç esirgemeyen anne, baba ve ablama şükranlarımı ifade etmek için bu tez çalışmasını onlara ithaf etmek, naçizane bir adım. Umarım daha nicelerini atabilirim.

Senelerdir en yakınımda olup akademik ve akademik olmayan başarılı çalışmalarımda en büyük paylardan birinin tek başına sahibi Özde ÖZDAL’a, dostlarım Yiğit ÇAY ve Serkan GÖKTAŞ’a, tez sürecimi anbean takip etmek durumunda kalıp ilgisini eksiltmeyen oda arkadaşım Hülya ERASLAN’a ve son ama özel olarak bana bolca zaman ayırıp RRT algoritması gibi gerçek zamanlı bağlanabilirlik yolu veren, yer yer çalışmamı pürüzsüzleştirmeme dahi yardımcı olan değerli arkadaşım Cihat Bora YİĞİT’e samimi teşekkürlerimi sunarım.

Bu tez çalışması, 38826 proje numarası ile İTÜ BAP Birimi tarafından desteklenmiştir.

Haziran 2016

Burak Boyacıoğlu
(Uçak Mühendisi)



İÇİNDEKİLER

Sayfa

İÇİNDEKİLER	ix
KISALTMALAR	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ.....	xv
ŞEKİL LİSTESİ.....	xvii
ÖZET.....	xix
SUMMARY	xxi
1. GİRİŞ.....	1
1.1 Tezin Amacı	1
1.2 Literatür Taraması	2
1.3 Yöntem	4
1.4 Tez Düzeni	4
2. TEMEL KAVRAMLAR	7
2.1 Planlama Uzayları	7
2.2 Geometrik ve Kinematik Modeller	9
2.2.1 İleri ve ters geometrik modeller	9
2.2.2 İleri ve ters kinematik modeller	10
2.2.3 Tekil nokta civarında ters kinematik çözümü	11
2.3 Kısıtlar	13
2.3.1 Yerel kısıtlar	13
2.3.2 Küresel kısıtlar	13
2.3.3 Diğer kısıtlar	14
2.4 Planlama Uzaylarının Karşılaştırması	15
3. ÖRNEKLEME TABANLI PLANLAMA	17
3.1 Olasılıksal Yol Haritası Yöntemi	17
3.2 Rastgele Potansiyel Alanlar Yöntemi	18
3.3 Hızlıca Keşfeden Rastgele Ağaç Yöntemi	19
3.4 Çarpışma Sınaması	22
3.5 Yerel Planlama	25
3.5.1 Eklem yörüngelerinin kübik polinomlar ile ifade edilmesi	26
3.5.2 Uygulanabilir çözüm dönüştürücü	27
4. ITECH İNSANSI ROBOT KOLU	29
4.1 Geometrik Kısıtlar ve Tekillikler	31
4.2 Önerilen Ters Geometrik Çözüm	32
5. ÖNERİLEN YÖRÜNGE PLANLAMA YÖNTEMİ	35
5.1 Nelder-Mead Optimizasyon Yöntemi	35
5.2 Kübik Polinom Eklem Yörüngelerinin Optimizasyonu	37
5.3 HRA Sonuçlarının Süreklileştirilmesi için Bir Yöntem.....	38
5.4 Benzetim Senaryosu ve Sonuçlar	41
5.5 Manipülatör Dinamiği ve Kontrolü	47

6. SONUÇ.....	51
KAYNAKLAR.....	53
EKLER.....	57
ÖZGEÇMİŞ.....	63



KISALTMALAR

D-H	: Denavit-Hartenberg
DK	: Dallarını Kırmak
G-uzayı	: Görev uzayı
HRA	: Hızlıca Keşfeden Rastgele Ağaç
İTÜ	: İstanbul Teknik Üniversitesi
K-uzayı	: Konfigürasyon uzayı
kg	: kilogram
mm	: milimetre
mNm	: milinewton metre
Nm	: Newton metre
OYH	: Olasılıksal Yol Haritası
RPA	: Rastgele Potansiyel Alanlar
s	: saniye
UÇD	: Uygulanabilir Çözüm Dönüştürücüsü



SEMBOLLER

A	: Katsayılar matrisi
$\vec{a}$	: Uç eyleyici normal vektörü
a, d, α, θ	: D-H parametreleri
d, r, α, θ	: Değiştirilmiş D-H parametreleri
C	: Coriolis matrisi
D	: Kütle ya da atalet matrisi
$\hat{D}, \hat{C}, \hat{g}, \hat{F}$	: Atalet, Coriolis, yerçekimi ve sürtünme terimlerinin yaklaşık ifadesi
e	: Yörünge izleme hatası
$\vec{e}$	: Birim vektör
F	: Eksen takımı
F_0	: Temel eksen takımı
F_e	: Uç eyleyiciye ait eksen takımı
g	: Yerçekiminin kaynaklanan tork vektörü
h	: Zaman aralığı
HK, IK, SK	: Eklemlerin hız, ivme ve sarsım kısıtları
i	: Motor akımı
J	: Jakobiyen matrisi
K	: Kinetik enerji
K_h, K_k	: Kazanç katsayıları
$K_{serbest}$	: Engelsiz K-uzayı
k_t	: Motor tork sabiti
L	: Lagranjiyen
N	: Örneklem sayısı
N	: Eklem sayısı
n	: Serbestlik derecesi
o	: Konum vektörü
$Q(t)$	: Geçiş noktaları arasındaki konum fonksiyonu
q	: Uç eyleyici eklem konumu vektörü
P	: Potansiyel enerji
R	: Dönme matrisi
T	: Dönüşüm matrisi
T	: Rastgele ağaç
t	: Zaman
v	: Çizgisel hız
X	: Zaman aralıkları vektörü
z	: Dönme matrisinin 3. Sütunu
Λ	: Sönüm sabiti
δ	: Yönelim toleransı
λ	: Uygulanabilirlik ayar çarpanı
μ	: Uzaklık toleransı
τ	: Tork
$\odot$	: Nokta çarpım



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Günlük aktivitelerde insan kolunun sınırları (Rosen ve diğ., 2005). ...	15
Çizelge 2.2 : İnsansı robot kollarına ait geometrik hareket limitleri.	15
Çizelge 2.3 : Planlama uzaylarının birbirine üstünlükleri.	16
Çizelge 4.1 : D-H ve değiştirilmiş D-H parametreleri.....	30
Çizelge 4.2 : Motor ve dişli mekanizmalarına ait özellikler (Güleç, 2014).	30
Çizelge 4.3 : ITECH İnsansı Robot Kolu'na ait geometrik kısıtlar.....	31
Çizelge 5.1 : Şekil 5.2'de yer alan 12 kareye ait nokta konum ve maliyetleri.	37
Çizelge 5.2 : G-uzayındaki küresel engellerin merkezlerinin koordinatları.	41
Çizelge 5.3 : Robot kol için tanımlanan hız, ivme ve sarsım sınır değerleri.	42
Çizelge 5.4 : HRA tarafından verilen geçiş noktalarındaki eklem konfigürasyonları	43



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Planlama kavramlarının şematik gösterimi.	7
Şekil 2.2 : G-uzayındaki bir noktanın K-uzayında birden fazla karşılığı olması durumunun bir örneği.	8
Şekil 2.3 : D-H konvansiyonuna göre eksenlerin belirlenmesi ve diğer eksen parametreleri (Spong, Hutchinson ve Vidyasagar, 2006).....	9
Şekil 2.4 : Ters kinematiğe çözüm olarak kullanılan sezgisel algoritmalarından Döngüsel Koordinat Yaklaşması.	12
Şekil 2.5 : Çarpışma ve kaçınılmaz çarpışma durumları (Fraichard ve Asama, 2003).	14
Şekil 3.1 : OYH yönteminin aşamaları: a) N adet rastgele nokta oluşturma, b) yol haritası elde etme, c) başlangıç ve hedef noktaların ilavesi ve d) yolu bulma.	18
Şekil 3.2 : OYH yönteminde yol haritası oluşturma algoritması.....	18
Şekil 3.3 : a) Önceden bilinen hedef nokta ve engeller, b) hedef noktaya yakınlık fonksiyonu, c) engellere uzaklık fonksiyonu ve d) iki fonksiyonun toplamı (Taylor, 2016).	19
Şekil 3.4 : Temel HRA algoritması.	20
Şekil 3.5 : Rastgele ağacın genişleme ilkesi.	21
Şekil 3.6 : Ölçev olarak a) Öklid mesafesinin ve b) Manhattan mesafesinin kullanıldığı 2 boyutlu uzayda Voronoi hücreleri (Url-2).	22
Şekil 3.7 : DK algoritması (Boyacıoğlu ve Ertuğrul, 2015).	22
Şekil 3.8 : HRA ve diğer örnekleme tabanlı algoritmaların alt fonksiyonları (Elbanhawi ve Simic, 2014).	23
Şekil 3.9 : a) İki nokta arasında çarpışma olasılığı dağılımı ve b) doğru parçası uzunluğuna göre çarpışma durumu (Elbanhawi ve Simic, 2014).	24
Şekil 3.10 : a) Artımsal çarpışma sınaması ve b) orta noktalarda çarpışma sınaması (Elbanhawi ve Simic, 2014).	24
Şekil 3.11 : Sınırlayıcı hacimler (Weller, 2013).	24
Şekil 3.12 : Stanford tavşanı ve kürelerle sınırlanması (de Sousa Rocha ve Rodrigues, 2008).	25
Şekil 3.13 : Değişken gösterimleri.	26
Şekil 4.1 : ITECH İnsansı Robot Kolu (Güleç, 2014).	29
Şekil 4.2 : a) D-H ve b) değiştirilmiş D-H parametreleri için tanımlanan eksenler. .	30
Şekil 4.3 : ITECH İnsansı Robot Kolu'na ait bazı tekillikler.	32
Şekil 4.4 : Asfour ve Dillmann'ın önerdiği analitik ters geometri yaklaşımı (2003).	33
Şekil 5.1 : Nelder-Mead yöntemine ait algoritma.	35
Şekil 5.2 : $f(x, y)$ fonksiyonunun minimum değerinin Nelder-Mead yöntemi ile bulunuşu.	36
Şekil 5.3 : HRA tarafından verilen yolun optimum zaman için pürüzsüzleştirilmesi. .	38
Şekil 5.4 : Hedef duruştan konumsal sapma.	39

Şekil 5.5 : Hedef duruştan açısal sapma.	40
Şekil 5.6 : Başlangıç durumunda robot kol ile engellerin konumlandırması.....	42
Şekil 5.7 : Süreklileştirilen yörüngeye ait konum, hız, ivme ve sarsım profilleri.	44
Şekil 5.8 : Uç eyleyici için HRA tarafından verilen yol (kesikli çizgi), süreklileştirilmiş yörünge (yeşil çizgi) ve hedef bölge (açık mavi kafes).	45
Şekil 5.9 : Robot kolun hareketine ait 15 ardışık kare.	45
Şekil 5.10 : SimMechanics™ tarafından verilen hesaplanmış tork değerleri.....	46
Şekil 5.11 : Hareketin gerçekleşmesi için gerekli motor akım değerleri.	46
Şekil 5.12 : Ters dinamik modelin çalışma mantığı.	48
Şekil 5.13 : Yaklaşık hesaplanmış tork kontrolcüsüne ait blok diyagramı.....	48
Şekil 5.14 : Her bir eklem için yörünge takip hatası.	49



HIZLICA KEŞFEDEN RASTGELE AĞAÇ YÖNTEMİ İLE İNSANSI ROBOT KOLU YÖRÜNGE PLANLAMASI

ÖZET

Bir robotik sistemin ya da bir otonom aracın bir konumdan bir başka konuma hangi yolu, hangi hız ve ivme profilleri ile izleyeceğinin belirlenmesi, yörünge planlama olarak tanımlanmaktadır. Yörünge planlama, robotiğin en çok çalışılan konularından biri olmakla beraber bu problem için tek, kesin ve optimum bir çözümden söz edilemez. Bu durum, küresel engellerden sakınılması gereken senaryolar için de aynen geçerlidir. Hatta problemin daha da karmaşıklaşması kaçınılmazdır.

Yörünge planlama için önerilen yöntemler temel olarak belirlenimci ve rastlantısal olarak ikiye ayrılabilir. Belirlenimci yöntemler genellikle eksiksiz olmakla birlikte, planlama uzayının boyutları arttıkça hesaplama maliyetinin artmasından muzdarip olacaktır. Öte yandan, rastlantısal yöntemler tüm olası çözümleri değerlendirmez.

Yörünge planlama için bulanık mantık kontrolden yapay sinir ağlarına, karınca kolonisinden genetik algoritmalara çeşitli yöntemler kullanılmış; ancak rastlantısal yöntemler arasında örnekleme tabanlı sezgisel algoritmalar sade olmaları ve hesaplama maliyetlerinin düşük olması sebebiyle kabul görmüştür. Bu yöntemlerden en yaygın olanları Olasılıksal Yol Haritası, Rastgele Potansiyel Alanlar ve Hızlıca Keşfeden Rastgele Ağaç algoritmaları ve onların türevleridir.

Başlıca örnekleme tabanlı planlama algoritmaları için eksiksizlikten söz edilemez. Yani bir çözüm olmadığı takdirde, bu yöntemler ile çözümsüzlük durumu tespit edilemeyecektir. Dolayısıyla örnekleme için ya örnek sayısı ya da örnekleme süresinin sınırlandırılması gereklidir. Öte yandan yeterince örnekleme yapıldığı takdirde, uygulanabilir bir çözüm mevcut ise bunu bulmayı vaat eder yani olasıksal olarak eksiksizdir.

Örnekleme tabanlı algoritmalar; önceden seçilecek bir ölçeğe göre hangi örneklerin birbirine bağlanacağını belirlemek, başlangıç noktasından hedef noktaya hangi hız ve ivme profillerini takip ederek ulaşacağını planlamak, planlanan yol için çarpışma sınaması yapmak gibi alt fonksiyonlar ile birlikte çalışır.

Bu çalışmada, 6 serbestlik dereceli ITECH İnsansı Robot'una ait kolun küresel engellerden kaçan yörünge planlamasını yapmak üzere, öncelikle, Hızlıca Keşfeden Rastgele Ağaç yöntemi geçiş noktalarını belirlemek üzere çağrılmıştır. Planlama uzayı olarak, eklemlerin bulunduğu konfigürasyon uzayı kullanılmış olup bu sayede ters geometri ve ters kinematik hesaplarından kaçınılmıştır. Görev uzayında olan hedef noktanın da konfigürasyon uzayına aktarılması yerine, konfigürasyon uzayında oluşturulan ağacın görev uzayındaki izdüşümünün hedef noktaya yeterince yakın oluncaya dek genişlemesi istenmiştir. Bunun için, konum ve yönelim toleransları tanımlanmıştır.

Hızlıca Keşfeden Rastgele Ağaç yöntemi ile bağlanabilirlik yolu elde edildikten sonra, yerel planlayıcı alt fonksiyonu olarak her bir geçiş noktası arasına kübik yörünge eğrileri uydurulmuştur. Bunun için, manipülatörlere özgü bir çalışmadan yararlanılmıştır. Eklem konumlarının kübik eğrilerle ifade edilmesi; eklem hız profillerinin karesel, eklem ivme profillerinin ise doğrusal olması anlamına gelir. Bir

başka deyişle bu yaklaşımla; konum, hız ve ivme profillerinde süreklilik sağlanmış olur.

Bir sonraki aşamada, hareketin toplam zamanının maliyet fonksiyonu olarak ele alındığı kübik eğrilerin optimizasyonu yapılmış ve bu işlem gerçekleştirilirken sadece sınır koşulları değil; hız, ivme ve sarsım sınırları da dikkate alınmıştır. Optimizasyon algoritması olarak, Nelder-Mead arama yöntemi kullanılmıştır.

Çalışma, yörünge süreklileştirmenin açıkça anlatılması, verilen geçiş noktaları için zaman optimizasyonu yapılması ve sarsımın sınırlandırılması açılarından önceki çalışmalardan farklılık gösterdiği için literatüre katkı niteliğindedir.

Süreklileştirme işlemini takiben, yörünge üzerinde çarpışma sınamaları yapılarak sadece uç eyleyicinin değil, tüm robot kolun engellere çarpmadan bir hareket sergilemesi sağlanmıştır.

Kullanılan Hızlıca Keşfeden Rastgele Ağaç algoritması, rastgeleliğe dayalı olduğu için her çalıştırılışında farklı sonuçlar vermektedir. Bu çalışmada, önceden verilen sınır koşulları ve kinodinamik kısıtlar göz önünde bulundurularak elde edilen çok sayıda uygulanabilir yörüngeden yalnız biri örnek olarak sunulmuş ve hareket görselleştirilmiştir.

Hareket için gerekli torklar, dinamik kararlılık ve manipülatörün kontrolü gibi dinamik hususlar bu çalışmanın merkezinde yer almamış olup sadece elde edilen yörüngenin robota nasıl beslenebileceğine bir örnek olarak, yaklaşık hesaplanmış tork denetleyicisinin kullanıldığı bir benzetim, sonuçlarıyla paylaşılmıştır. Benzetimde Coriolis etkileriyle viskoz ve dinamik sürtünmeler modellenmemiş olmasına karşın planlanan yörüngenin çok küçük bir hata ile takip edilebildiği görülmüştür.

Çalışmanın devamı için öneri olarak başka optimizasyon algoritmalarının denenerek sonuçların Nelder-Mead yöntemindeki ile karşılaştırılması, çalışmada kullanılan yerel planlayıcının dinamik Hızlıca Keşfeden Rastgele Ağaç yöntemine uyarlanması ve algoritmanın asimptotik olarak optimum olan sürümü için denenmesi sunulmaktadır.

Hızlıca Keşfeden Rastgele Ağaç yaklaşımının alt fonksiyonları tek tek alınacak olursa; uzaklık ölçeği olarak geleneksel olan Öklid mesafesi yerine Mahalanobis mesafesi gibi diğer seçenekler denenebilir, kullanılan en yakın komşu algoritması yerine ise yaklaşık en yakın komşu algoritmaları kullanılarak planlayıcı hesaplama süresi kısaltılabilir. Ayrıca daha yörünge süreklileştirme aşamasına geçmeden önce, görüş hattı yaklaşımlarından yararlanılarak robotik sistemin eylemi gerçekleştirme zamanı da düşük bir hesaplama maliyeti karşılığında büyük ölçüde azaltılabilir.

Çalışmada süreklileştirme için önerilen yaklaşım, Olasılıksal Yol Haritası ve Rastgele Potansiyel Alanlar gibi bağlanabilirlik yolu veren diğer algoritmalar için aymen çağrılabilir ve yöntemlere özgü olarak elde edilecek sonuçlar değerlendirilebilir.

Son olarak, montaj aşamasında olan robot kolu çalışır duruma geldiğinde geliştirilen yaklaşımın deneysel olarak da test edilmesi ve elde edilecek sonuçların benzetimden elde edilenle karşılaştırılması planlanmaktadır. Önerilen tüm yaklaşım seçenekleri kapsamlı bir çalışma ile deneysel ve benzetim sonuçlarıyla derlenebilir.

TRAJECTORY PLANNING OF A HUMANOID ROBOT ARM BY USING RAPIDLY-EXPLORING RANDOMIZED TREE METHOD

SUMMARY

The determination of the motion of a robotic system or an autonomous vehicle from one position to another following which path, what velocity and acceleration profiles is defined as trajectory planning. Although trajectory planning is one of the issues of robotics that is most studied on, it is not possible to talk about a unique and optimum solution for this problem. This also goes for scenarios that necessitate avoiding global obstacles. In fact, the problem inevitably becomes more complex.

A wide range of studies exists in path planning including both deterministic and stochastic approaches. Deterministic ones such as evaluating all possible configurations in discretized configuration space are available and they are generally complete which means they give an exact solution in a finite amount of time. However, these methods are usually computationally inefficient for high-dimensional spaces and as a matter of fact, they often fail since they generally give one possible solution which can be infeasible.

Different methods have been used for trajectory planning, such as fuzzy logic control, neural networks, ant colony optimization, genetic algorithms; but among stochastic methods, sampling-based heuristic algorithms have been accepted due to the fact that they are simple and the cost of calculation is low. The most common ones of these methods are Probabilistic Roadmap Method, Randomized Potential Fields and Rapidly-exploring Randomized Tree algorithms and their derivatives.

We cannot talk about completeness for the main sampling-based planning algorithms. In other words, unless there is a solution, it won't be possible to determine a no-solution-situation with these methods. Therefore, for sampling, it will be necessary to limit either the number of samples or the period of sampling. On the other hand, if enough number of samplings is done, a feasible solution -if there is one-, will be found; meaning, it is probabilistically complete.

Sampling-based algorithms work together with sub-functions, such as determining which samples will connect with each other according to a pre-selected metric, planning the velocity and acceleration profiles it will use to go from the starting point to the target point and making collision checks for the planned path.

In this study, first of all, Rapidly-exploring Randomized Tree method has been called upon for the obstacle avoided trajectory planning of the six-degree-of-freedom ITECH Humanoid Robotic arm.

As planning space, the configuration space where the joints are, have been used, thus the calculations of inverse geometry and inverse kinematic have been avoided. And instead of transferring the target point from the task space to the configuration space, it should be determined that the projection of the expanding tree in the configuration space onto the task space is near enough to the target position. In order for this to occur, position and orientation tolerances have been defined.

After obtaining the connectivity path through Rapidly-exploring Randomized Tree method, cubic polynomial joint trajectories have been calculated between each viapoint as a local planner sub-function. For this, a study carried out for manipulators

has been used as a reference. The expression of joint positions with cubic splines means that joint velocity profiles are quadratic whereas joint acceleration profiles are linear. That is to say, with this approach, position, velocity and acceleration profiles are continuous.

In the next step, the optimization of the cubic trajectories, where the total time of the motion has been dealt with as the cost function, was done; and during this process not only boundary conditions but also velocity, acceleration and jerk limits were considered. Nelder-Mead search method was used as an optimization algorithm.

This study is a contribution to the literature due to the fact that smoothing of the trajectory given by Rapidly-exploring Randomized Tree method is clearly defined, time optimization for the given viapoints is done and jerk is limited.

Following the smoothing process, collision checks are done on the trajectory in order to enable both the end-effector and the whole robotic arm to move without hitting obstacles.

The Rapidly-exploring Randomized Tree algorithm that has been used, gives different results every time it is called upon as it is based on randomness. In this study, the pre-given boundary conditions and kinodynamic constraints have been taken into consideration, and from the many feasible trajectories only one has been presented as an example, and the motion has been visualised.

Dynamic issues such as the torques that are necessary for the motion, dynamic stability and the control of the manipulator have not taken place at the center of this study. In order to show how the obtained trajectory can be fed into the robot; the simulation results, where an approximate torque controller has been used, were shared. In the simulation, although Coriolis effects and viscous and dynamic frictions have not been modelled, it has been observed that the planned trajectory can be tracked with a very small position error.

For the future studies, it has been suggested that other optimization algorithms be called and the results compared with those of the Nelder-Mead method, the local planner used in the study be adapted to the dynamic Rapidly-exploring Randomized Tree method, and the algorithm be called for the asymptotically-optimal version of the method.

If the sub-functions of the Rapidly-exploring Randomized Tree approach were to be dealt with one by one; instead of the traditional Euclidean distance, other alternatives such as Mahalanobis distance can be tried, and instead of the nearest neighbour algorithm that has been used, approximate nearest neighbour algorithms can be used to reduce the calculation time of the planner.

Additionally, before the smoothing phase, line-of-sight approaches can be used and thus the time required for the robot to carry out the action can be greatly reduced with a low calculation cost.

The approach suggested in the study for the smoothing, can be called in a similar way for other algorithms such as Probabilistic Roadmap Method and Randomized Potansiyel Fields that give connectivity path, and specific results that will be obtained by these methods can be evaluated.

Lastly, when the robotic arm, which is at the assembly stage, is working, we plan to test the approach that has been developed and compare the results with those obtained by simulation. All the approach alternatives suggested can be compiled with experimental and simulation results in a comprehensive study.





1. GİRİŞ

Yol planlama ile, robotiğin başlıca çalışma konularından biri olarak sıklıkla karşılaşilmektedir. Öte yandan gerek mobil robotlar gerek otonom araçlar gerekse manipülatörler için tek, kesin ve optimum bir planlama çözümünden bahsedilemez. Bu durum engellerin bulunduğu bir çevrede planlama yapılırken de aynen geçerlidir. Sadece geometrik kısıtların dikkate alındığı bir planlama yapılacak olsa dahi bu meşakkatli bir iştir. Dinamik kısıtları da göz önünde bulunduran yörünge planlama ise, bu zorluğu bir adım öteye götürerek yüksek boyutlu durum vektörleriyle de uğraşmayı gerektirir.

1.1 Tezin Amacı

Bu tez çalışmasında, İstanbul Teknik Üniversitesi (İTÜ) Makina Fakültesi Prof. Dr. M. Nimet Özdaş Otomatik Kontrol Laboratuvarı'nda yürütülen ITECH İnsansı Robot Projesi'nin bir parçası olan 6 serbestlik dereceli insansı robot kolunun önceden tanımlanmış engellerden sakınan yörünge planlamasının yapılması amaçlanmıştır.

Planlama yapılırken manipülatörün geometrik kısıtları ile hız, ivme ve sarsım sınırlarının dikkate alınması isterler arasındadır. Ayrıca önerilecek yöntemin, başlangıç ve hedef noktada hız ve ivme değerlerinin önceden verilmesine olanak sağlaması ve bu değerleri dikkate alarak planlamayı gerçekleştirmesi istenir.

Hesaplama zamanı açısından verimlilik bu çalışma kapsamında ana amaç olmamakla birlikte, bu kısıtası sağlayan örnekleme tabanlı algoritmalar aynı zamanda sadelikleriyle de öne çıktığı için tercih edilmiştir. Kullanılan yöntemin birçok türevi olmasına karşın, yörünge süreklileştirme konusunda yeni bir yaklaşım ortaya konması amaçlandığı için en basit sürümü üzerinden tüm alt fonksiyonlar en baştan ele alınmıştır.

1.2 Literatür Taraması

Yol planlama üzerine gerek belirlenimci (deterministic) gerek rastlantısal (stochastic) çok sayıda yaklaşım vardır. Bunlardan belirlenimci olanlara örnek olarak ayrıklaştırılmış uzayda tüm olası konfigürasyonları değerlendirip sonlu bir zamanda kesin bir çözüm bulma yaklaşımı verilebilir; ancak buna benzer yöntemler hesaplama maliyeti açısından çok boyutlu uzaylarda verimsiz olacaktır (Canny ve diğ., 1988). Hatta bu yöntemler, genellikle tek bir çözüm önerdikleri için bu çözümün uygulanabilir olmaması durumunda başarısızlığa uğrayabilir. Öte yandan rastlantısal yaklaşımlar, hesaplama maliyetlerinin görece düşük olmasının yanı sıra birden fazla çözüm seçeneği de sunabilir.

Bulanık mantık kontrol, yapar sinir ağları, genetik algoritmalar, karınca koloni optimizasyonu ve benzetilmiş tavlama yaklaşımlarının her biri yol planlamada kullanılmıştır (Elbanhawi ve Simic, 2014). Öte yandan Rastgele Potansiyel Alanlar (Barraquand ve diğ., 1991), Olasılıksal Yol Haritası (Kavraki ve diğ., 1996) ve Hızlıca Keşfeden Rastgele Ağaç (HRA; Rapidly-exploring Randomized Tree, RRT) (LaValle ve Kuffner, 1999) gibi örnekleme tabanlı sezgisel yaklaşımlar son 20 yılda giderek yaygınlaşmıştır. Bunlardan HRA, holonomik olmayan kısıtlar ile yapılan planlamada ön plana çıkmıştır.

Gerek kinodinamik HRA gerek temel HRA oluşturma algoritmaları, geçiş noktalarını tanımlayıp kullanıcıya sarsak (jerky) yörüngeler verdikten sonra hareketi süreklileştirmek için kesin bir yol önermez. Geçiş noktalarının düz çizgilerle birleştirilip oluşturacağı yol, hareket çok yavaş olmadığı müddetçe uygulamada mümkün değildir. Öte yandan yapılacak yörünge süreklileştirme işlemi hız, ivme ve sarsım (jerk) sınırları dâhilinde bir hareket önermelidir.

Hauser ve Ng-Thow-Hing çalışmalarında sadece HRA ile elde edilen manipülatör yörüngelerini hız ve ivme sınırlarını dikkate alarak süreklileştirmekle kalmamış, ilaveten kestirme yollar kullanarak hedefe varış süresini kısa yolu hesaplama zamanı dâhil olmak üzere kısaltmıştır (2010). Bu çalışmaya dayanan yeni bir çalışma ise özellikle HRA'nın başarısız olduğu çok boyutlu planlama uzaylarında daha güvenilir çözümler önermektedir (Lau ve Byl, 2015).

Planlamanın yapılacağı uzayı belirlemek de dikkat gerektiren bir husustur. Klasik HRA algoritması (Kuffner and LaValle, 2000) örnekleme ve planlamayı

konfigürasyon uzayında yapmayı önerirken, Shkolnik ve Tedrake öncü GU-HRA algoritmalarıyla 1 dakikadan kısa sürede 1500 serbestlik dereceli bir sistem için görev uzayında planlama yapmayı başarmıştır (2009). Öte yandan bu algoritma, Jakobiyen matrisinin yalancı tersini kullanmayı gerektirir; ki bu her zaman uygulanabilir sonuçlar vermeyebilir. Konfigürasyon uzayında başlangıç ve hedef noktalardan birbirine doğru planlama yapan HRA-Bağla (RRT-Connect) algoritması (Kuffner ve LaValle, 2000) Diankov ve arkadaşlarına ilham kaynağı olmuştur. Çalışmalarında ortaya koydukları BiSpace Planning’de, konfigürasyon uzayındaki başlangıç noktasından bir ağaç ile görev uzayındaki hedef noktadan bir ağaç için aynı anda genişletilerek her iki uzay da kullanılmış olur (2008).

Her ne kadar çarpışma sınavasını görev uzayında yapmak daha kolay olsa da, manipülatör uygulamalarında ters kinematik model ile bir zorluk olarak karşılaşılmaktadır. Literatürde bu probleme birçok çözüm önerisi mevcuttur. Bunlardan Bertram ve arkadaşlarının çalışması (2006), bu çalışmaya da ilham vermiş ve detayları ileride açıklanacağı üzere hedef noktanınki dâhil olmak üzere ters kinematik çözüme ihtiyaç duymayan bir yaklaşım ortaya koymuştur. JD-HRA algoritması (Weghe ve diğ., 2007), bu çalışmayı bir adım daha öteye taşıyıp Jakobiyen matrisinin devriğini kullanarak yol planlama için hedef yönünde bir eğilim yaratmıştır. Tahmin edileceği üzere bu durum, yerel minimum olan bir noktaya takılıp kalma riskini de beraberinde getirir. BiSpace Planning ise ters kinematiği bir zorunluluktan ziyade bir seçenek olarak konfigürasyonlar civarında örnekleyerek karşımıza çıkar (Diankov ve diğ., 2008).

Son olarak, Perez ve arkadaşlarının PR2 deneysel platformu üzerindeki çalışmaları (2011) bahsetmeye değerdir. Bu çalışmada asimptotik optimum HRA algoritması olan ve HRA ile aynı hesaplama karmaşıklığına sahip HRA* (Karaman ve Frazzoli, 2011), 7 serbestlik dereceli tek kol ile 12 ve 14 serbestlik dereceli çift kol manipülatör uygulamaları için çağrılmış ve manipülatörün tutma öncesi duruşuna (pre-grasp pose) gelmesi sağlanmıştır. 12 serbestlik dereceli çift kol manipülasyon uygulamasında bilekteki yuvarlanma hareketinin engellendiği not edilir. Son olarak sonuçlar, HRA’nın verdikleriyle karşılaştırılmıştır (*ibid.*)

1.3 Yöntem

Bu tez çalışmasında, herhangi bir ters kinematik model kullanmaksızın HRA algoritması tarafından verilen geçiş noktalarının optimum zaman kıstası ile süreklileştirilmesi için manipülatörlere özgü bir yöntem önerilmiştir. Planlamada HRA algoritması kullanıldığı için engellerden sakınılabilmektedir.

Yörünge planlama yapılırken ilk ve son hız ve ivme bilgileri ile kinematik kısıtlar dikkate alınarak Lin ve arkadaşlarının (1983) formüle ettiği kübik eğriler uydurulmuş ve eğriler yine aynı çalışmanın devamında önerildiği üzere maliyet fonksiyonu toplam zaman olacak şekilde Nelder-Mead yöntemiyle optimize edilmiştir. Kübik eğrilerin kullanılması sadece konum ve hız için değil, ivme için de süreklilik sağlamıştır.

Son olarak oluşturulan sürekli yörünge, sınırlayıcı hacimler ve temel geometrik yaklaşımlarla çarpışma sınavına tâbi tutulmuş, çarpışma olması durumunda algoritmanın yeni bir yol önermesi için süreç başa döndürülmüştür.

Dinamik hususlar göz önünde bulundurulmaksızın önerilen yöntemin kapalı çevrim denetimli bir uygulaması için, örnek olarak, yaklaşık hesaplanmış tork ile PD denetleyicisi verilmiştir.

1.4 Tez Düzeni

Yörünge planlama geniş kapsamlı bir konu olup bu tez çalışmasında örnekleme tabanlı yöntemler üzerinden ele alınmıştır. Bunun için ilk olarak 2. Bölüm’de, temel kavramların tanımlarının verilerek geriye kalan kısmın açıklıkla değerlendirilmesini sağlamak hedeflenmiştir. Çalışma boyunca terimler için tezin de dili olan Türkçe kullanılmış; öte yandan, karışıklığa neden olmamak ve okuyucuya ileriki araştırmalarında kolaylık sağlamak amacıyla gerekli durumlarda terimlerin İngilizce karşılıklarının veya Türkçe eş anlamlılarının parantez içinde verilmesi uygun görülmüştür.

3. Bölüm’de HRA da dâhil olmak üzere en yaygın örnekleme tabanlı yörünge planlama algoritmalarından detaylı bir şekilde bahsedildikten sonra, çarpışma sınavı ve süreklileştirme alt fonksiyonları ele alınmıştır. 4. Bölüm yörünge planlaması yapılması istenen ITECH İnsansı Robot Kolu’nun temel kavramlar üzerinden tanıtımına ayrılırken; 5. Bölüm’de sürekli yörüngeler veren bir yöntem

önerilmiş ve yöntemin robot kol üzerindeki benzetimi detaylarıyla paylaşılmıştır. Son olarak Sonuç bölümünde, tez çalışmasının literatüre katkıları ile gelecek çalışmalara yönelik öneriler özetlenmiştir.

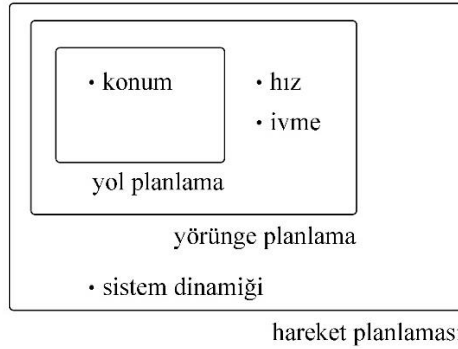




2. TEMEL KAVRAMLAR

Bir nesnenin, bir noktadan bir başka noktaya hangi yolu izleyerek aktarılacağı problemi yol (path) planlama olarak tanımlanır. Böyle bir planlamada esas olan, nesnenin engellere çarpmaksızın taşınmasıdır, bir başka deyişle nesnenin ve çevrede yer alan engellerin geometrisi göz önünde bulundurulur. Literatürde bu problem, yaygın olarak Piyano Taşıyıcısının Problemi (Piano Mover's Problem) olarak isimlendirilir (Schwartz ve Sharir, 1983). Piyano, kapladığı hacim itibarıyla bir noktasal kütle olarak ele alınamayacağı gibi, yönelimin önemli olmadığı küresel cisimler gibi de düşünülemeyecektir.

Yörünge (trajectory) planlama ise, daha geniş olarak, zaman değişimini de hesaba katarak yapılan yol planlamasıdır. Bu tanım, yer yer, eyleyiciler tarafından uygulanan kuvvet ya da torku dâhil edecek şekilde de yapılabilir ya da bu kapsamda olan planlama Şekil 2.1'de görüldüğü gibi hareket planlama olarak isimlendirilebilir. Bu aşamada, yol planlamadaki geometrik kısıtların yanı sıra kinematik ve dinamik kısıtlar da dikkate alınır.



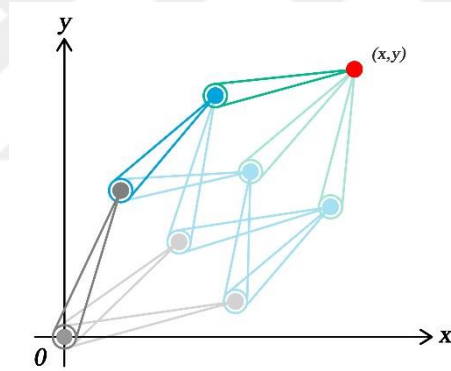
Şekil 2.1 : Planlama kavramlarının şematik gösterimi.

2.1 Planlama Uzayları

Bir robotik sistemin, geleneksel olarak Kartezyen koordinat sisteminde uç eyleyicinin erişebileceği tüm noktaların kümesi görev uzayı (G-uzayı) olarak tanımlanır. G-uzayı 3 boyutlu ya da sadece düzlemsel hareketin olduğu bir sistemde, 2 boyutlu olabilir. Yönelimlerin de görevin yapılmasında rol oynadığı durumda, sırasıyla, 6 ve 3 boyutlu

G-uzayından bahsedilebilir. Otonom araçlar ile mobil robotlarda planlama genellikle bu uzayda gerçekleştirilir. Örneğin, bir insansız hava aracının izleyeceği yörüngenin planlamasında yani tırmanma, seviye uçuşu, alçalma gibi uçuş evrelerine geçeceği yerlerin belirlenmesinde G-uzayı kullanılır.

n serbestlik dereceli bir robotik sistem için, G-uzayından ayrı olarak, n boyutlu bir konfigürasyon uzayı (K-uzayı) da tanımlanabilir. Eklem uzayı olarak da adlandırılabilen bu uzay, G-uzayına benzer şekilde, tüm olası konfigürasyonlardan meydana gelir. Bu konfigürasyonlardan uç eyleyicinin engel üzerinde olmayanları, $K_{serbest}$ 'i oluşturur. $K_{serbest}$ kümesindeki her noktanın, G-uzayında tek bir karşılığı vardır. Öte yandan, G-uzayındaki bir noktanın K-uzayında birden fazla sayıda karşılığı olabilir. Bu duruma örnek olarak Şekil 2.2'de, 3 serbestlik derecesine sahip düzlemsel bir robot kol verilmiştir. Görüldüğü üzere uç nokta, aynı konuma 6 farklı konfigürasyonla erişebilmektedir. Hatta son uzvun yöneliminin de aynı olduğu ikişer konfigürasyondan söz edilebilir.



Şekil 2.2 : G-uzayındaki bir noktanın K-uzayında birden fazla karşılığı olması durumunun bir örneği.

Artıksıl (redundant) yani serbestlik derecesi görev uzayının boyutundan (m) fazla olan robotlar için ise, G-uzayındaki bir noktanın K-uzayında sonsuz sayıda karşılığı vardır.

Planlama hangi uzayda yapılmış olursa olsun, eyleyicilerin bulunduğu K-uzayına geçmek kaçınılmazdır. Bu açıdan K-uzayında planlama, G-uzayına göre avantajlıdır. Diğer taraftan, G-uzayı görevin tanımlandığı uzay olduğu için en az bir kez kullanılmak durumundadır. K-uzayından G-uzayına geçiş ileri geometrik model, G-uzayından K-uzayına geçiş ise ters geometrik model ile gerçekleştirilir. Şayet planlama hız bilgisini de içeriyorsa, sırasıyla, ileri ve ters kinematik modeller de kullanılabilir.

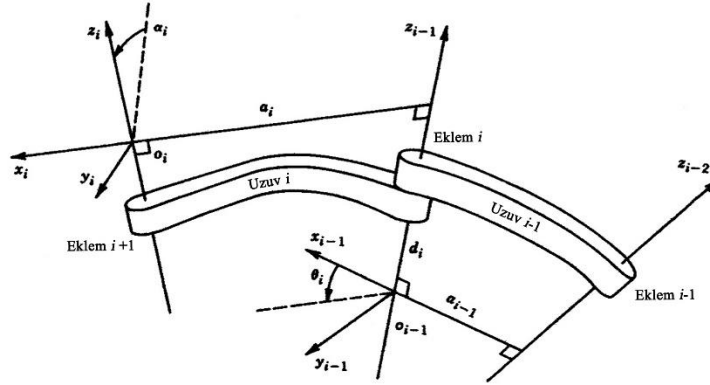
2.2 Geometrik ve Kinematik Modeller

2.2.1 İleri ve ters geometrik modeller

İleri geometrik modelde, K-uzayında tanımlı noktalar G-uzayındaki karşılıklarıyla bir fonksiyonla ilişkilendirir. Bunun için manipülatörlerde, yaygın olarak, bir eksen takımının diğer bir eksen takımına göre ifade edilmesine yarayan dönüşüm (transform) matrisleri kullanılır. Bu matris, 3 boyutlu G-uzayı için 2.1 numaralı denklemde verildiği üzere, dönme (rotation) matrisi (R) ve konum vektöründen (o) meydana gelir.

$$T = \begin{bmatrix} R_{3 \times 3} & o_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (2.1)$$

Eksen ve parametrelerin tanımlanmasında kullanılan Denavit-Hartenberg (D-H) konvansiyonunun verildiği Şekil 2.3'teki ilişkiye sahip ardışık eklemler için dönüşüm matrisi, 2.2'de verilmiştir. Uç eyleyicinin konum ve yönelim bilgisini içeren eksen takımının (F_e) temel eksen takımına (F_0) göre ifadesi, 2.3'te görüleceği üzere ardışık eksen takımlarının (F_i) dönüşüm matrislerinin çarpılmasıyla elde edilir.



Şekil 2.3 : D-H konvansiyonuna göre eksenlerin belirlenmesi ve diğer eksen parametreleri (Spong, Hutchinson ve Vidyasagar, 2006).

$$T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$A = \sum_{i=1}^n T_i \quad (2.3)$$

İstenen konum ve yönelim için eklem değişkenlerini veren ters geometrik modelin elde edilmesi ise nazaran daha zordur. Modelin birden fazla ya da sonsuz sayıda çözümü olabilir veyahut hiç çözüm bulunmayabilir. Khalil ve Dombre, artıksıl olmayan manipülatörler için geometrik modelin çözülebilir olduğunu ifade etmiştir (2004). 6 ve 7 serbestlik dereceli manipülatörler için çeşitli analitik çözümler önerilmiştir (Hollerbach, 1984; Megahed, 1992; Tolani ve Badler, 1996; Asfour ve Dillmann, 2003).

2.2.2 İleri ve ters kinematik modeller

2.4 numaralı denklemde verilen ileri kinematik modelde, uç eyleyici hız vektörü $\dot{X}$ ile eklem hızları vektörünü ($\dot{q}$) Jakobiyen matrisi ilişkilendirir. Jacobiyen matrisi, $m \times n$ boyutunda bir matris olup burada m , X vektörünün eleman sayısıdır ve yönelimin de göz önünde bulundurulduğu 3 boyutlu bir uzayda bu sayı, 6'ya eşittir.

$$\dot{X} = J\dot{q} \quad (2.4)$$

$J = [J_1, J_2, \dots, J_n]$ olmak üzere, Jakobiyen matrisine ait n adet sütun 2.5'teki gibi hesaplanır. Burada, z_{i-1} , $i - 1$. eklemin dönme matrisinin üçüncü sütunudur.

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix}, & i \text{ döner eklem ise} \\ \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix}, & i \text{ kayar eklem ise} \end{cases} \quad (2.5)$$

Ters kinematik model ise, verilen uç eyleyici hızı vektöründen eklem hızlarının ne olması gerektiğini hesaplamakta kullanılır. Bunun için, 2.6 numaralı denklemde Jakobiyen matrisinin tersinden faydalanılır.

$$\dot{q} = J^{-1}\dot{X} \quad (2.6)$$

Öngörülebileceği üzere, Jakobiyen matrisinin determinantının 0'a eşit olduğu durumlarda matrisin tersini almak mümkün olmayacaktır. Bu durumun meydana geldiği noktalar, manipülatörün tekil (singüler) noktaları olarak isimlendirilir. Ters kinematik problemiyle, tekil noktalar civarında da karşılaşılır. Bu bölgelerde, Jakobiyen matrisinin tersi alınabilse dahi, determinantının sıfıra (0) çok yakın olduğu durumlarda, hesaplanan eklem hızları uygulanabilir olmayacaktır.

2.2.3 Tekil nokta civarında ters kinematik çözümü

Tekil nokta civarında ters kinematik probleminin çözümü için çeşitli sayısal yöntemler önerilmiştir. Bunlardan ilki, Jakobiyen matrisinin 2.7’de verilen yalancı tersini (psuedoinverse) kullanmaktır. Burada yer alan JJ^T veya J^TJ ifadeleri Jakobiyen matrisinin determinantı sıfır olmadıkça tersi alınabilir olmasına karşın, verdiği çözüm önerilerinin tekil nokta civarında kararlılık (stabilite) sorunu vardır (Aristidou ve Lasenby, 2009).

$$J^\dagger = J^T(JJ^T)^{-1} = (J^TJ)^{-1}J^T \quad (2.7)$$

Ters kinematik çözümü için bir başka yöntem olarak, 2.8’de, Jakobiyen matrisinin tersi yerine devriğinin uygun bir katsayı (β) ile çarpımının kullanılması önerilmiştir. α ’nın değeri, nokta çarpımın $\odot$ ile gösterildiği 2.9 numaralı denklemden hesaplanır (*ibid.*).

$$\dot{q} = \beta J \dot{X} \quad (2.8)$$

$$\alpha = \frac{\dot{X} \odot (JJ^T \dot{X})}{(JJ^T \dot{X}) \odot (JJ^T \dot{X})} \quad (2.9)$$

Sönümlenmiş en küçük kareler (damped least squares) yaklaşımı ise Jakobiyen matrisinin tersi yerine sıfırdan farklı sönüm sabitini (Λ) içeren 2.10 ifadesi kullanır. Bu ifadede sönüm sabitinin seçimi önemlidir. Çok büyük seçildiği takdirde yakınsama hızı düşük olurken, yeterince büyük olmalı ki uygulanabilir çözümler versin (*ibid.*).

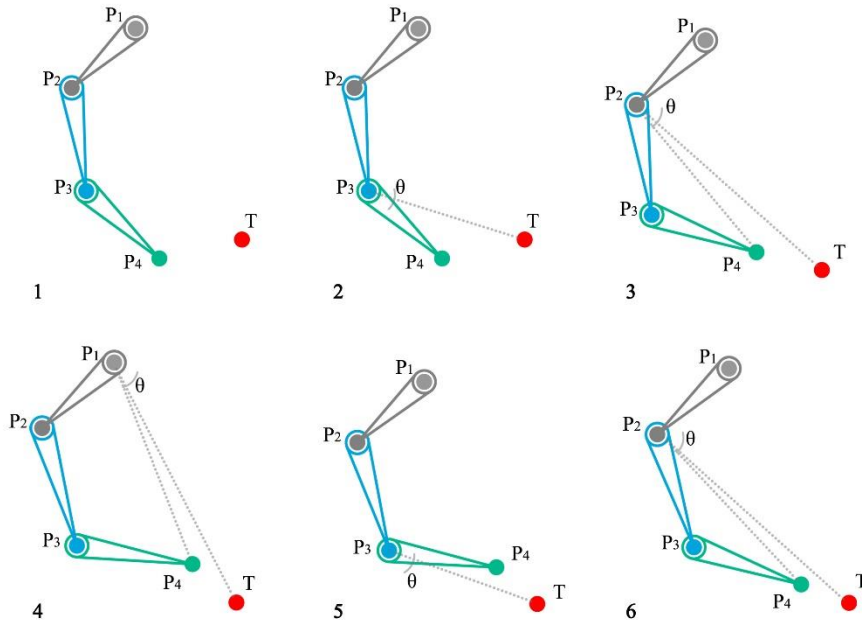
$$J^{-1} = J^T(JJ^T + \Lambda^2 I)^{-1} \quad (2.10)$$

Sönümlenmiş en küçük kareler ile yalancı ters yöntemleri, tekil değer ayrışımı (singular value decomposition) ile geliştirilebilir. Örneğin, en küçük kareler için tekil nokta civarında elde edilen çözümler daha pürüzsüz olabilir; ancak bu, hesaplama maliyetini de beraberinde getirecektir (*ibid.*).

Titizlikle sönümlenmiş en küçük kareler (selectively damped least squares) metodu ise, tekil değer ayrışımının kullanıldığı sönümlenmiş en küçük kareler yöntemini bir adım öteye taşıyarak, sönüm sabitini hedef konuma ulaşma zorluğuna göre ayrılmış Jakobiyen matrisi vektörleri için tekrar tekrar ayarlayarak daha iyi sonuçlar verir. Bu yöntem, iyi bir sönüm sabiti seçme zorluğunun önüne geçse de hesaplama maliyetini daha da artırmaktadır (*ibid.*).

Son yıllarda Pechev tarafından ortaya atılıp patenti alınan geri beslemeli ters kinematik (feedback inverse kinematics) yaklaşımı, yukarıda bahsedilen tüm matris tersi alma, tekil değer ayrışımı ve sönüm sabiti hesaplama gibi yüklerden kullanıcıyı kurtararak tekil noktalardan kaçmayı vaat etmektedir (2008).

Son olarak, bu çözüm önerileri dışında; hesaplaması zor ve zaman alıcı olan Hessiyen matrisini içeren Newton yöntemi, insan hareketlerini temel alan yöntemler ve hesaplama maliyeti düşük sezgisel algoritmalar mevcuttur. Sözü edilen bu son yöntem grubu, G-uzayından K-uzayına geçmek için doğrudan bir çözüm sunmaktan ziyade, G-uzayındaki noktaların K-uzayındaki karşılığını rastlantısal bir şekilde bulur. Şekil 2.4'te, sezgisel yöntemlere örnek olarak, Döngüsel Koordinat Yaklaşması (Cyclic Coordinate Descent) verilmiştir. Görüldüğü üzere, uçtan başlayarak manipülatörün tüm uzuvları G-uzayındaki nokta doğrultusunda hareket ettirilir ve bu işlem, uç eyleyici hedef noktaya ulaşıncaya veya aynı sonucu vermeye başlayana dek devam eder (Aristidou ve Lasenby, 2009). Hesaplama açısından verimli olan bu yöntem, küresel kısıtların göz önünde bulundurulduğu uygulamalarda, uygulanabilir bir çözüm vermekte sıkıntı yaşamaktadır.



Şekil 2.4 : Ters kinematiğe çözüm olarak kullanılan sezgisel algoritmalarından Döngüsel Koordinat Yaklaşması.

2.3 Kısıtlar

Yörünge planlamanın robotik alanında uzun süredir çalışılan bir konu olmasının başlıca sebeplerinden biri, hareketi kısıtlayan unsurlardır. Bu unsurlar, doğrudan robottan kaynaklanan yerel kısıtlar olabileceği gibi, G-uzayındaki engeller yani küresel kısıtlar da hareket planlamasında göz önünde bulundurulması gereken hususlardır. Küresel ve yerel kısıtlar, senaryolar ile daha karışık haller alabilir.

2.3.1 Yerel kısıtlar

Bir robota ait yerel kısıtlar, robotun küresel konumu ne olursa olsun hareketini sınırlandırır. Yerel kısıtlar, geometrik ve kinodinamik olarak ikiye ayrılabilir. Geometrik kısıtlar, robotun mekanik tasarımı ve konfigürasyonu ile doğrudan ilişkilidir. Kendi kendine çarpma durumu, uzuvları kaplayan kılıf ve robotun kablolarının zarar görmemesi gereği de dikkate alınarak engellenmek istenir. Sistemin holonomik olup olmayışı da yerel kısıtlar arasında sayılabilir. Holonomi ile ilgili olarak, arabalardaki paralel park örneği verilebilir. Aracın holonomik olmayışı, park planlamasını doğrudan etkilemektedir.

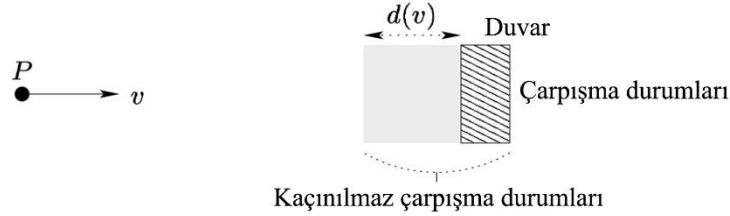
Kinodinamik yerel kısıtlar, eklemlerin eyleyicisi olan motorlar ile bu motorların sürücülerinden kaynaklı limitlerdir. Eklemin dönme hızı ile ekleme uygulanabilecek en büyük tork değeri doğrudan bu sınır değerlerle ilişkilidir.

2.3.2 Küresel kısıtlar

Robotun hareketini, kendinden kaynaklanan yerel kısıtlar dışında, G-uzayında konumlandırmasına bağlı küresel kısıtlar da sınırlayabilir. Bu kısıtlar hareketli ya da hareketsiz engellerden meydana gelir. Tahmin edilebileceği üzere, hareketli engellerin bulunduğu durumlarda zamandan bağımsız yol planlaması yetersiz olacaktır. Bu engellerin, nerede ve ne zaman olacağı bilgisi mevcut değilse çevrimdışı yörünge planlama da yetersiz kalıp yerine algılayıcıları da içeren uyarlamalı (adaptive) yöntemler çağrılabilir.

Fraichard ve Asama (2003), küresel kısıtların kapsamını bir adım öteye taşıyıp kaçınılmaz çarpışma (inevitable collision) durumunu ilk defa detaylı bir şekilde tanımlanıp incelemiştir. Buna göre, sistemin takip edeceği yörünge ne olursa olsun çarpmanın kaçınılmaz olduğu haller kaçınılmaz çarpışma durumu olarak ifade edilir. Şekil 2.5'te 1 boyutta hareket eden v çizgisel hızına sahip bir noktasal kütle için

çarpışma ve kaçınılmaz çarpışma durumları gösterilmiştir. Kaçınılmaz çarpışma durumlarının oluşturduğu alan, doğal olarak v ile doğrudan ilişkilidir ve en basit tahminle, hızın fazla olması durumunda kaçınılmaz çarpışma durumlarının da fazla olması beklenir. Aynı çalışmada, hareketli engellerin bulunduğu çevreler için de kaçınılmaz çarpışma durumları ele alınmıştır (*ibid.*).



Şekil 2.5 : Çarpışma ve kaçınılmaz çarpışma durumları (Fraichard ve Asama, 2003).

2.3.3 Diğer kısıtlar

Bir robotik sistemin, yerel ve küresel kısıtlar çerçevesinde yapabileceği bir hareketi diğer bazı kısıtlardan dolayı yapmaması tercih edilebilir. Örneğin bu kısıtlar, doğadan esinlenilerek yine ona benzer olma şartından gelebilir. Nasıl ki bir insan başının 180 derece dönerek tam arkaya bakması hem olanaksız hem de alışılmadık bir durum ise, özellikle insanlarla etkileşime geçecek bir insansı bir robotun da bu hareketi yapması aynı ölçüde rahatsız edici olacaktır.

Bu tez çalışması kapsamında da, ele alınan 6 serbestlik dereceli robotik manipülatörün insansı hareket sergilemesi isteri olması durumunda, insan kolunun hem geometrik hem de kinodinamik hareket sınırları dikkate alınmalıdır. Çizelge 2.1’de, Rosen ve arkadaşlarının günlük etkinliklerde insan kolunun geometrik ve kinodinamik sınırlarını ortaya koyduğu çalışması (2005) özetlenmiştir. Aynı çalışma, eklem hızı profillerinin de yapılan işe göre farklılık gösterdiğini ortaya koymuştur. Buna göre, insan kolu üst eklemleri (omuz ve dirsektekiler) kolu baş hizasına getirmek gibi kaba işlerde bilek eklemlerine göre daha hızlı, bir nesneyi bel hizasında hareket ettirmek gibi ince işlerdeyse daha yavaştır. Ayrıca ince işlerdeki bilek eklem hızı da, kaba işlerdeki omuz ve dirsek eklem hızlarına göre de daha fazladır (*ibid.*).

Literatürdeki insansı robot kolu uygulamalarından derlenen Çizelge 2.2’den de görüleceği üzere, eklem hareketlerinin geometrik sınırlandırmasında insandaki alt ve üst sınırlar yaklaşık olarak dikkate alınmıştır.

Çizelge 2.1 : Günlük aktivitelerde insan kolunun sınırları (Rosen ve diğ., 2005).

	eklem açısı[°]		eklem açısal hızı[°/s]		eklem açısal ivmesi[°/s ²]		Tork (mNm)	
	alt sınır	üst sınır	minimum	maksimum	minimum	maksimum	minimum	maksimum
Omuz (yunuslama)	-18,3	62,3	-136,7	99,9	-949,7	924,4	-4869,4	9626,0
Omuz (sapma)	-14,0	133,6	-134,1	171,5	-1282,7	1311,7	-3344,3	10057,0
Dirsek (yalpa)	-71,7	39,5	-140,7	113,0	-1049,0	1020,3	-2040,8	3100,8
Dirsek (sapma)	41,7	162,9	-172,8	145,8	-1266,1	1214,3	-1561,3	3761,1
Bilek (yalpa)	-124,6	134,9	-412,8	486,3	-4543,6	3715,5	-40,6	43,5
Bilek (sapma)	-60,1	32,7	-180,4	203,9	-2476,2	1822,4	-153,9	373,8
Bilek (yunuslama)	-40,1	55,2	-141,2	232,9	-2790,1	2116,2	-173,0	245,8

Çizelge 2.2 : İnsansı robot kollarına ait geometrik hareket limitleri.

	JUSTIN (Zacharias, 2012)		ROMEO (Url-1)		Tondur ve diğ. (2005)		AILA (Lemburg ve diğ., 2011)	
	alt limit [°]	üst limit [°]	alt limit [°]	üst limit [°]	alt limit [°]	üst limit [°]	alt limit [°]	üst limit [°]
Omuz (yunuslama)	-170	170	-82,7	127,2	0	100	-360	360
Omuz (sapma)	-120	120	-24,7	65,3	0	180	-10	100
Dirsek (yalpa)	-170	170	-120	120	-100	100	-360	360
Dirsek (sapma)	-120	120	0	90	0	150	-10	125
Bilek (yalpa)	-85	90	-210	30	-90	90	-360	360
Bilek (sapma)	-45	15	-25	25	-35	35	-45	45
Bilek (yunuslama)	-85	85	-56	56	-45	45	-15	15

Sistemler için kısıtlamalar güvenlik amacıyla da tanımlanabilir. Güvenlik konusundaki kısıtlar, diğer makina ve teçhizatınkine ilaveten robotlar için EN ISO 10218-1, EN ISO 10218-2 ve ISO/PDTS 15066 standartlarını da kapsar. 2015 yılı itibarıyla robotlarla aynı alanda güvenli çalışma ile ilgili 30 aktif Avrupa Birliği yönergesi ve 600’den fazla standart mevcuttur (Michalos ve diğ.). Bunlardan en önemlileri; uç eyleyicinin orta nokta hızının kontrol edilebilir ve 250 mm/s’den düşük olması, robotun hem koruyucu hem acil durdurma işlevlerinin olması ve güç ile kuvvetin kontrol sistemi tarafından sınırlandırılması olarak sayılabilir (*ibid.*).

2.4 Planlama Uzaylarının Karşılaştırması

Yukarıda görüldüğü üzere, ters kinematik problemi ile kısıtların göz önünde bulundurulmadığı bir yörünge planlaması düşünülemez. Bu açıdan planlama uzayları yeniden ele alınacak olursa, uzayların avantaj ve dezavantajları Çizelge 2.3’te özetlenmiştir. Buna göre, eklemler hâlihazırda K-uzayında bulunduğu için bu uzayda yapılan planlama ters kinematik ile tekil nokta problemlerinden bağımsızdır ve daha az hesaplama gerektirir. Öte yandan, görevler ve küresel kısıtlar genellikle G-uzayında tanımlanır. Örneğin, çoğu zaman holonomik olmayan robotik bir manipülatörün G-

uzayında düz bir çizgiyi takip etme görevi K-uzayında bu kadar kolay tanımlanamaz. Bu durumda ya K-uzayında planlanan yörünge ileri modeller yardımıyla G-uzayına aktarılıp küresel kısıtlara uygunluğu ve hedefe ulaşım ulaşmadığı Döngüsel Koordinat Yaklaşması Yöntemi'nde olduğu gibi test edilmeli ya da görev ile küresel kısıtlar ters modellerle K-uzayına aktarılmalıdır.

Yerel kısıtları K-uzayından G-uzayına aktarmak nazaran kolaydır; ki G-uzayı zaten kısıtlanmamış noktalardan meydana gelir. Küresel engelleri, K-uzayına aktarmak için ise çeşitli yaklaşımlar vardır; ancak bunların hiçbirisi tam olarak yeterli değildir. Bu yaklaşımların ilki, Lozano-Perez tarafından çokyüzlü küresel engeller için ortaya konmuştur (1983).

Çizelge 2.3 : Planlama uzaylarının birbirine üstünlükleri.

Planlama Uzayı	Avantaj
K-uzayı	Hareketi sağlayan eklemlerin bulunduğu uzayda olması Tekil noktalar ile problem olmaması Hesap yükünün daha az olması
G-uzayı	Küresel engellerin ve görevin tanımlandığı uzayda olması Düz çizgiler içeren bir şeklin takip edilebilmesi

3. ÖRNEKLEME TABANLI PLANLAMA

Genellikle K-uzayında rastgele örneklemeye dayanan yörünge planlama yöntemleri çok çeşitli olup konuyla ilgili detaylı ve güncel bir derlemeye Elbanhawi ve Simic'in çalışmasından (2014) ulaşılabilir. Bu kaynakta görüleceği üzere, bu yöntemlerin çoğu Olasılıksal Yol Haritası (OYH), Rastgele Potansiyel Alanlar (RPA) ve HRA algoritmalarının kendisi veya türevleridir.

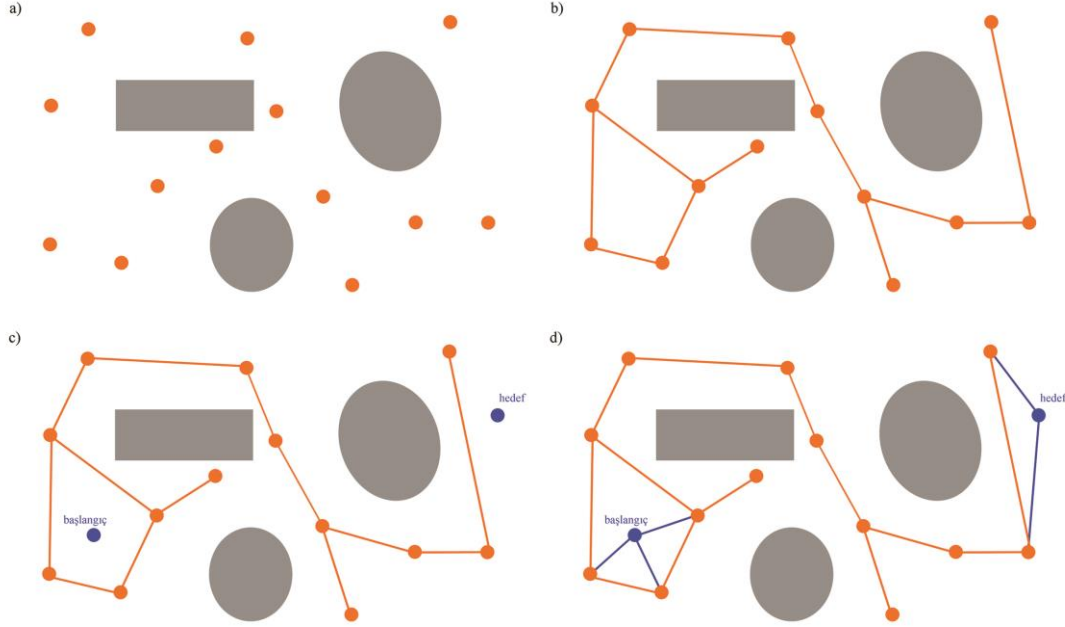
Bir algoritma için eksiksizlik (completeness), bir çözüm var ise bunu bulmayı yoksa da bir çözüm olmadığı çıktısını vermek anlamında ele alınacak olursa; bu üç temel algoritma da eksiksiz kabul edilemez. Öte yandan bu yöntemler, örnekleme sayısı sonsuza yakınsadıkça bir çözüm varsa bunu vermeyi vaat eder ve bu durum, olasılıksal eksiksizlik (probabilistic completeness) olarak tanımlanır.

3.1 Olasılıksal Yol Haritası Yöntemi

OYH algoritması, adından da anlaşılacağı üzere gerçek bir yol haritasına benzer. Şehirlerarası bir yolculukta bir yerden bir yere seyahat ederken başlangıç ve hedef noktalara en yakın anayollara bağlanıp onları izlemek gibi, önceden rastgele elde edilmiş bir yol haritası da yol planlaması yapılan sistem için kullanılabilir. HRA algoritması ile analogi olarak, oluşturulan yol haritası yer yer orman olarak da tanımlanır. Özellikle aynı çevrede birden fazla sefer yapılacak planlamalarda, yol haritasını bir kez oluşturmak yeterli olacağı için bu yöntem hesap yükü açısından avantajlıdır.

Şekil 3.1'de OYH yönteminin, 2 boyutlu bir uzayda yol planlaması 3 aşamada özetlenmiştir. Şekil 3.2'de ise, OYH algoritmasının ilk aşaması olan yol haritası oluşturma özetlenmiştir. Buna göre, N adet rastgele örnekleme yapıldıktan sonra, birbirine belli bir r mesafesinden yakın olan örnekler birer doğru parçası ile bağlanarak harita oluşturulur. Aynı esnada, örneklenen noktalar ile onları bağlayan doğru parçalarının $K_{serbest}$ 'te olduğu kontrol edilmelidir. Bu öğrenme aşaması tamamlandıktan sonra, başlangıç (q_{ilk}) ve hedef (q) konfigürasyonları, rastgele

noktalar ile aynı prosedürle yol haritasına katılır ve böylelikle, bu noktaları bağlayan en az bir adet yol bulunabilir. Yeterince örnekleme yapılmadığı takdirde, eksik olma durumunun bir getirisi olarak, bir çözüm olduğu halde bulunamayabilir.



Şekil 3.1 : OYH yönteminin aşamaları: a) N adet rastgele nokta oluşturma, b) yol haritası elde etme, c) başlangıç ve hedef noktaların ilavesi ve d) yolu bulma.

```

OYH_OLUŞTUR
1   G.baş() ; i=0
2   i<N olduğu sürece
3     α←ÖRNEKLE
4     eğer α ∈ Kserbest ise
5       G.ayrımnoktası_ekle(α); i=i+1
6     her q ∈ komşuları_belirle(α,G) için
7       eğer (α,G) bağlanır ise
8         G.yol_ekle(α,q)

```

Şekil 3.2 : OYH yönteminde yol haritası oluşturma algoritması.

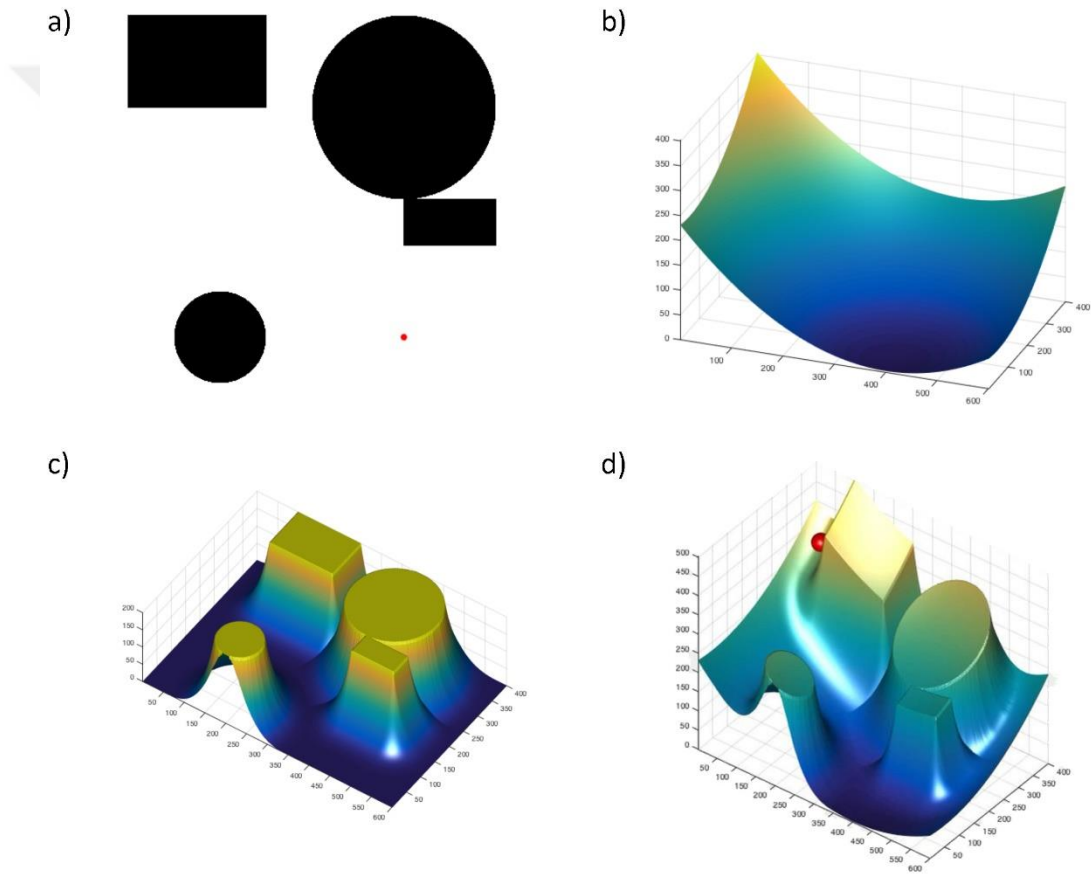
3.2 Rastgele Potansiyel Alanlar Yöntemi

RPA algoritması, genellikle, hedefe yakınlık ve küresel engellere uzaklığı göz önünde bulunduran iki ayrı fonksiyonu kullanarak noktalar için değer hesabı yapar. Bu fonksiyonlar, sırasıyla çekici (attractive) ve itici (repulsive) fonksiyonlar olarak tanımlanabilir ve toplamaları hesaplandığında, her nokta için birer değer elde edilir. Bu süreç, ceza ve ödül verme mantığına benzetilebilir.

Fonksiyon değerleri bir kez elde edildikten sonra, noktalar çevrelerindeki en çekici noktalara belli bir v hızında hareket ettirilir. v hızı, q 'ya yakın noktalarda

ölçeklendirilerek bu noktada durma sağlanabilir. Son aşamada, başlangıç noktasından q 'ya hız vektörleri takip edilerek ulaşılabilir. Başlangıç noktası değişse dahi elde edilen potansiyel alanların, OYH'deki yol haritaları için olduğu gibi, kullanılabilmesi, bu yöntemin önemli bir avantajıdır.

Öte yandan, tahmin edileceği üzere, bu yöntem dar geçitler ve yerel minimum noktalardan muzdarip olmaya çok açıktır. Dolayısıyla, her ne kadar birçok sezgisel yöntemle ilham vermiş olsa da çok fazla sezgisel değişkenin (parametre) tanımlanması gereği dolayısıyla da dezavantajlı bir yöntemdir (LaValle, 2006). Şekil 3.3'te RPA yönteminin bir uygulaması görülebilir.



Şekil 3.3 : a) Önceden bilinen hedef nokta ve engeller, b) hedef noktaya yakınlık fonksiyonu, c) engellere uzaklık fonksiyonu ve d) iki fonksiyonun toplamı (Taylor, 2016).

3.3 Hızlıca Keşfeden Rastgele Ağaç Yöntemi

Temel HRA yöntemi, q_{ilk} ve q 'nın bulunduğu K-uzayında bu iki noktadan birinden diğerine bir ağacı (T), Şekil 3.4'te verilen algoritma doğrultusunda genişleterek yörünge bulmayı amaçlar. Algoritmanın kullanılması için ilk olarak $K_{serbest}$ ile q_{ilk}

ve q 'nın önceden tanımlanmış olması gereklidir. Tezin geri kalan bölümünde, ağaç genişlemesinin her zaman q_{ilk} 'ten başladığı kabul edilecektir.

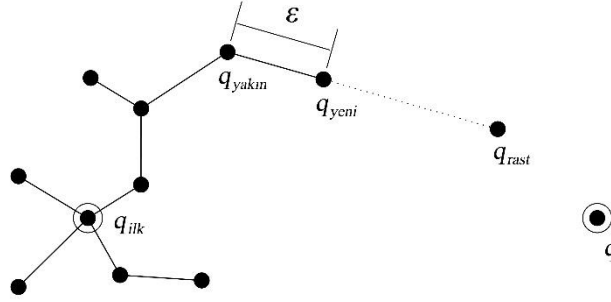
HRA_ OLUŞTUR(q_{ilk}, q)	
1	$T.baş(q_{ilk})$
2	$k=1$ 'den K 'ye
3	$q_{rast} \leftarrow \text{RASTGELE_DURUM}()$
4	$\text{GENİŞLET}(T, q_{rast})$
5	T 'yi ver
GENİŞLET(T, q_{rast})	
1	$q_{yakın} \leftarrow \text{ENYAKIN_KOMŞU}(q_{rast}, T);$
2	eğer $ q_{yakın} - q_{rast} < \varepsilon$ ise
3	$q_{yeni} = q_{rast}$
4	aksi halde
5	$q_{yeni} = q_{yakın} + \varepsilon$
6	$T.\text{ayrımnoktası_ekle}(q_{yeni});$
7	$T.\text{dal_ekle}(q_{yakın}, q_{yeni});$
8	eğer $ q - q_{yeni} < \mu$ ise
9	$T.\text{ayrımnoktası_ekle}(q);$
10	$T.\text{dal_ekle}(q_{yeni}, q);$
11	<i>Ulaşıldı</i>
12	aksi halde
13	<i>İlerletildi</i>

Şekil 3.4 : Temel HRA algoritması.

Şekil 3.5'te HRA'nın genişleme ilkesi 2 boyutlu bir uzay için gösterilmiştir. Buna göre, K-uzayında üretilen rastgele bir nokta (q_{rast}) şayet bir engel üzerinde değilse yani $K_{serbest}$ 'te ise başlangıçta sadece q_{ilk} 'ten oluşan dallanma noktaları (vertex) kümesinden kendine en yakın olan noktayı ($q_{yakın}$) seçer. Eğer bu iki nokta arasındaki mesafe, belli bir ε değerinden küçük ise noktalar bir doğru parçasıyla birleştirilir ve q_{rast} , dallanma noktaları kümesine katılır. Aksi durumda, yine genişleme gerçekleşecektir; ancak ε uzunluğundaki bir doğru parçası $q_{yakın}$ 'dan q_{rast} doğrultusunda uzandığında hangi noktaya denk geliyorsa o nokta (q_{yeni}) ve $q_{yakın}$ ile q_{yeni} 'yi birleştiren doğru parçası ağaca ilave olacaktır.

En yakın komşunun belirlenmesi için, uygun bir ölçeğin (metric) seçilmesi gereklidir. Yaygın olarak bu ölçek, Öklid mesafesi olarak tanımlansa da Manhattan ve Minkowski mesafeleri de mevcuttur (Amato ve diğ., 2000). Bu mesafelerin hesabına ilişkin denklemler, 2 boyutlu XY düzleminde koordinatları (x_1, y_1) ve (x_2, y_2) olan iki nokta için 3.1-3'te verilmiştir. Görüldüğü üzere, Minkowski mesafesi Öklid mesafesinin bir r parametresi ile daha genel bir ifadesidir (*ibid.*). Döner eklemler, sonsuz dönme kapasitesine sahipse bu durum en yakın dallanma noktası bulunurken göz önünde bulundurulmalıdır. Yönelimin dikkate alındığı bir yol planlamada ise, yönelim farkı

terim olarak aynen ilave olabileceği gibi önemine göre bu ifadenin ölçeklendirilmesi de mümkündür. Son olarak, Glassman ve Tedrake'in ortaya attığı sistemin dinamiği ile kısıtlarını dikkate alan ve kesin minimum-zaman sözde ölçeğine (exact minimum-time pseudometric) yaklaşan çalışmasından (2010) söz edilebilir. En yakın dallanma noktasını bulmak, nokta sayısı arttıkça zaman alıcı olacağı için yaklaşık en yakın komşu (approximate nearest neighbor) algoritmalarından da faydalanılabilir.



Şekil 3.5 : Rastgele ağacın genişleme ilkesi.

$$d_{öklid} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.1)$$

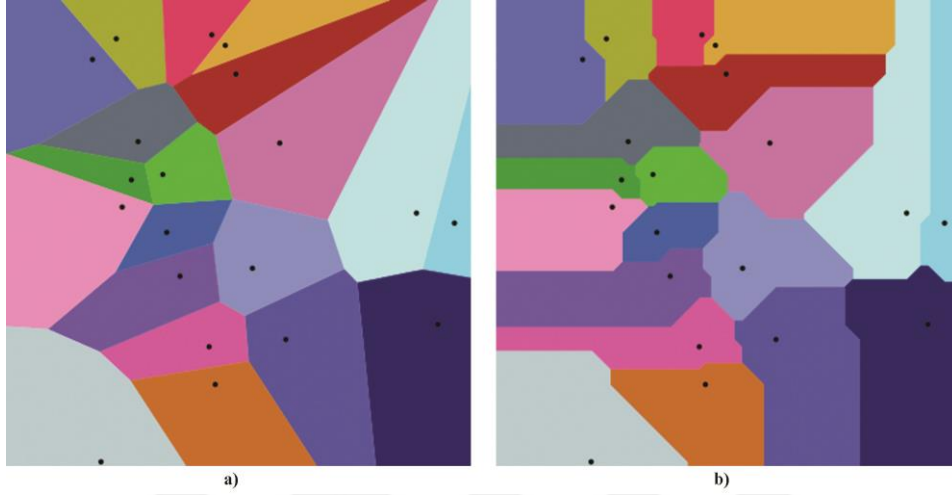
$$d_{manhattan} = (x_2 - x_1) + (y_2 - y_1) \quad (3.2)$$

$$d_{minkowski} = \sqrt[r]{(x_2 - x_1)^r + (y_2 - y_1)^r} \quad (3.3)$$

Bir düzlemdeki tüm noktalar uygun bir ölçeğe göre hâlihazırda var olan ayırım noktaları ile ilişkilendirilecek olursa, aynı ayırım noktalarıyla ilişkilendirilen noktalar birer alan oluşturur ve bu alanlar, Voronoi hücresi olarak isimlendirilir. 3 boyutlu bir uzayda da benzer şekilde 3 boyutlu Voronoi hücrelerinden söz edilebilir. Voronoi hücrelerinin büyüklüğü ile orantılı olarak, rastgele noktanın o hücreye düşme olasılığı da artacaktır. Bu özellik, HRA algoritmasına keşfedilmemiş bölgeleri hızlıca aydınlatma kabiliyeti verir. Şekil 3.6'da iki farklı ölçeğe göre oluşturulan 20 noktaya karşılık gelen Voronoi hücreleri görülmektedir.

Hedef noktaya ulaştıktan sonra yol, bu noktaya hangi noktalardan geçerek geldiğinin kaydına bakılarak ağaçtan elde edilebilir. Şayet böyle bir kayıt, kısıtlı bellek ya da başka bir sebepten dolayı yapılması ise oluşturulan ağaç bir labirentin tüm geçeneklerine (koridor) yayılmış gibi düşünülerek labirentten çıkma algoritmaları çağrılarak yol bulunabilir. Bunlardan biri olan çıkmaz doldurmanın (dead-end filling) HRA'ya uyarlanmış sürümü, Dalları Kırma (DK) algoritması olarak Şekil 3.7'de

verilmiştir. Burada, HRA algoritmasının doğrudan ya da içinde çalışan DK'yi kullanarak verdiği, genellikle doğru parçalarıyla birleştirilen yol bağlanabilirlik yolu (connectivity path), geçtiği noktalar ise geçiş noktaları (viapoint) olarak tanımlanmıştır.



Şekil 3.6 : Ölçev olarak a) Öklid mesafesinin ve b) Manhattan mesafesinin kullanıldığı 2 boyutlu uzayda Voronoi hücreleri (Url-2).

DALLARI_KIR(T_{son})	
1	$kırılan_dal=0$ olmadıkça
2	$kırılan_dal=0$
3	$k=1$ 'den K 'ye
4	eğer $q_k \neq q$ ve q_k bir uç nokta ise
5	$T_{ayrım_noktası_sil}(q_k)$
6	$T_{dal_kır}(q_k, q_{yakın})$
7	<i>Bağlanabilirlik Yolu</i> 'nu ver

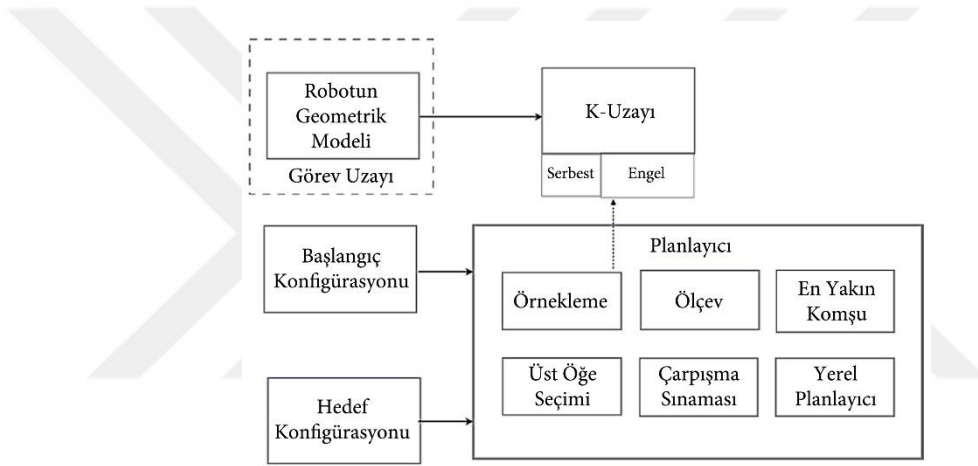
Şekil 3.7 : DK algoritması (Boyacıoğlu ve Ertuğrul, 2015).

Bağlanabilirlik yolu, ileriki bölümlerde de bahsedileceği üzere genellikle uygulanabilir olmaktan uzak olduğu için, planlanan yol bir yerel planlayıcı ile yörüngeye dönüştürülmelidir. Son aşamada ise, yörünge üstündeki yeterli sayıda nokta için çarpışma sınaması yapılmalıdır. Şekil 3.8'de, HRA algoritmasının bahsedilen tüm bileşenleri özetlenmiştir (Elbanhawı ve Simic, 2014). Bu şema, diğer örnekleme tabanlı planlama yöntemleri için de aynen kullanılabilir.

3.4 Çarpışma Sınaması

Bir robotik manipülatör için kendi kendine çarpma durumunun önüne, yukarıda da bahsedildiği üzere, henüz yörünge planlamanın örnekleme aşamasında eklem hareketlerini kısıtlayarak geçilebilir. Diğer yandan, küresel engeller için örnekleme

tabanlı planlayıcıda örneklerin $K_{serbest}$ 'te olup olmadığını sınamak yeterli değildir ve iki örnek arasında kurulacak bağlantıdaki noktaların da çarpışma sınaması yapılmalıdır. Sınama, çarpışma var ya da yok şeklinde çıktı veren bir Boolean fonksiyonu olarak düşünülebilir. Bu noktada dikkat edilmesi gereken bir husus, Şekil 3.9(a)'da görselleştirildiği gibi, iki nokta arasında çarpışma olma ihtimali en yüksek olan yerin orta nokta olduğu saptamasıdır (*ibid.*). Şekil 3.9(b)'de ise, daha kısa doğru parçaları üzerinde çarpışma ihtimalinin daha az olduğu gösterilmiştir. Bu gözlemlerden yararlanarak Şekil 3.10(a)'da görülen artımsal (incremental) çarpışma sınaması yerine, Şekil 3.10(b)'de yer alan orta noktalarda çarpışma sınamasının daha verimli olacağı söylenebilir. Yine aynı amaçla, sınamaya uzun parçalardan başlanmalıdır.

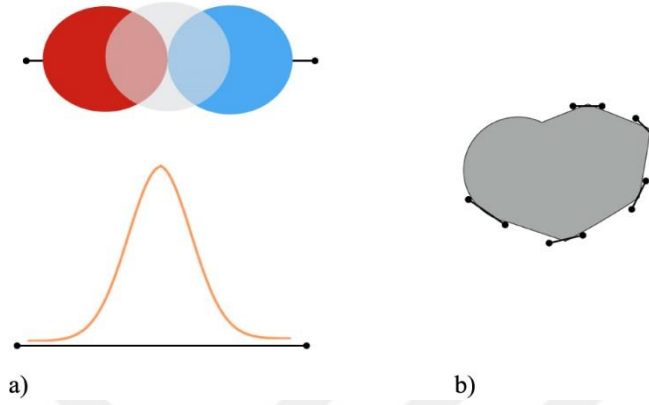


Şekil 3.8 : HRA ve diğer örneklem tabanlı algoritmaların alt fonksiyonları (Elbanhawi ve Simic, 2014).

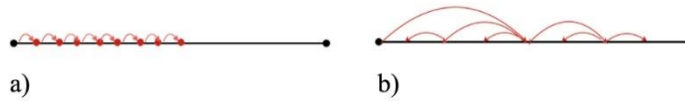
Çarpışma sınamasının bazı planlamalarda hesaplama maliyetinin %90'ına karşılık geldiği görülmüştür ve buna karşı, sınamayı gerekli oluncaya dek erteleyen algoritmalar önerilmiştir (*ibid.*). Bu yaklaşıma örnek olarak, HRA'da oluşturulan her dal için sınama yapılması yerine, sadece bağlanabilirlik yolunda çarpışma olup olmadığına bakılması verilebilir.

Noktasal olmayan kütleler için yol üzerindeki belli çözünürlükte sınamalar yapmak da yeterli olmayacaktır. Bunun için, kütleleri tamamen kaplayan küre veya çok yüzlü geometriler tanımlanır ve bunların engellerle çakışıp çakışmadığı düşük bir hesaplama maliyeti ile denetlenebilir. Şekil 3.11'de aynı cismin farklı geometrik yaklaşımlar ile çevrelenmesi görülmektedir. Bunlar sırasıyla küre, eksenlerle hizalı sınırlayan kutu, yönlendirilmiş sınırlayan kutu ve eksenlerle ayrık yönlenmiş çokyüzlüdür (Weller, 2013). Bir hacmin, Şekil 3.12'deki Stanford tavşanı örneğindeki gibi birden fazla

şekille ifade edilmesi, daha doğru sonuç vermeyi vaat etmekle beraber; bu yöntemde hesaplama süresi de daha uzun olacaktır. Steinbach ve arkadaşları da ARMAR insansı robot prototipi için kürelerle sınırlama yaklaşımını temel alan çarpma ve kendi kendine çarpma sınaması yapan bir yol önermiştir (2006).

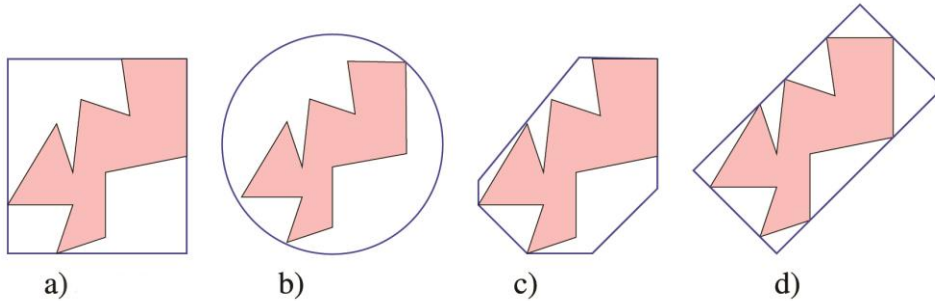


Şekil 3.9 : a) İki nokta arasında çarpışma olasılığı dağılımı ve b) doğru parçası uzunluğuna göre çarpışma durumu (Elbanhawi ve Simic, 2014).

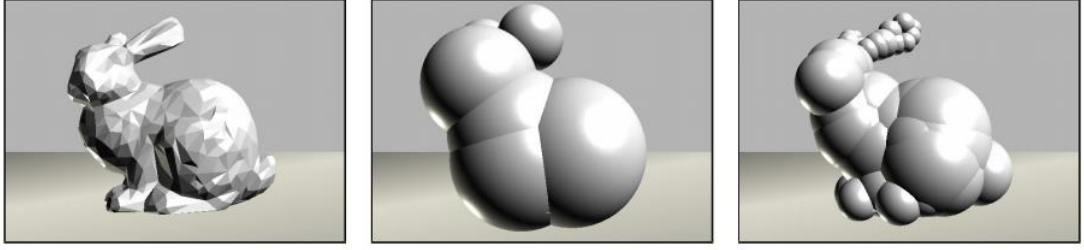


Şekil 3.10 : a) Artımsal çarpışma sınaması ve b) orta noktalarda çarpışma sınaması (Elbanhawi ve Simic, 2014).

Çarpışma sınaması ile ilgili son olarak, hazır kütüphanelerden söz edilebilir. Bunların başında, Kuzey Karolina Üniversitesi'nden GAMMA araştırma grubunun yayınladığı C++ dilinde yazılmış I-COLLIDE, RAPID, V-COLLIDE ve PQP programları gelmektedir (Url-3). Reggiani ve arkadaşlarının çalışmasında (2002), başlıca paket programlar deneysel olarak değerlendirilmiştir.



Şekil 3.11 : Sınırlayıcı hacimler (Weller, 2013).



Şekil 3.12 : Stanford tavşanı ve kürelerle sınırlanması (de Sousa Rocha ve Rodrigues, 2008)

3.5 Yerel Planlama

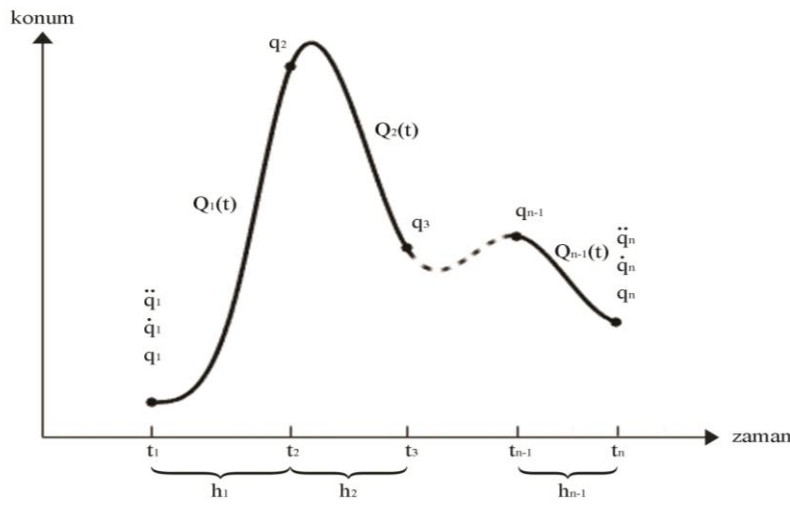
Geçiş noktaları, bir başka deyişle bağlanabilirlik yolu, nasıl elde edilmiş olursa olsun hareketin pürüzsüz olması için bir ileri işleme (post-processing) gereklidir. Aksi durum, bağlanabilirlik yolunu izleyen robotun çok yavaş hareket etmesi ya da hızlı hareket edecek bile olsa anlık durma ve kalkma devinimleriyle mekanik olarak boş yere zorlanması anlamına gelecektir.

Sadece konum değil, hız ve ivmede de sürekliliği beraberinde getirmesi gereken pürüzleri giderme işlemi için çeşitli yöntemler vardır. Bunlardan biri, geçiş noktaları arasında Bézier eğrileri uydurmaktır. Bağlanabilirlik yolunun tamamına karşılık gelen sürekli Bézier eğrileri, B-Spline eğrisini verir. Bu yaklaşımın örnekleri, gerek manipülatörler gerekse robotlar ile insansız araçlar için 30 yıla yakın süreden beri mevcuttur (Thompson ve Patel, 1987; Koyuncu ve İnalhan, 2008; Simba ve diğ., 2013). Yerel planlamada B-Spline eğrilerinin kullanılmasının avantajı, parça bazında değişikliğe yani noktalar arasında bir Bézier eğrisinin çıkartılıp bir başkasının yerleştirilmesine müsaade vermesidir. Öte yandan, eğrilerin oluşturulması için sezgisel olarak tespit edilen kurallar ile kontrol noktaları belirlenmesi gereklidir.

Diğer bir seçenek olarak konum kübik polinomlar ile ifade edilirse; hız karesel, ivme ise doğrusal fonksiyonlarla ifade edilebilecektir. Bu noktada sarsım sabit olur. Şayet geçiş noktalarının konum ve zaman bilgisi ile ilk ve son konumların hız ve ivme bilgileri mevcutsa, her ardışık geçiş noktası arasına birer kübik konum fonksiyonu tanımlanarak yörüngede süreklilik sağlanabilir. Bunun için, ardışık iki zaman aralığında tanımlı kübik fonksiyonların kesiştikleri geçiş noktasındaki konum, hız ve ivme değerlerinin aynı olması şarttır. Bu durumdan hareketle Lin ve arkadaşları, eklem yörüngelerinin kübik polinomlarla ifade edilmesi için gerekli formülasyonu ortaya koymuştur (1983).

3.5.1 Eklem yörüngelerinin kübik polinomlar ile ifade edilmesi

Yörünge planlaması yapılacak robotik manipülatörün bir eklemine, $[t_1, t_n]$ zaman aralığı için ele alalım. Bu eklem geçiş noktalarındaki konum ve bu geçiş noktalarındaki zaman değişkenleri, Şekil 3.13'te gösterildiği gibi, sırasıyla bilinen $[q_1, q_2, \dots, q_{n-1}, q_n]$ ve $[t_1, t_2, \dots, t_{n-1}, t_n]$ olsun. Burada n geçiş noktası sayısıdır. Ek olarak ilk ve son noktadaki hız ve ivme değerlerinin, kısaca sınır koşullarının, bilindiğini varsayalım. Zaman aralıkları, h_i 'ler, $i = 1, 2, \dots, n-2, n-1$ için 3.4'teki gibi halihazırda bilinen t_i değerleri ile hesaplanabilir. Öte yandan, sınır noktaları haricindeki geçiş noktalarındaki ivme değerleri, $[\ddot{q}_2, \dots, \ddot{q}_{n-1}]$, bilinmemektedir.



Şekil 3.13 : Değişken gösterimleri.

$$h_i = t_{i+1} - t_i \quad (3.4)$$

i ve $i + 1$ numaralı geçiş noktaları arasındaki doğrusal ivme fonksiyonu, $Q_i''(t)$, $i = 1, 2, \dots, n-1$ için 3.5'te ifade edilecek olursa, bu denklemin iki kez integrali alındığında aynı aralıktaki kübik konum fonksiyonu 3.6'daki gibi yazılabilir (*ibid.*).

$$Q_i''(t) = \frac{t_{i+1} - t}{h_i} Q_i''(t_i) + \frac{t - t_i}{h_i} Q_i''(t_{i+1}) \quad (3.5)$$

$$Q_i(t) = \frac{Q_i''(t_i)}{6h_i} (t_{i+1} - t)^3 + \frac{Q_i''(t_{i+1})}{h_i} (t - t_i)^3 \quad (3.6)$$

$$+ \left[\frac{q_i}{h_i} - \frac{h_i Q_i''(t_i)}{6} \right] (t_{i+1} - t)$$

Lin ve arkadaşları, bilinmeyen $n - 2$ adet ivmenin çözülebilmesi için, mevcut sınır koşulları altında gerekli serbestliği sağlayan fazladan 2 adet sabit olmayan geçiş noktası eklemeyi önermiş ve n 'yi, bu iki noktayı da dâhil edecek şekilde toplam geçiş noktası sayısı olarak yeniden tanımlamıştır. Fonksiyonların uç sınırdaki değerleri ile ardışık ivme fonksiyonlarının 3.7 numaralı denklemde de ifade edilen eşitliğinden elde edilen doğrusal denklem sistemi 3.8'de, A katsayılar matrisi ise 3.9'da verilmiştir. Denklemin sağ tarafının elde edilişi, detaylı olarak EK A'da yer almaktadır (*ibid.*).

$$Q_i''(t_{i+1}) = Q_i''(t_{i+1}) \quad (3.7)$$

$$A \begin{bmatrix} Q_2''(t_2) \\ Q_3''(t_3) \\ \vdots \\ Q_{n-1}''(t_{n-1}) \end{bmatrix} = [f(q, \dot{q}, \ddot{q}, h)]_{n-2 \times 1} \quad (3.8)$$

$$A = \begin{bmatrix} a_{11} & a_{12} & & & 0 \\ a_{21} & a_{22} & a_{23} & & \\ & & \vdots & a_{n-3,n-3} & a_{n-3,n-2} \\ 0 & & & a_{n-2,n-3} & a_{n-2,n-2} \end{bmatrix}$$

$$= \begin{bmatrix} 3h_1 + 2h_2 + \frac{h_1^2}{h_2} & h_2 & & & 0 \\ & & \vdots & 2(h_{n-3} + h_{n-2}) & h_{n-2} - \frac{h_{n-1}^2}{h_{n-2}} \\ & & & h_{n-2} & 3h_{n-1} + 2h_{n-2} + \frac{h_{n-1}^2}{h_{n-2}} \\ 0 & & & & \end{bmatrix} \quad (3.9)$$

Lin ve arkadaşları çalışmalarının devamında, problemin tek bir çözüm verdiğini yani A matrisinin tekil olmadığını ispatlamışlardır (*ibid.*).

3.5.2 Uygulanabilir çözüm dönüştürücü

Verilen zaman aralıkları ile geçiş noktaları bilgilerine göre uydurulan kübik konum eğrisi, geometrik kısıtlara uysa da hız, ivme ve sarsım kısıtları, kısaca kinematik kısıtları, aşıyor olabilir. Bu durumda, bulunan çözümün zamana yayılarak uygulanabilir hale getirilmesi önerilir. Uygulanabilir çözüm dönüştürücü (UÇD), N sayıda eklemi aynı anda göz önünde bulundurarak bunu gerçekleştirmekte kullanılır.

HK_j , IK_j ve SK_j sırasıyla j numaralı ekleme ait hız, ivme ve sarsım kısıtları olsun. Tüm eklemler için, bu kısıtlar dikkate alınarak bir λ , uygulanabilirlik ayar çarpanı (feasibility adjustment factor) 3.10-13'ten yararlanılarak hesaplanır. Ardından zaman aralıkları $[\lambda h_1, \lambda h_2, \dots, \lambda h_{n-1}]$ ile, ivmeler ise $[Q''_{j2}/\lambda^2, Q''_{j3}/\lambda^2, \dots, Q''_{j(n-1)}/\lambda^2]$ ile yer değiştirilir (*ibid.*). Burada yapılan uydurulan eğrinin ölçeklendirilmesinden ibarettir.

$$\lambda_1 = \max_j \left[\max_{\substack{t \in [t_i, t_{i+1}] \\ \forall i}} |Q'_{ji}(t)| / HK_j \right] \quad (3.10)$$

$$\lambda_2 = \max_j \left[\max_{\substack{t \in [t_i, t_{i+1}] \\ \forall i}} |Q''_{ji}(t)| / IK_j \right] \quad (3.11)$$

$$\lambda_3 = \max_j \left[\max_{\substack{t \in [t_i, t_{i+1}] \\ \forall i}} |Q'''_{ji}(t)| / SK_j \right] \quad (3.12)$$

$$\lambda = \max(1, \lambda_1, \lambda_2^{1/2}, \lambda_3^{1/3}) \quad (3.13)$$

Şu ana kadar manipulatörler, HRA ve kübik eğriler ile yörünge süreklileştirme özelinde yörünge planlama ile ilgili temel kavram ve yöntemlerden bahsetmiş bulunmaktayız. Bir sonraki bölümde, yörünge planlama probleminin ele alınacağı ITECH insansı robot kolu tanıtılacaktır.

4. ITECH İNSANSI ROBOT KOLU

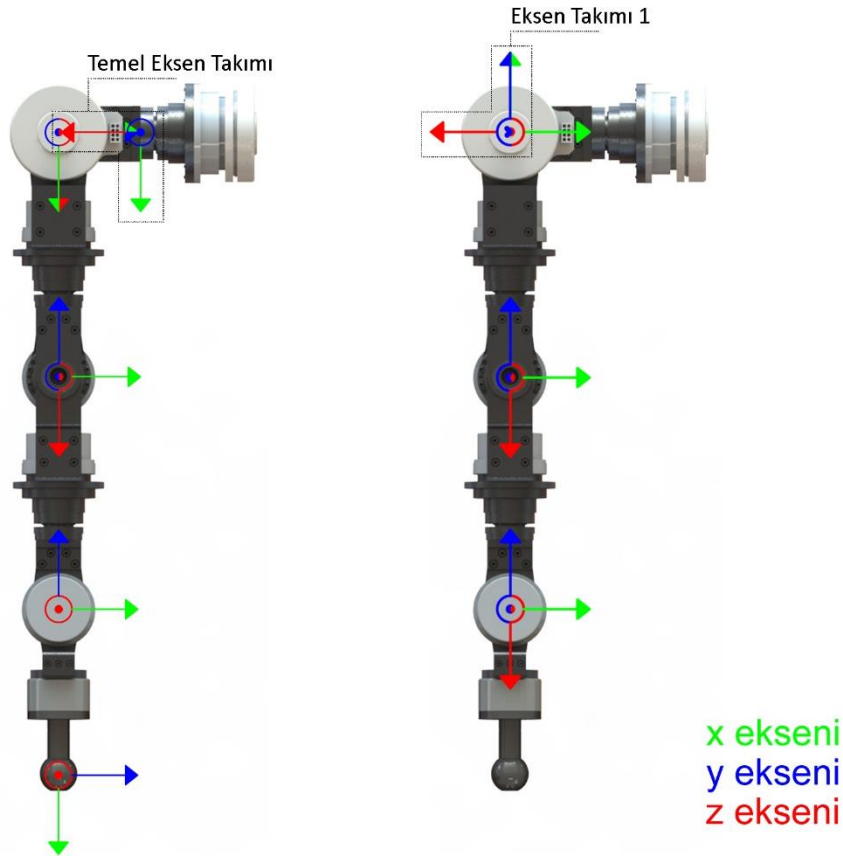
ITECH İnsansı Robotu'na ait kolun ön tasarımı Meşeli tarafından yapılmış olup bu çalışmada Güleç tarafından SolidWorks'te verilen son hâli kullanılmıştır (2014). Tasarıma ait bir görünüş Şekil 4.1'de yer almaktadır. D-H ve değiştirilmiş (modified) D-H parametreleri ise Şekil 4.2'de verilen eksen takımları ve temel konfigürasyona göre Çizelge 4.1'de tanımlanmıştır. Görüldüğü üzere robot kol 3'ü omuzda, 1'i dirsekte ve 2'si bilekte olmak üzere 6 adet döner ekleme sahiptir. Yani, Çizelge 2.2'de yer alan insansı robot kolu çalışmalarından farklı olarak kol, bilekte sapma hareketi kabiliyetine sahip değildir; bir başka deyişle bilek küresel bir eklem olarak değerlendirilemez.



Şekil 4.1 : ITECH İnsansı Robot Kolu (Güleç, 2014).

Robot kolun tüm eklemlerinde Maxon marka fırçasız doğru akım motorları kullanılmıştır. İlk 2 eklem için EC 90 flat modeli seçilirken, diğer 4 ekleme EC 60 flat tercih edilmiştir. Dişli olarak, EC 90 flat kullanılan eklemlerde 1:160 diğerlerinde 1:120 çevrim oranına sahip Harmonic Drive® mekanizmaları kullanılmıştır. Motor ve dişli mekanizmalarının özelliklerinin verildiği Çizelge 4.2'den de görüleceği üzere,

motorlar için verilen en yüksek sürekli tork değerleri, olması gerektiği gibi, dişli mekanizmaları için verilen en yüksek tork giriş değerlerinin altındadır (*ibid.*).



Şekil 4.2 : a) D-H ve b) değiştirilmiş D-H parametreleri için tanımlanan eksenler.

Çizelge 4.1 : D-H ve değiştirilmiş D-H parametreleri.

Eklem, j	D-H				Değiştirilmiş D-H			
	a_j [mm]	α_j [rad]	d_j [mm]	θ_j [rad]	α_j [rad]	d_j [mm]	θ_j [rad]	r_j [mm]
1	0	$-\pi/2$	75.5	θ_1^*	0	0	θ_1^*	0
2	0	$\pi/2$	0	$\theta_2^* + \pi/2$	$\pi/2$	0	$\theta_2^* - \pi/2$	0
3	0	$-\pi/2$	225	θ_3^*	$\pi/2$	0	θ_3^*	225
4	0	$\pi/2$	0	θ_4^*	$-\pi/2$	0	θ_4^*	0
5	0	$-\pi/2$	214	θ_5^*	$\pi/2$	0	θ_5^*	214
6	166,31	0	0	$\theta_6^* - \pi/2$	$-\pi/2$	0	$\theta_6^* - \pi/2$	0

Çizelge 4.2 : Motor ve dişli mekanizmalarına ait özellikler (Güleç, 2014).

Eklem, j	Motor			Harmonic Drive® Dişli Mekanizması		
	Model	En Yüksek Sürekli Tork [Nm]	Ağırlık [kg]	Model	En Yüksek Giriş Torku [Nm]	Ağırlık [kg]
1-2	EC 90 flat	0,444	0,6	CPL-20A-(1:160)	0,575	0,14
3-4	EC 60 flat	0,264	0,47	CPL-17A-(1:120)	0,450	0,1

Robot kolu montaj aşamasında olup hâlihazırda tasarlanmış bir tutucuya (gripper) sahip değildir. Uç eyleyici konumu hakkında bir fikir vermesi için geçici olarak bir topuz kullanılmaktadır.

4.1 Geometrik Kısıtlar ve Tekillikler

Robot koldan G-uzayında sanal bir doğru parçası üzerinde hareket etmesi istendiğinde, bu görevi aynı anda ya da peş peşe birden fazla eyleyiciyi harekete geçirmeden yapması mümkün değildir yani kolun holonomik olmadığı söylenebilir. Buna ek olarak, kendi kendine çarpma durumlarından ve insansı olmayan hareketlerden sakınma amacıyla eklem hareketleri Şekil 4.2a’da verilen referans konumlara göre Çizelge 4.3’te yer alan geometrik kısıtlar da robot için bağlayıcıdır. Bu sınırlar, gerek Çizelge 2.1’de yer alan günlük hareketler için insandan alınan verilerle gerekse Çizelge 2.2’de yer alan diğer insansı robot kol çalışmalarıyla karşılaştırılacak olursa, ITECH’in literatür ile paralel bir noktada olduğu görülecektir.

Çizelge 4.3 : ITECH İnsansı Robot Kolu’na ait geometrik kısıtlar.

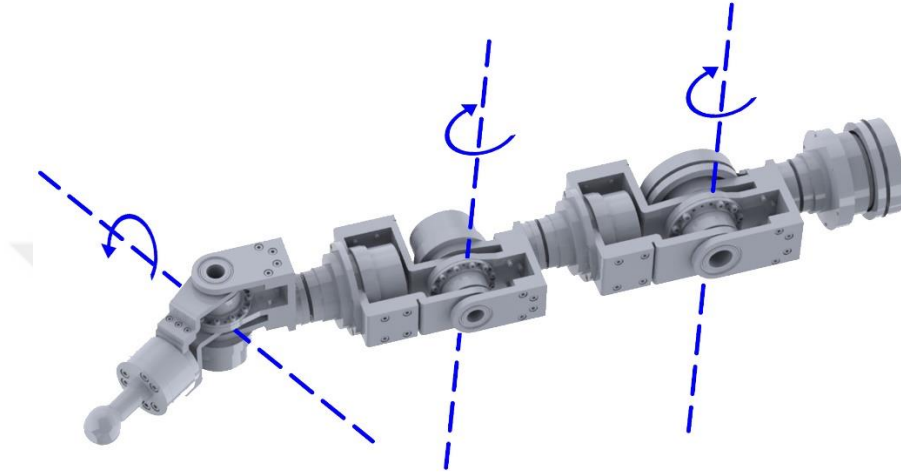
	alt limit [°]	üst limit [°]
Omuz (yunuslama)	-45	225
Omuz (sapma)	-60	120
Dirsek (yalpa)	-120	120
Dirsek (sapma)	0	90
Bilek (yalpa)	-90	90
Bilek (yunuslama)	-75	75

Robot kola ait tekil konfigürasyonlar, 2. Bölüm’de de açıklandığı üzere Jakobiyen matrisinin determinanı ile doğrudan ilişkilidir. 6 serbestlik derecesine sahip robot kolun Jakobiyen matrisi 6×6 boyutlarında olacaktır. Bu matris simgesel olarak hesaplanıp determinanı alınacak olursa sadeleştirilmiş sonuç aşağıdaki gibi olacaktır. Burada A uzuv uzunluklarına bağlı bir sabittir.

$$\det(J) = A \cos \theta_2 \cos \theta_5 (\cos^2 \theta_4 - 1)$$

Bu ifade 0’a eşitlenip çözüldüğünde θ_2 ile θ_5 ’in $\pm \pi/2$ ve θ_4 ’ün 0 ile π değerlerinin tekilliğe neden olduğu görülür. Bunlardan θ_2 ’nin $-\pi/2$ ve θ_4 ’ün π ’ye eşit olduğu konfigürasyonların zaten geometrik kısıtlar dâhilinde olmadığı görülecektir. Geriye kalan 4 tekillikten 3’ü Şekil 4.3 üzerinde gösterilmiştir. Bunlar temel eksen takımından uca doğru sırasıyla omuz, dirsek ve bilek tekilliği olarak adlandırılır. Görüldüğü üzere tamamen açık konumda kol en fazla tekillik durumuyla karşı karşıyadır.

θ_4 ve θ_5 için tekillik durumlarının aynı zamanda geometrik sınırlarda olduğu göz önünde bulundurarak kısıtlar yeniden değerlendirilecek olursa; alt/üst limitler θ_4 için $5^\circ/90^\circ$, θ_5 içinse $-85^\circ/85^\circ$ olarak güncellenerek hareket yeteneğinden büyük bir ödün vermeden tekilliklerin çoğundan sakınmış olunacaktır. Bu yaklaşım, bu çalışmaya has olmayıp örnek olarak, ASIMO İnsansı Robotu'nun yürüyüşü esnasında ayaklarının sürekli bükülü konumda tutulmasında da görülebilir.

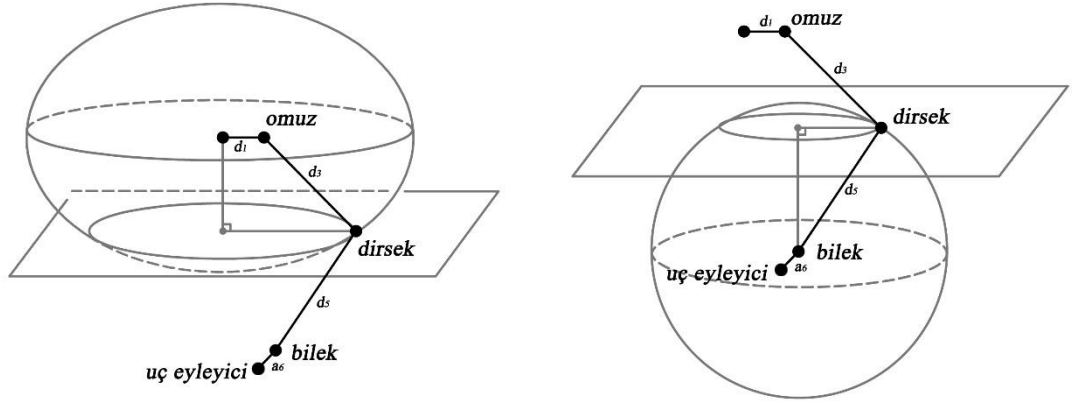


Şekil 4.3 : ITECH İnsansı Robot Kolu'na ait bazı tekillikler.

4.2 Önerilen Ters Geometrik Çözüm

Güleç ve Ertuğrul (2014), Asfour ve Dillmann'ın ARMAR İnsansı Robotu'na ait 7 serbestlik dereceli kol için önerdiği analitik yaklaşımını (2003) ITECH İnsansı Robot Kolu için çağırmıştır. Bu yaklaşım temel olarak, Şekil 4.4'te görselleştirildiği üzere, omuz ve bilek konumları tespit edildikten sonra dirsek konumunun omuz ve bilek merkezli sırasıyla bir elipsoit ve d_5 yarıçaplı bir kürenin eliptik kesitleri üzerinde yer alması gereğinden yararlanarak eklem açıları için bir çözüm sunar.

Yöntemin ilk olarak önerildiği çalışmada (*ibid.*) insansı harekete ait yörüngeler, G-uzayında referans olarak verilip belli bir sıklıkta K-uzayına aktarılırken; Güleç ve Ertuğrul (2014), G-uzayında verilen başlangıç ve hedef noktalarını K-uzayına aktardıktan sonra tüm eklem değişkenleri için kübik ya da yamuk (trapezoid) hız profilleri ile yörünge planlamıştır. Bu ikinci yaklaşımda robot kol, insansı olmayan hareketler sergileyebileceği gibi G-uzayında engeller bulunması durumuna karşı ileri geometrik modeli kullanarak, bu uzayda çarpışma sınamaları yapma gereği de duyar.



Şekil 4.4 : Asfour ve Dillmann'ın önerdiği analitik ters geometri yaklaşımı (2003).



5. ÖNERİLEN YÖRÜNGE PLANLAMA YÖNTEMİ

Bu bölümde, ITECH İnsansı Robot Kolu özelinde manipülatörler için, HRA algoritması tarafından verilmiş geçiş noktaları boyunca toplam hareket zamanı amaç fonksiyonuna sahip bir optimum yerel planlama yöntemi önerilmiştir. Optimizasyon için Lin ve arkadaşlarının, HRA algoritmasının ortaya çıkışından 15 yıl önce ortaya attıkları Nelder-Mead yöntemine dayanan yaklaşımı (1983) kullanılmıştır.

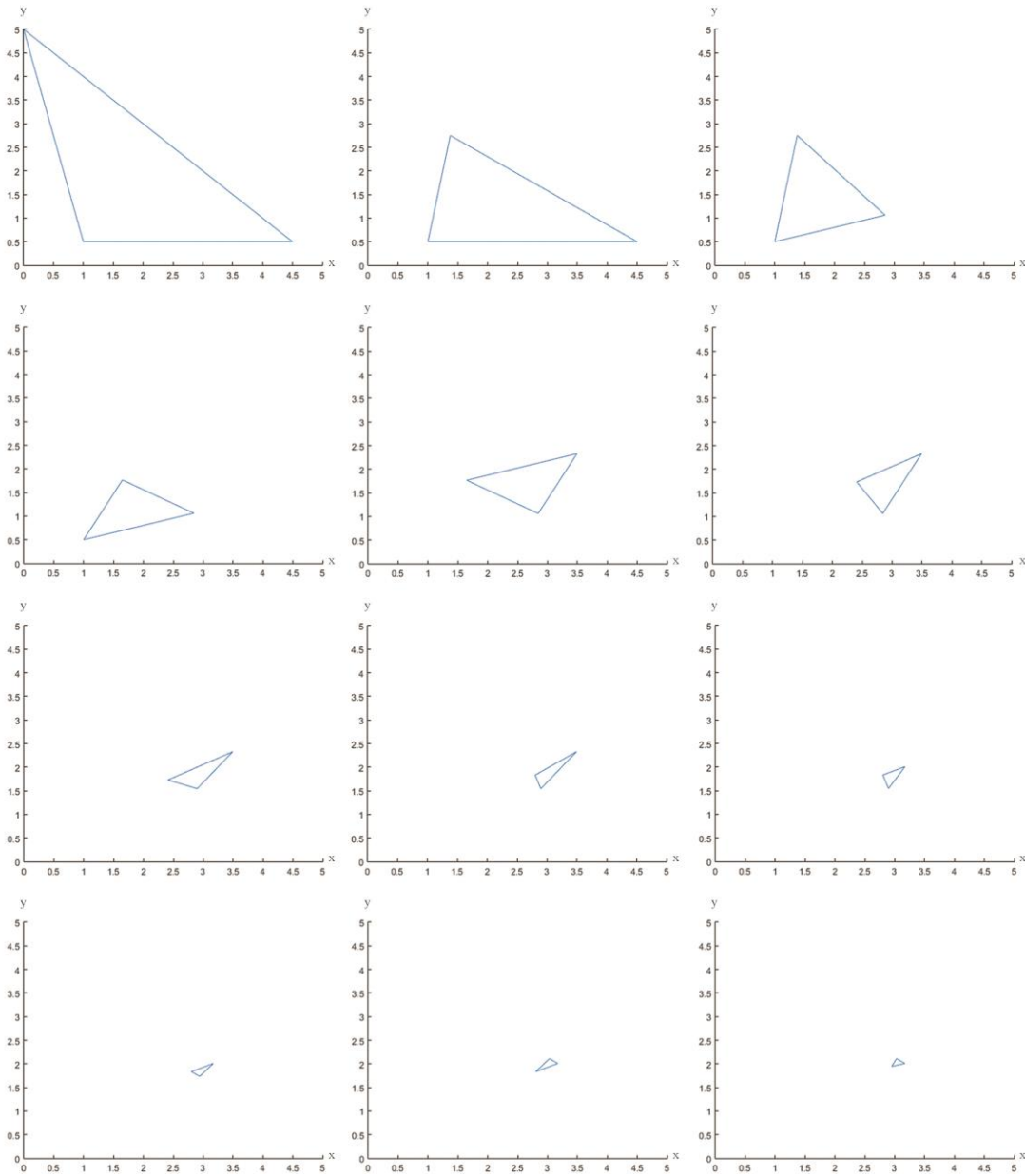
5.1 Nelder-Mead Optimizasyon Yöntemi

Nelder-Mead (1965), doğrudan aramaya dayalı bir optimizasyon yöntemi olup temel algoritması Şekil 5.1’de verilmiştir. Yöntemde ilk olarak n adet değişkene bağlı bir maliyet fonksiyonu için, $n + 1$ adet $\mathcal{R}^n$ ’de tanımlı olası çözüm noktası (vertex) önerilir. Ardından bu noktalar maliyetlerine göre sıralanarak en maliyetli olan X_b , en az maliyeti olan ise X_k olarak tespit edilir (*Satır 1-2*). Ardından X_b ’nin yerine belli kurallara göre türetilen maliyeti daha az noktalar yerleştirilir ve süreç döngüsel olarak, yeni X_b ’lerin maliyeti X_k ’ye neredeyse eşit oluncaya dek devam eder.

```
NELDER-MEAD( $X_1, \dots, X_{m+1}$ )
1    $X_i$ ’leri sırala
2    $maliyet(X_1) \leq \dots \leq X_m \leq maliyet(X_b)$ 
3    $X_b$ ’yi hariç tutarak kütle merkezini,  $X_o$ , hesapla
4   Yansıt;  $X_y = 2X_o - X_b$ 
5   eğer  $maliyet(X_y) < maliyet(X_s)$  ise
6       Genişlet;  $X_g = 2X_y - X_o$ 
7       eğer  $maliyet(X_y) \leq maliyet(X_g)$  ise
8            $X_b = X_y$ 
9       aksi halde
10           $X_b = X_g$ 
11  aksi halde eğer  $maliyet(X_k) \leq maliyet(X_y) < maliyet(X_m)$ 
12       $X_b = X_y$ 
13  aksi halde
14      Küçült;  $X_c = 0.5(X_b - X_o)$ 
15      eğer  $maliyet(X_c) < maliyet(X_{m+1})$  ise
16           $X_b = X_c$ 
17      aksi halde
18          İndirge;  $X_i \leftarrow 0.5(X_k - X_i)$  for  $i = 2, \dots, m+1$ 
19  Adım 1’e dön
```

Şekil 5.1 : Nelder-Mead yöntemine ait algoritma.

Şekil 5.2’de, örnek olarak iki değişkenli (x ve y) bir fonksiyonun ($f(x, y) = x^2 - 4x + y^2 - y - xy$) en düşük değerinin Nelder-Mead yöntemi ile bulunuşu aşama aşama verilmiştir. Çizelge 5.1’de bu örneğe ait noktalar ve bunların maliyetleri yer almaktadır. Görüldüğü üzere, değişken sayısından bir fazla sayıda çözüm noktası önerilmiş ve bunlar bir üçgen oluşturacak şekilde birleştirilmiştir. Algoritma genişleme, yansıtma, küçültme ve indirgeme işlemleri ile yeni noktalar türettikçe üçgen, aranan nokta civarında küçülecektir. Daha fazla değişkene bağlı maliyet fonksiyonları için noktaların bulunacağı uzay büyüyecek olsa da mantık aynı olacak; noktalardan oluşacak çokyüzlünün optimum bir noktaya odaklanması beklenecektir.



Şekil 5.2 : $f(x, y)$ fonksiyonunun minimum değerinin Nelder-Mead yöntemi ile bulunuşu.

Çizelge 5.1 : Şekil 5.2’de yer alan 12 kareye ait nokta konum ve maliyetleri.

kare	X_k		X_2		X_b	
	(x, y)	maliyet	(x, y)	maliyet	(x, y)	maliyet
1	(1, 0.5)	-3,7500	(4.5, 0.5)	-0,2500	(0, 5)	20,0000
2	(1, 0.5)	-3,7500	(1.3750, 2.7500)	-2,5781	(4.5, 0.5)	-0,2500
3	(2.8438, 1.0625)	-6,2432	(1, 0.5)	-3,7500	(1.3750, 2.7500)	-2,5781
4	(2.8438, 1.0625)	-6,2432	(1.6484, 1.7656)	-5,4351	(1, 0.5)	-3,7500
5	(3.4922, 2.3281)	-6,8116	(2.8438, 1.0625)	-6,2432	(1.6484, 1.7656)	-5,4351
6	(3.4922, 2.3281)	-6,8116	(2.4082, 1.7305)	-6,7366	(2.8438, 1.0625)	-6,2432
7	(2.8970, 1.5459)	-6,8300	(3.4922, 2.3281)	-6,8116	(2.4082, 1.7305)	-6,7366
8	(2.8970, 1.8337)	-6,9659	(2.8970, 1.5459)	-6,8300	(3.4922, 2.3281)	-6,8116
9	(3.1707, 2.0090)	-6,9723	(2.8970, 1.8337)	-6,9659	(2.8970, 1.5459)	-6,8300
10	(3.1707, 2.0090)	-6,9723	(2.8970, 1.8337)	-6,9659	(2.9415, 1.7336)	-6,9412
11	(3.0306, 2,1091)	-6,9905	(3.1707, 2.0090)	-6,9723	(2.8970, 1.8337)	-6,9659
12	(2.9510, 1.9464)	-6,9974	(3.0306, 2,1091)	-6,9905	(3.1707, 2.0090)	-6,9723

5.2 Kübik Polinom Eklem Yörüngelerinin Optimizasyonu

Lin ve arkadaşları, eklem yörüngeleri için kübik polinomlar uydurmayı formüle ettikten sonra, Nelder-Mead algoritmasında maliyet fonksiyonunu, 5.1 numaralı denklemde görüldüğü gibi zaman aralıkları toplamı yani başlangıç noktasından hedefe ulaşmak için geçen toplam zaman olarak tanımlayarak bu değerin optimizasyonunu ortaya koymuştur (2003).

$$maliyet = \sum_{i=1}^{n-1} h_i \quad (5.1)$$

n adet ardışık geçiş noktası arasında $n - 1$ adet zaman farkı olduğuna göre, Nelder-Mead yöntemini çağırmak için öncelikle, n adet olası çözüm ortaya konmalıdır. Bu ilk çözüm noktaları, 5.2’de tanımlanan zaman aralıkları vektörü alt sınırı, X' ’den türetilir. İlk olası noktayı ifade eden X_1^0 vektörü, X' ’nin UÇD yardımı ile uygulanabilir değerlere getirilmiş hâlidir. Diğer olası noktalar, 5.3 numaralı denklemde elde edilen noktaların yine UÇD ile uygulanabilir duruma getirilmiş versiyonlarıdır (*ibid.*).

$$X' = [h'_1, h'_2, \dots, h'_{n-1}] = [\max_j \frac{|q_{j2} - q_{j1}|}{HK_j}, \dots, \frac{|q_{jn} - q_{j(n-1)}|}{HK_j}] \quad (5.2)$$

$$X'_i = X_1^0 + d_i \quad (5.3)$$

d_i bir mesafe vektörü olup 5.4-7 yardımıyla bulunabilir. Burada D , ilk çokyüzlünün boyutuna etki edebilecek olan bir değişkendir. Bu değerin çok küçük olması durumunda, çözüm yerel bir değere sıkışıp kalabilir (*ibid.*).

$$d_{i,j} = \begin{cases} \delta_1, & i = j + 1 \\ \delta_2, & i \neq j + 1 \end{cases} \quad (5.4)$$

$$\delta_1 = \frac{D}{\sqrt{2}(n-1)}(\sqrt{n} + n - 2) \quad (5.5)$$

$$\delta_2 = \frac{D}{\sqrt{2}(n-1)}(\sqrt{n} - 1) \quad (5.6)$$

$$D = 10 \min \left\{ \frac{0.2}{n} \sum_{i=1}^{n-1} (h_i^1 - h_i'), (h_1^1 - h_1'), \dots, (h_{n-1}^1 - h_{n-1}') \right\} \quad (5.7)$$

5.3 HRA Sonuçlarının Süreklileştirilmesi için Bir Yöntem

Yukarıda bahsedildiği üzere, örnekleme tabanlı klasik HRA algoritması bir başlangıç noktası ile bir hedef noktayı bağlanabilirlik yoluyla hesaplama açısından verimli bir şekilde birleştirirken bir başka çalışmada (*ibid.*), manipülatörler için verilen bir yolda kinematik kısıtlar ile ilk ve son noktadaki hız ve ivme değerleri yani sınır koşullarını dikkate alarak bir süreklileştirme yaklaşımı önerilmiştir. Şekil 5.3'te de görüleceği gibi, gerek konfigürasyon gerekse görev uzayında sürekli bir yörünge elde etmek için bu iki yöntemin bir araya getirilişi ortaya konmuştur.

HRA_PÜRÜZSÜZLEŞTİRME($q_{ilk}, A_{hedef}, HK, \dot{IK}, SK$)	
1	HRA_OLUŞTUR(q_{ilk}, A_{hedef})
2	$X_1, \dots, X_n = f(\text{Bağlanabilirlik Yolu}, HK, \dot{IK}, SK)$
3	NELDER-MEAD($X_1, \dots, X_n$); $HK, \dot{IK}, SK$ 'yi dikkate alarak
4	Yörünge'yi ver

Şekil 5.3 : HRA tarafından verilen yolun optimum zaman için pürüzsüzleştirilmesi.

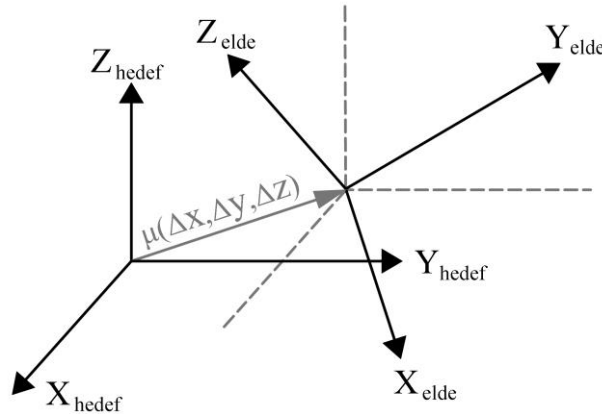
Yerel ve küresel tüm kısıtlar ile başlangıç konfigürasyonu (q_{ilk}) ve G-uzayındaki hedefe ait bilgilerin önceden tanımlanmış olduğu bir durumu ele alalım. Hedeflenen konum (x_{hedef}) ve yönelim (R_{hedef}), A_{hedef} temel eksen takımına göre tanımlı dönüşüm matrisinde 5.8'deki gibi ifade edilmiş olsun. Önerilen yöntemde ilk olarak K-uzayında geometrik kısıtlar dâhilinde örnekleme yapılır. q_{ilk} 'ten başlayarak

genişleyen ağacın ileri geometrik model ile x_{hedef} 'e varıp varmadığı kontrol edilir. Bu yaklaşımla, ters geometrik modele hiçbir şekilde ihtiyaç duyulmaması sağlanmış olacaktır.

$$A_{hedef} = \begin{bmatrix} R_{hedef} & x_{hedef} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

Diğer yandan, kullanılan yöntem ile x_{hedef} 'in karşılık geldiği eklem açıları tam olarak bilinemeyeceği için bir hedef bölge tanımlanması gereklidir. Bölgenin belirlenmesi için bir μ uzaklık toleransı tanımlanabilir. Bu tolerans, 5.9'da da verildiği gibi Öklid mesafesine göre tanımlanacak olursa hedef bölge μ yarıçaplı bir küreye karşılık gelecektir. Buradaki Δx , Δy ve Δz ; Şekil 5.4'te de gösterildiği üzere sırasıyla F_{hedef} eksen takımına ait x, y ve z eksenlerinde hedefe olan izdüşüm uzaklıklarıdır.

$$\mu = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (5.9)$$



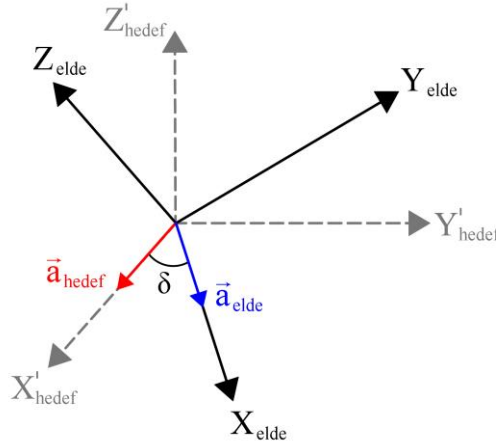
Şekil 5.4 : Hedef duruştan konumsal sapma.

Genişleyen ağaç hedef bölgeye girdiğinde uç eyleyici yöneliminin R_{hedef} 'in tolerans sınırları içinde olup olmadığına bakılmalıdır. Yönelimlere ait tolerans, δ , hedef yönelim ile elde edilen yönelimin arasındaki açı için tanımlanacaktır. Bunun için her iki yönelimde uç eyleyicinin normali doğrultusunda olan x eksenlerinin oluşturacağı düzlem kullanılır. Normallerin tanımlanması için, dönme matrislerinin ilk sütunları ele alınabilir. Diğer sütunlar gibi, bu sütunlar da birer birim vektörden meydana gelmektedir.

Hedef ve elde edilen birim uzunluktaki normal vektörler, Şekil 5.5'te gösterildiği üzere sırasıyla $\vec{a}_{hedef}$ ve $\vec{a}_{elde}$ olarak gösterilsin. Bu durumda, iki vektör arasındaki açı kosinüs teoremi (5.10) yardımıyla ya da nokta çarpımın tanımından (5.11) rahatlıkla bulunabilir. Buradaki $\vec{a}_{fark}$ iki normal vektör arasındaki fark vektörüdür.

$$|\vec{a}_{fark}|^2 = |\vec{a}_{hedef}|^2 + |\vec{a}_{elde}|^2 - 2|\vec{a}_{hedef}||\vec{a}_{elde}| \cos \delta \quad (5.10)$$

$$\vec{a}_{hedef} \cdot \vec{a}_{elde} = |\vec{a}_{hedef}||\vec{a}_{elde}| \cos \delta \quad (5.11)$$



Şekil 5.5 : Hedef duruştan açısal sapma.

5.11 ele alınacak olursa, normal vektörlerin aynı zamanda birim vektör oldukları yani uzunluklarının 1 birim olduğu göz önünde bulundurularak iki vektör arasındaki açı değeri, 5.12'de yer alan denklem yardımıyla hesaplanabilir.

$$\delta = \cos^{-1}(\vec{a}_{hedef} \cdot \vec{a}_{elde}) \quad (5.12)$$

Ağaç genişlerken en yakın komşuların bulunmasında da, geleneksel olan K-uzayındaki Öklid mesafesi kullanılmış olup herhangi bir yaklaşık en yakın komşu algoritmasına başvurulmadan rastgele üretilen noktanın ağaçta önceden bulunan ayırım noktalarının her birine olan uzaklığı hesaplanıp en kısa olan tespit edilmiştir.

Ağacın genişlemesi tamamlandıktan sonra, yaklaşık olarak hedef duruşta yer alan son noktaya hangi noktalardan geçilerek gelindiği kayıtlı bilgisi kullanılarak bağlanabilirlik yolu çizilir.

Eğri uydurma için Lin ve arkadaşlarının önerdiği ve yukarıda verilen formülasyon aynen çağrılmıştır. Zaman aralıkları toplamının maliyet olarak ele alındığı bu

yaklaşımında, h_i zaman aralığı değerlerinin doğal olarak sıfırdan büyük olmasına dikkat edilmelidir. Hatta sıfıra çok yakın pozitif değerlerin dahi elenmesi önerilir (*ibid.*).

Son olarak uç eyleyicinin G-uzayında izlediği yörünge için çarpışma sınaması, orta noktalar kullanılarak yapılmıştır. Burada artımsal yaklaşımda da tanımlanması gerekli ara noktalar arası mesafenin (Δ) tanımlanması gereklidir. Robotun hacminden kaynaklanacak çarpışma sınaması için uzuvlar etrafında 4 adet silindir ile ifade edilip basit geometrik ilişkiler kullanılmıştır.

5.4 Benzetim Senaryosu ve Sonuçlar

HRA tarafından verilen bağlanabilirlik yolunun kübik eğriler ile zaman optimizasyonu için önerilen yaklaşıma ait kod MATLAB®’de yazılmıştır. Robot kolun ilk ve hedef konfigürasyonlarda hareketsiz olması yani bu noktalarda hız, ivme ve sarsım değerlerinin sıfıra eşit olacak şekilde tanımlanması istenmiştir.

Yukarıda bahsedildiği üzere, ilk duruşa ait konfigürasyonun önceden bilinmesi istenmiştir. Bunun için bir seçenek olarak her bir eklemde mutlak enkoderler kullanılabilir ya da- robot kol harekete geçirilmeden evvel bilinen bir konfigürasyona getirilir. Başlangıç konfigürasyonu $q_{ilk} = [q_{1i}, q_{2i}, q_{3i}, q_{4i}, q_{5i}, q_{6i}] = [0, 0, 0, 0, 0, 0]$ olarak derece cinsinden tanımlanmıştır. Çalışmanın geri kalanında verilecek tüm değerler gibi bu değerler de, sonsuz seçenek arasından bir örnek olarak verilmiştir ve önerilen yöntem, olası tüm seçeneklere uyarlanabilir.

Çalışma kapsamında algortimanın ilk olarak çevrimdışı çalışması amaçlanmıştır. Çevrede yer alan engellerin konumununsa önceden bilindiği kabul edilmiştir. Benzetim senaryosu için, 75 mm çapında 4 adet hareketsiz engel G-uzayında tanımlanmıştır. Bu engellerin robot kol ile birlikte konumlandırılmasının yandan (XY düzleminde) görünüşü Şekil 5.6’da, küresel engellerin geometrik merkezlerinin koordinatları Çizelge 5.2’de verilmiştir.

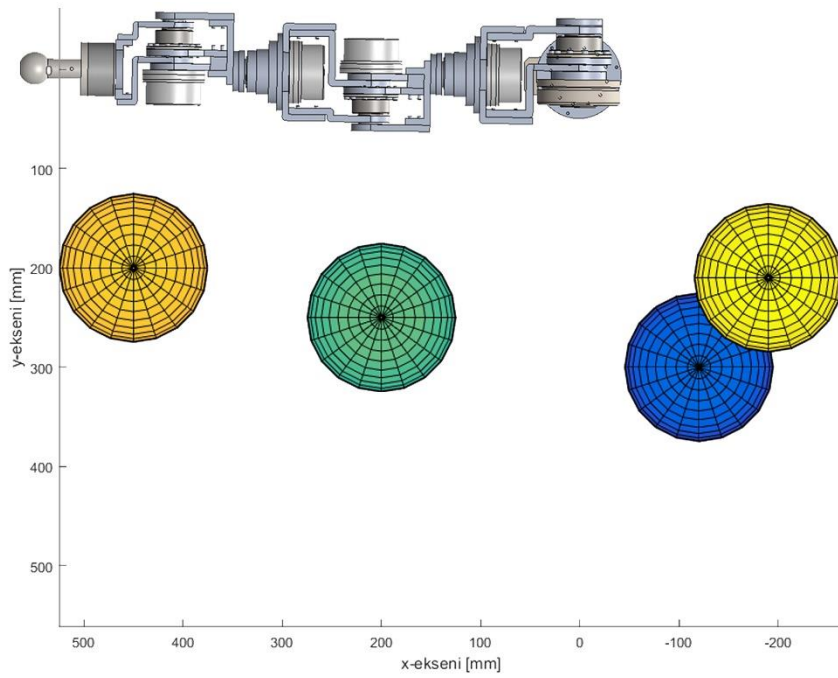
Çizelge 5.2 : G-uzayındaki küresel engellerin merkezlerinin koordinatları.

Engel	x [mm]	y [mm]	z [mm]
1	200	250	50
2	450	200	300
3	-190	210	400
4	-120	300	-400

Hedef konum ve yönelimi ifade eden A_{hedef} matrisi

$$A_{hedef} = \begin{bmatrix} 0.7071 & \vec{e}_2 & \vec{e}_3 & 200 \\ 0 & & & 300 \\ -0.7071 & & & -50 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

olarak belirlenmiştir. Bu matriste yer alan $\vec{e}_2$ ve $\vec{e}_3$ birim vektörler olup mevcut tasarımda uç eyleyici kendisine ait x eksenine göre simetrik olduğu için bu vektörlerin elemanlarına ait değerlerin bu durumda bir önemi yoktur.



Şekil 5.6 : Başlangıç durumunda robot kol ile engellerin konumlandırılması.

HRA algoritmasının çalıştırılmasından önce, son olarak, robot kola ait her bir eklem için hız, ivme ve sarsım sınır değerleri verilmelidir. Yukarıda bahsedildiği üzere bu; bağlanabilirlik yolu için değil, yörüngenin süreksizleştirilmesi için gereklidir. Sınır değerler, Çizelge 5.3'te tanımlanmıştır.

Çizelge 5.3 : Robot kol için tanımlanan hız, ivme ve sarsım sınır değerleri.

eklem, j	1	2	3	4	5	6
hız ($^{\circ}/s$)	97.125	97.125	187	187	187	187
ivme ($^{\circ}/s^2$)	920.62	920.62	2239.52	2239.52	2239.52	2239.52
sarsım ($^{\circ}/s^3$)	920.62	920.62	2239.52	2239.52	2239.52	2239.52

Bağlanabilirlik yolu elde edilene kadar sadece ε değişkeninin tanımlanması yeterli olacaktır. Bu değer, planlaması yapılacak sisteme bağlı olarak seçilir. Çok küçük olması halinde hedef konfigürasyona yaklaşması uzun sürecektir. Öte yandan, çok büyük olduğu takdirde, karışık çevrelerde uzun ve düz boşluklar kolaylıkla bulunamayacağı için olası bir çözümün bulunması yine gecikecektir. Planlama sırasında ε değeri, $\pi/2$ radyan olarak alınmıştır. Ayrıca hedef konuma olan uzaklık toleransı, μ , 45 mm ve hedef yönelimden en fazla açısal sapma, δ , $\pi/4$ radyan olarak tanımlanmıştır. Son olarak bağlanabilir yolunun çarpışma sınaması için en fazla 10 cm uzunluğunda doğru parçalarının ele alınması ve orta noktada sınama yaklaşımının kullanılması tercih edilmiştir.

Tüm hesaplamalar 12 GB RAM'e sahip 2.4 GHz hızında çift çekirdek i7 işlemcili bir dizüstü bilgisayarda yapılmıştır. Program 101 sefer çalıştırılmış, 68 defasında sınır değer olan 5 dakikadan kısa bir hesaplama süresinde bir çözüm elde edilmiş; ancak elde edilen her olası çözüm birbirinden farklı olmuştur. Bu çözümlerden bir tanesi Çizelge 5.4'te verilmiştir. Bu çizelgede başlangıç ve son konfigürasyonlar da dâhil olmak üzere bağlanabilirlik yoluna ait geçiş noktalarındaki eklem konfigürasyonları görülmektedir.

Çizelge 5.4 : HRA tarafından verilen geçiş noktalarındaki eklem konfigürasyonları.

eklem, j	1 [°]	2 [°]	3 [°]	4 [°]	5 [°]	6 [°]
Geçiş Noktası 1	0	0	0	0	0	0
Geçiş Noktası 2	17.61	25.21	-14.54	48.48	-65.64	16.84
Geçiş Noktası 3	44.91	-2.02	-36.91	46.89	-41.98	44.38
Geçiş Noktası 4	71.72	-29.93	-21.99	19.45	-30.50	56.12
Geçiş Noktası 5	81.82	-2.56	-7.62	42.24	-38.73	73.43

Örnekte, zaman aralıkları vektörü 5 elemanlı olup yörüngenin sürekli hale getirilmesi için başlangıçta iki adet ilave geçiş noktası, 1. ve 2. ile 4. ve 5. geçiş noktaları arasında peşpeşe noktaların aritmetik ortalaması olacak şekilde önerilir. Zaman aralıkları vektörü, X' , 5.2'den

$$X' = [0.1755 \quad 0.1755 \quad 0.2811 \quad 0.2873 \quad 0.1409 \quad 0.1409]$$

şeklinde hesaplanır. Bu vektörün UÇD ile uygulanabilirleştirilmiş hali ise,

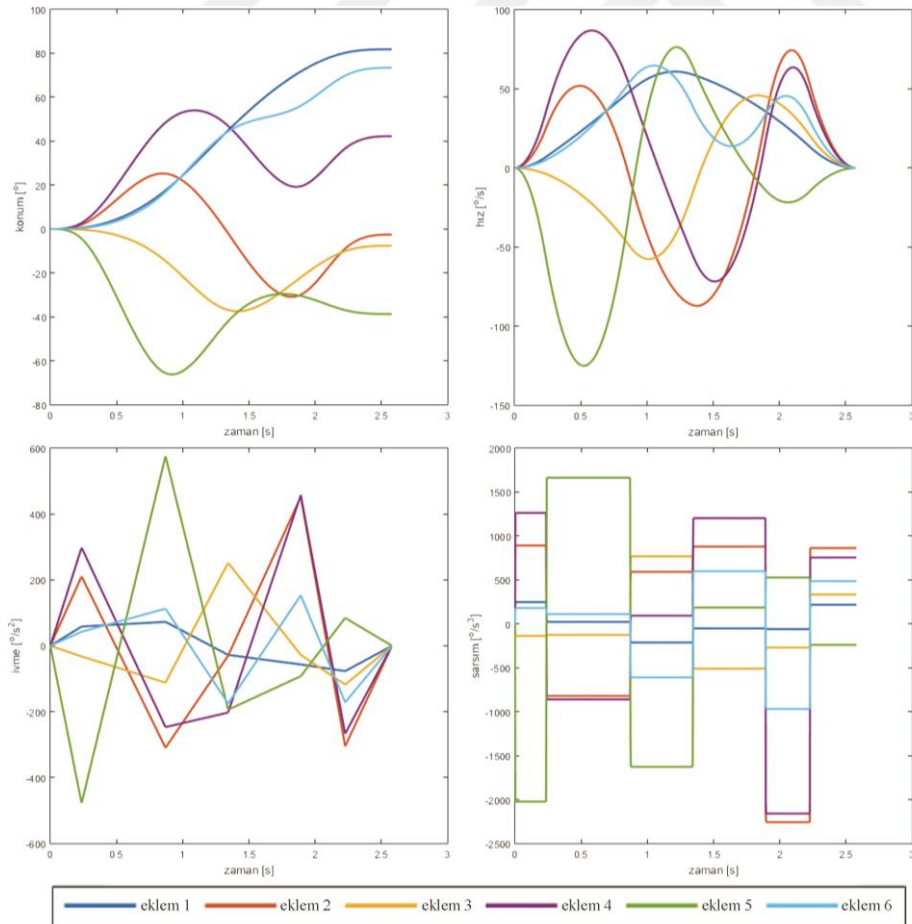
$$X_1^0 = [0.4351 \quad 0.4351 \quad 0.6968 \quad 0.7121 \quad 0.3492 \quad 0.3492]$$

idir. Bundan türetilecek diğer başlangıç vektörleri, X_i^0 'lar, 5.3'te verildiği gibi hesaplanır. Burada D , δ_1 ve δ_2 sırasıyla 0.5075, 0.4573 ve 0.0984 olarak 5.4-7 numaralı denklemlerden elde edilmiştir.

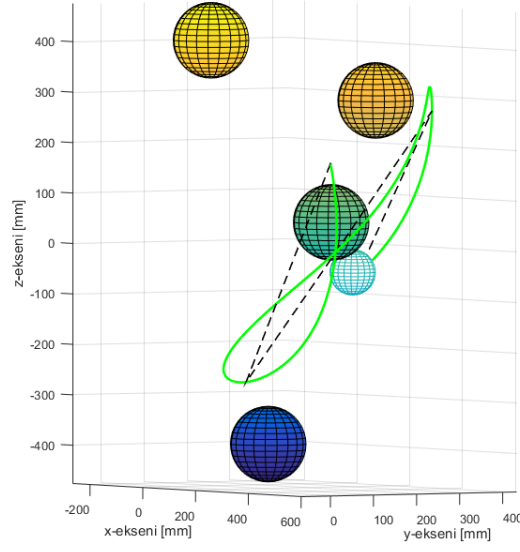
Süreklileştirme işleminden sonra elde edilen optimum zaman aralıkları vektörü, X_{opt} , aşağıdaki gibi hesaplanır. Görüldüğü üzere, hareketin gerçekleşmesi için planlanan toplam zaman 2.5776 saniyedir.

$$X_{opt} = [0.2307 \quad 0.6271 \quad 0.4864 \quad 0.5493 \quad 0.3367 \quad 0.3474]$$

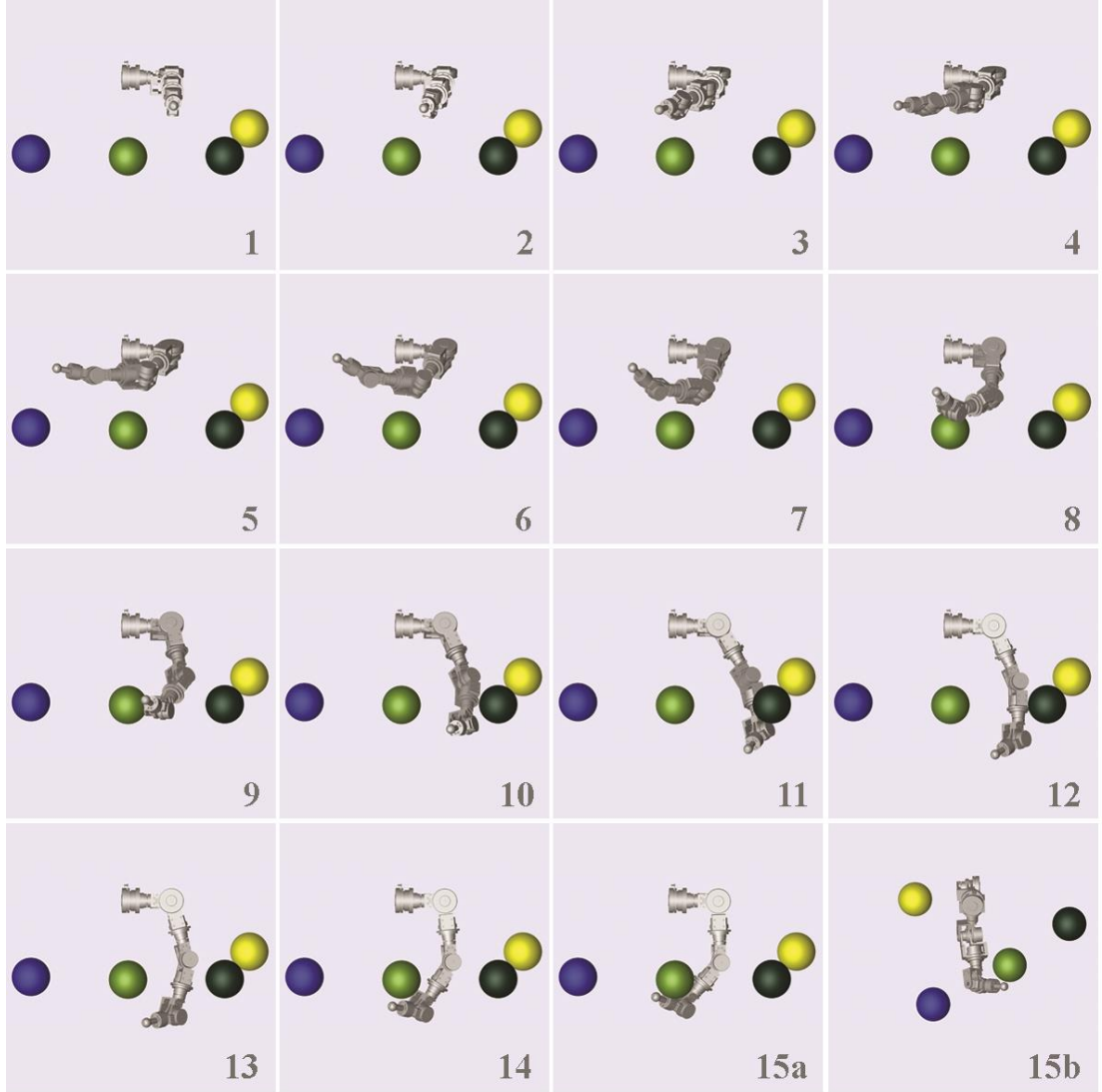
Şekil 5.7'de her bir eklem için elde edilen konum, hız, ivme ve sarsım profilleri verilmiştir. Sınır koşulları ile kinematik kısıtlar dikkate alınmış olup konum, hız ve ivme için süreklilik koşulu sağlanmıştır. Başlangıçta HRA tarafından verilen bağlanabilirlik yolu ile K-uzayında verilen pürüzsüz hareketin G-uzayındaki karşılığı aynı şekil (Şekil 5.8) üzerinde gösterilmiştir. Hedef bölgenin mavi bir kafes olarak gösterildiği bu şekilde, uç eyleyiciye ait 3 boyutlu G-uzayındaki hareketin de pürüzsüz olduğu görülmektedir.



Şekil 5.7 : Süreklileştirilen yörüngeye ait konum, hız, ivme ve sarsım profilleri.

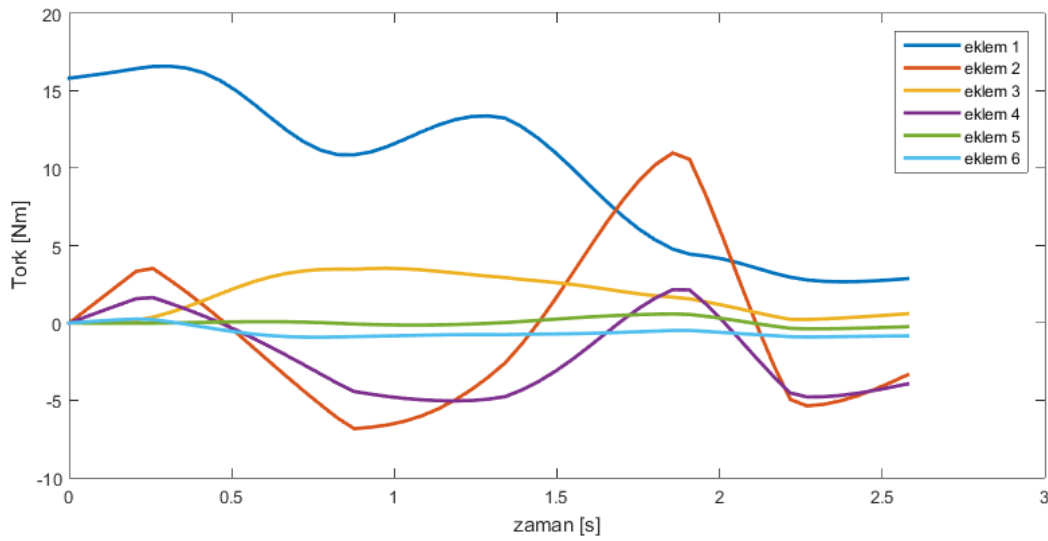


Şekil 5.8 : Uç eyleyici için HRA tarafından verilen yol (kesikli çizgi), süreklileştirilmiş yörünge (yeşil çizgi) ve hedef bölge (açık mavi kafes).



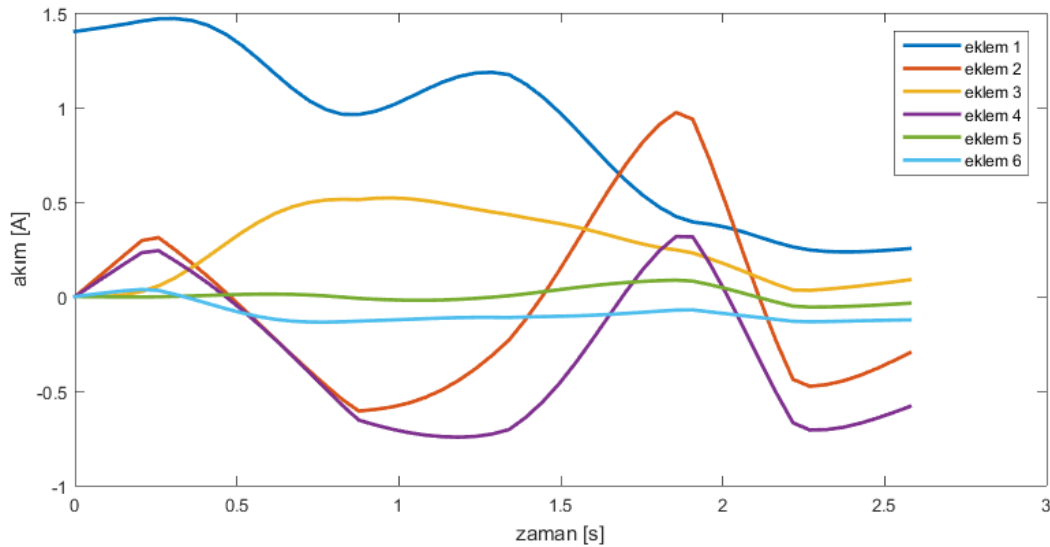
Şekil 5.9 : Robot kolun hareketine ait 15 ardışık kare.

Hareketin gerçekleşmesi için, SimMechanics™ tarafından verilen hesaplanmış tork değerleri Şekil 5.10'da görülmektedir. Bu değerler, dişli çevrim oranlarına bölündüğünde en yüksek tork değerinin 1. eklemdaki 103.69 mNm olduğu görülür. En yüksek motor akımı değeri ise, Şekil 5.11'de de görüleceği üzere yine 1. motor için 1.47 Amper olarak tespit edilir. Bunun için motor akımı (i) ile tork (τ) ilişkisini veren 5.13 kullanılır. Motor tork sabiti, k_t , katalogu EK B'de verilen EC 90 flat ve EC 60 flat motorlar için sırasıyla 70.5 ve 53.4 mNm/A'dır.



Şekil 5.10 : SimMechanics™ tarafından verilen hesaplanmış tork değerleri.

$$\tau = k_t i \quad (5.13)$$



Şekil 5.11 : Hareketin gerçekleşmesi için gerekli motor akım değerleri.

Görüldüğü üzere, en yüksek motor akımı ve tork değerleri katalogta verilen sınır değerlerin çok altında kalmaktadır.

Gerek HRA tarafından verilen bağlanabilirlik yolu gerekse sürekli hale getirilmiş yörünge optimum olmadığı, robot kolunun Şekil 5.9’da verilen ilk 7 karede sol tarafa gereksiz yere uzanıp zaman ve enerji kaybettiği görülmektedir. Bu durum, kısıtlar altında algoritmanın tamamen rastgele çalışmasından kaynaklanmakta ve hesap yükünden edilen tasarrufun bir sonucu olarak ortaya çıkmaktadır.

5.5 Manipülatör Dinamiği ve Kontrolü

Çalışmanın bu aşamasına kadar, hareketinde etkili olan yerçekimi kuvveti, merkezci etki, sürtünme ve Coriolis kuvvetlerini dikkate almaksızın bir yörünge planlama önerilmiştir. Bu alt bölümde, planlanan yörünge robot koluna kapalı çevrim bir sistemde nasıl beslenebileceğine dair klasik bir yaklaşım özetlenmiştir. Bu aşamada, MATLAB® SimMechanics™ araç kutusunda Güleç’in hazırladığı benzetim ortamı aynen kullanılmıştır (2014).

Bu benzetimde ilk olarak verilen yörüngeyi takip etmek için gerekli tork değerleri, Euler-Lagrange denklemleriyle hesaplanmıştır ve bunun için, Lagranjiyen (L) ifadesi 5.14’te kinetik enerji (K) ile potansiyel enerjinin (P) farkı olarak tanımlanmıştır. Sürtünmenin ve katılığın ihmal edildiği durumda, potansiyel enerji sadece yerçekiminden gelecektir. Euler-Lagrange denklemi (5.15) yazıldığında, tork 5.16’daki gibi bulunur. Burada q açısal konum, $\dot{q}$ açısal hız ve $\ddot{q}$ açısal ivme vektörleridir. D , C ve g ise sırasıyla kütle ya da atalet matrisi, Coriolis matrisi ve yerçekiminden kaynaklanan tork vektörüdür. Tork hesabı için kullanılan bu ters dinamik model Şekil 5.12’de görselleştirilmiştir.

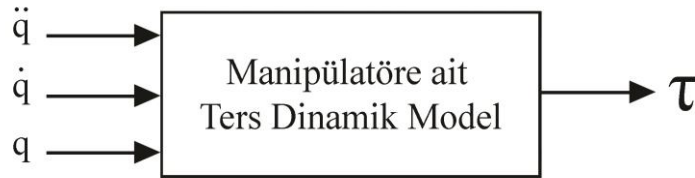
$$L = K - P \quad (5.14)$$

$$\tau = \frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} \quad (5.15)$$

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (5.16)$$

Dinamik modelin kesin olarak hiçbir şekilde bilinmeyeceği aşîkârdır. Bir başka deyişle, hesaplanmış tork değerleri açık çevrimde robota aynen verilirse, istenen

yörüngelerin elde edilmesi hiçbir zaman garanti değildir ve hatta, bu yörünge kuvvetle muhtemel izlenemeyecektir. Bunun için, yaklaşık hesaplanmış tork denetleyicisi (approximate computed-torque controller) 5.17'deki gibi önerilmiştir. Burada $\hat{D}$, $\hat{C}$, $\hat{g}$ ve $\hat{F}$ sırasıyla atalet, Coriolis, yerçekimi ve sürtünme terimlerinin yaklaşık ifadesidir. K_k ve K_h denetleyiciye ait kazanç matrisleri, q_i izlenmesi istenen (referans) yörünge, e ise yörünge izleme hatasıdır (Lewis ve diğ., 2004).



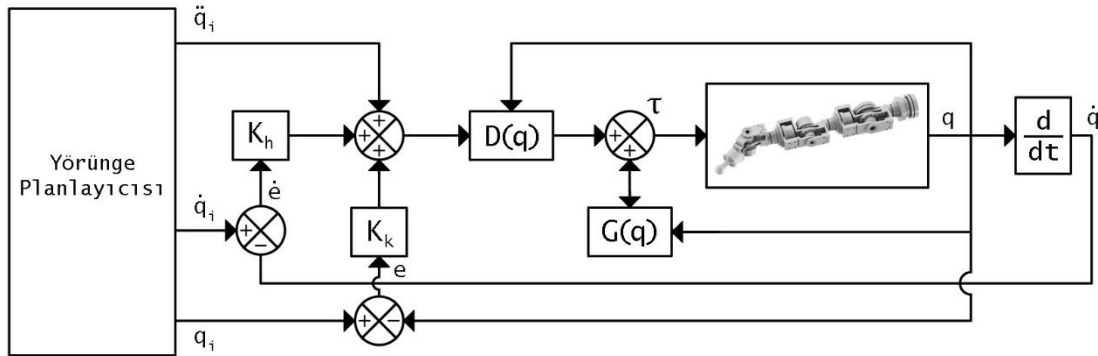
Şekil 5.12 : Ters dinamik modelin çalışma mantığı.

$$\tau = \hat{D}(q)(\ddot{q}_i + K_h \dot{e} + K_k e) + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) + F(\dot{q}) \quad (5.17)$$

Güleç'in çalışmasında (2014) yalnız D matrisi ve g vektörünün yaklaşık ifadesi ele alınmış ve Coriolis ile sürtünme modellenmemiştir. Dolayısıyla denklem

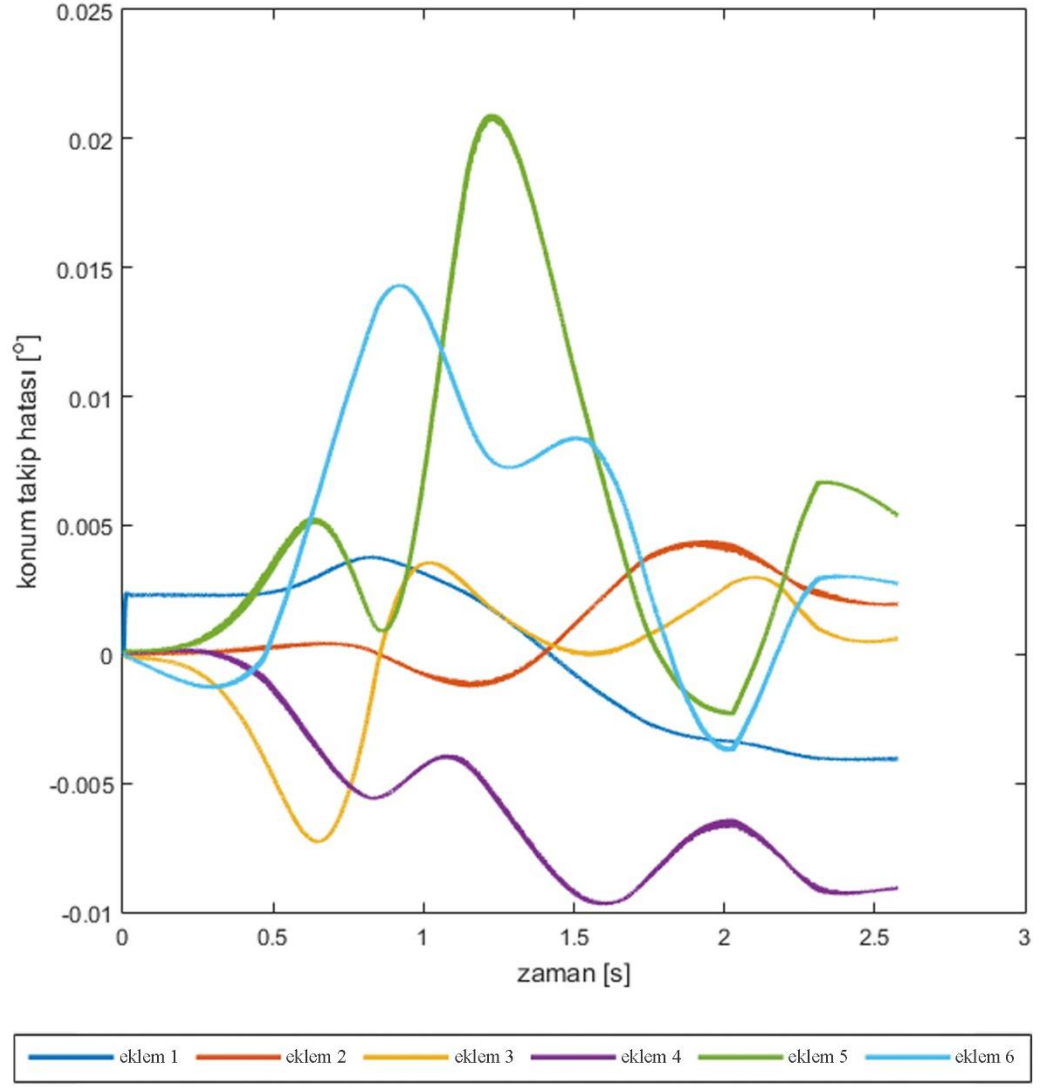
$$\tau = \hat{D}(q)(\ddot{q}_i + K_h \dot{e} + K_k e) + \hat{g}(q)$$

ifadesine indirgenir. Bu ifadeye indirgenmiş denetleyici yapısının blok diyagramında gösterimi Şekil 5.13'te verilmiştir.



Şekil 5.13 : Yaklaşık hesaplanmış tork kontrolcüsüne ait blok diyagramı.

K_h ve K_k kazanç katsayıları sırasıyla 1000 ve 550000 olarak belirlenip SimMechanics™'te bu benzetim gerçekleştirildiğinde, referans yörüngenin 0.022 derece gibi çok az bir hata ile takip edildiği görülür (Şekil 5.14).



Şekil 5.14 : Her bir eklem için yörünge takip hatası.



6. SONUÇ

HRA algoritması tarafından verilen yörünge'nin süreklileştirilmesi halen üzerinde çalışılan bir konudur. Bu çalışmada, öncelikle, 6 serbestlik dereceli bir manipulator için bağlanabilirlik yolu; örnekleme tabanlı yol planlama algoritmalarından son 20 yılda ortaya çıkıp yaygınlaşan HRA ile elde edilmiştir. Planlama K-uzayında yapıldığı için ters geometrik ya da ters kinematik modele bir kez dahi başvurulmamıştır. Bu işlemin peşi sıra, sürekli eklem yörüngelerinin zaman optimizasyonunu için Nelder-Mead yöntemi tabanlı bir çalışma (Lin ve diğ., 1983) kullanılmıştır. Bu uyarlama, diğer yörünge süreklileştirme yaklaşımlarından (Hauser ve Ng-Thow-Hing, 2010; Lau ve Byl, 2015) farklı olarak sarsım sınırlama ve zaman optimizasyonu avantajlarına sahiptir. Yukarıda bahsedilen manipulator için MATLAB®'de yazılan program, bir çözümün var olduğu bir durum için birçok defa çalıştırılmış ve birbirinden farklı uygulanabilir birçok yörünge bulmakta başarılı olmuştur.

Hareket için uygulanması gereken torklar, dinamik kararlılık ve manipulatorün kontrolü gibi dinamik hususlar bu çalışmanın merkezinde yer almamış olup sadece elde edilen yörünge'nin manipulator'e nasıl beslenebileceğine dair bir örnek verilmiştir. Bu örnekte, yaklaşık hesaplanmış tork denetleyicisi kullanılmış ve dinamik etkiler modellenirken Coriolis ile sürtünme terimleri hesaba katılmamıştır. Kinematik kısıtlar altında yapılan benzetimler sonucunda, motor tork ve akım sınır değerlerinin aşılmadığı görülmüştür. Ayrıca, yörünge takip hatasının çok düşük mertebelerde kaldığı gözlemlenmiştir.

İleriki çalışmalarda, tezde önerilen yörünge planlama yaklaşımının kinodinamik HRA'nın kullanıldığı bir hareket planlamasında kullanılması düşünülebilir. Ayrıca yöntem, asimptotik olarak optimize edilmiş HRA algoritması olan HRA* için de uyarlanıp sürekli ve optimum yörüngeler elde edilebilir.

İkinci olarak, HRA tarafından verilen bağlanabilirlik yolunun zaman optimizasyonu için Nelder-Mead yönteminden başka optimizasyon yöntemleri kullanılıp elde edilecek sonuçlar karşılaştırılabilir ya da Nelder-Mead yönteminde, en başta önerilen

zaman aralıkları vektörleri için farklı bir yaklaşım önerilerek optimizasyon süreci hızlandırılabilir.

Örnekleme tabanlı bir algoritma olan HRA'nın alt fonksiyonları tek tek alınacak olursa, uzaklık ölçeği olarak geleneksel olan Öklid mesafesi yerine Mahalanobis mesafesi gibi diğer seçenekler denenebilir, kullanılan en yakın komşu algoritması yerine ise yaklaşık en yakın komşu algoritmaları kullanılarak planlayıcı hesaplama süresi kısaltılabilir. Ayrıca daha yörünge süreklileştirme aşamasına geçmeden önce, görüş hattı (line of sight) yaklaşımlarından yararlanılarak robotik sistemin eylemi gerçekleştirme zamanı da ufak bir hesaplama maliyeti karşılığında büyük ölçüde azaltılabilir.

Son olarak algoritmanın, üretimi tamamlanan robot üzerinde montajı tamamlandıktan sonra deneysel olarak uygulanması ve alınacak sonuçların benzetim ortamındaki sonuçlarla karşılaştırılması planlanmaktadır. Bu bölümde önerilen tüm yaklaşım seçenekleri, kapsamlı bir çalışmada derlenip deneysel sonuçlarla birlikte sunulabilir.

KAYNAKLAR

- Amato, N., Bayazit, O., Dale, L., Jones, C., & Vallejo, D.** (2000). Choosing good distance metrics and local planners for probabilistic roadmap methods. In *IEEE Transactions on Robotics and Automation* (pp. 442-447).
- Aristidou, A. & Lasenby, J.** (2009). *Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver*. University of Cambridge.
- Asfour, T. & Dillmann, R.** (2003). Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1407 - 1412 vol.2). Las Vegas, NV.
- Barraquand, J., Langlois, B., & Latombe, J.** (1991). Numerical potential field techniques for robot path planning. In *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 2 (pp. 224-241).
- Bertram, D., Kuffner, J., Dillmann, R., & Asfour, T.** (2006). An Integrated Approach to Inverse Kinematics and Path Planning for Redundant Manipulators. In *IEEE International Conference on Robotics and Automation* (pp. 1874-1879). Orlando, FL.
- Boyacıoğlu, B. & Ertuğrul, Ş.** (2015). 6 Serbestlik Dereceli İnsansı Robot Kolun Engelden Kaçan Yörünge Planlaması. In *2. Türkiye Robotbilim Konferansı*. İstanbul.
- Canny, J., Donald, B., Reif, J., & Xavier, P.** (1988). *On the complexity of kinodynamic planning*. Ithaca, NY: Cornell Univ., Dep. of Computer Science.
- de Sousa Rocha, R. & Rodrigues, M.** (2008). An evaluation of a collision handling system using sphere-trees for plausible rigid body animation. In *2008 ACM Symposium on Applied Computing* (pp. 1241-1245). New York, NY.
- Diankov, R., Ratliff, N., Ferguson, D., Srinivasa, S., & Kuffner, J.** (2008). BiSpace Planning: Concurrent Multi-Space Exploration. In *Robotics: Science and Systems IV*.
- Elbanhawi, M., & Simic, M.** (2014). Sampling-Based Robot Motion Planning: A Review. *IEEE Access*, 2, 56-77. <http://dx.doi.org/10.1109/access.2014.2302442>
- Fraichard, T. & Asama, H.** (2003). Inevitable Collision States. A Step Towards Safer Robots?. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 388-393). Las Vegas, NV.

- Glassman, E. & Tedrake, R.** (2010). A Quadratic Regulator-Based Heuristic for Rapidly Exploring State Space. In *IEEE International Conference on Robotics and Automation* (pp. 5021-5028). Anchorage, AK.
- Güleç, M. Ö.** (2014). *İnsansı robot kolu tasarımı ve yörünge kontrolü*. (Yüksek lisans tezi). İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- Güleç, M. Ö. & Ertuğrul, Ş.** (2014). İnsansı robot kolu tasarımı ve yörünge kontrolü. In *Otomatik Kontrol Ulusal Toplantısı* (pp. 90-95). Kocaeli.
- Hauser, K., & Ng-Thow-Hing, V.** (2010). Fast Smoothing of Manipulator Trajectories using Optimal Bounded-Acceleration Shortcuts. In *IEEE International Conference on Robotics and Automation* (pp. 2493-2498). Anhorage, AK.
- Hollerbach, J. & Sahar, G. (1983).** *Wrist-partitioned inverse kinematic accelerations and manipulator dynamics*. [Cambridge, Mass.]: Massachusetts Institute of Technology, Artificial Intelligence Laboratory.
- Karaman, S. & Frazzoli, E.** (2011). Sampling-based algorithms for optimal motion planning. *The International Journal Of Robotics Research*, 30(7), 846-894. <http://dx.doi.org/10.1177/0278364911406761>
- Kavraki, L., Švestka, P., Latombe, J., & Overmars, M.** (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *IEEE Transactions on Robotics and Automation*, vol 12, no. 4 (pp. 566-580).
- Khalil, W., & Dombre, E.** (2004). *Modeling, identification & control of robots*. London: Kogan Page Science.
- Koyuncu, E. & İnalhan, G.** (2008). A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 815-821). Nice.
- Kuffner, J., & LaValle, S.** (2000). RRT-Connect: An Efficient Approach to Single-Query Path Planning. In *IEEE International Conference on Robotics and Automation* (pp. 1-7).
- Lau, C., & Byl, K.** (2015). Smooth RRT-Connect: An extension of RRT-Connect for practical use in robots. In *IEEE International Conference on Technologies for Practical Robot Applications* (pp. 1-7).
- LaValle, S., & Kuffner, J.** (1999). Randomized Kinodynamic Planning. In *IEEE International Conference on Robotics & Automation* (pp. 473-479). Detroit, MI.
- LaValle, S.** (2006). *Planning algorithms*. Cambridge: Cambridge University Press.
- Lemburg, J., de Gea Fernández, J., Eich, M., Mronga, D., Kampmann, P., & Vogt, A. et al.** (2011). AILA - Design of an autonomous mobile dual-arm robot. In *IEEE International Conference on Robotics and Automation* (pp. 5147-5153). Shangai.
- Lewis, F., Abdallah, C., Dawson, D., & Lewis, F.** (2004). *Robot manipulator control*. New York: Marcel Dekker.

- Lin, C., Chang, P., & Luh, J.** (1983). Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transactions on Automatic Control*, 28(12), 1066-1074. <http://dx.doi.org/10.1109/tac.1983.1103181>
- Lozano-Perez, T.** (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, C-32(2), 108-120. <http://dx.doi.org/10.1109/tc.1983.1676196>
- Megahed, S.** (1992). Inverse kinematics of spherical wrist robot arms: Analysis and simulation. *Journal of Intelligent and Robotic Systems*, 5(3). <http://dx.doi.org/10.1007/bf00247418>
- Michalos, G., Makris, S., Tsarouchi, P., Guasch, T., Kontovrakis, D., & Chryssolouris, G.** (2015). Design Considerations for Safe Human-robot Collaborative Workplaces. *Procedia CIRP*, 37, 248-253. <http://dx.doi.org/10.1016/j.procir.2015.08.014>
- Nelder, J. & Mead, R.** (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7(4), 308-313. <http://dx.doi.org/10.1093/comjnl/7.4.308>
- Pechev, A.** (2008). Inverse Kinematics without matrix inversion. In *IEEE International Conference on Robotics and Automation* (pp. 2005-2012). Pasadena, CA.
- Perez, A., Karaman, S., Shkolnik, A., Frazzoli, E., Teller, S., & Walter, M.** (2011). Asymptotically-optimal Path Planning for Manipulation using Incremental Sampling-based Algorithms. In *IEEE International Conference on Intelligent Robots and Systems* (pp. 4307-4313). San Francisco, CA.
- Reggiani, M., Mazzoli, M., & Caselli, S.** (2002). An experimental evaluation of collision detection packages for robot motion planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2329-2334). Lausanne.
- Rosen, J., Perry, J., Manning, N., Burns, S., & Hannaford, B.** (2005). The human arm kinematics and dynamics during daily activities - toward a 7 DOF upper limb powered exoskeleton. In *12th International Conference on Advanced Robotics* (pp. 532-539). Seattle, WA.
- Schwartz, J., & Sharir, M.** (1983). On the “piano movers” problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36(3), 345-398. <http://dx.doi.org/10.1002/cpa.3160360305>
- Shkolnik, A., & Tedrake, R.** (2009). Path planning in 1000+ dimensions using a task-space Voronoi bias. In *IEEE International Conference on Robotics and Automation* (pp. 2061-2067).
- Simba, K., Uchiyama, N., & Sano, S.** (2013). Real-time trajectory generation for mobile robots in a corridor-like space using Bézier curves. In *IEEE/SICE International Symposium on System Integration* (pp. 37-41). Kobe.

- Spong, M., Hutchinson, S., & Vidyasagar, M.** (2006). *Robot modeling and control*. Hoboken, NJ: John Wiley & Sons.
- Steinbach, K., Kuffner, J., Asfour, T., & Dillmann, R.** (2006). Efficient Collision and Self-Collision Detection for Humanoids Based on Sphere Trees Hierarchies. In *6th IEEE-RAS International Conference on Humanoid Robots* (pp. 560-566). Genova.
- Taylor, C.** (2016). *Robotics: Computational Motion Planning*. <https://www.coursera.org/learn/robotics-motion-planning> erişim tarihi 14.03.2016.
- Thompson, S. & Patel, R.** (1987). Formulation of Joint Trajectories for Industrial Robots Using B-Splines. *IEEE Trans. Ind. Electron., IE-34*(2), 192-199. <http://dx.doi.org/10.1109/tie.1987.350954>
- Tolani, D., & Badler, N.** (1996). Real-Time Inverse Kinematics of the Human Arm. *Presence: Teleoperators and Virtual Environments*, 5(4), 393-401. <http://dx.doi.org/10.1162/pres.1996.5.4.393>
- Tondu, B., Ippolito, S., Guiochet, J., & Daidie, A.** (2005). A Seven-degrees-of-freedom Robot-arm Driven by Pneumatic Artificial Muscles for Humanoid Robots. *The International Journal Of Robotics Research*, 24(4), 257-274. <http://dx.doi.org/10.1177/0278364905052437>
- Weghe, M., Ferguson, D., & Srinivasa, S.** (2007). Randomized Path Planning for Redundant Manipulators without Inverse Kinematics. In *IEEE-RAS International Conference on Humanoid Robots* (pp. 477-482).
- Weller, R.** (2013). *New geometric data structures for collision detection and haptics*. Cham: Springer.
- Zacharias, F.** (2012). *Knowledge representations for planning manipulation tasks*. Berlin: Springer.
- Url-1** <<http://projetromeo.com>>, erişim tarihi 23.06.2015.
- Url-2** <https://en.wikipedia.org/wiki/Voronoi_diagram>, erişim tarihi 01.05.2016.
- Url-3** <<http://gamma.cs.unc.edu/research/collision/packages.html>>, erişim tarihi 23.04.2016.

EKLER

EK A: 3.8 numaralı denklemin sađ tarafının elde edilmesi (Lin ve diđ., 1983)

EK B: EC 90 flat ve EC 60 flat motorlarına ait kataloglar (Güleç, 2014)



EK A

Hız için süreklilik şartı $i = 2, 3, \dots, n-1$ için $Q'_i(t_i) = Q'_{i-1}(t_i)$ eşitliğini verir ki bu 3.5 ve 3.6 numaralı denklemler kullanılarak A.1'in yazılmasını sağlayacaktır.

$$\begin{aligned} \frac{h_{i-1}}{h_i} Q''_{i-1}(t_{i-1}) + \frac{2(h_i + h_{i-1})}{h_i} Q''_i(t_i) + Q''_i(t_{i+1}) \\ = \frac{6}{h_i} \left[\frac{q_{i+1} - q_i}{h_i} - \frac{q_i - q_{i-1}}{h_{i-1}} \right] \end{aligned} \quad (\text{A.1})$$

İlave edilen iki geçiş noktasının kesin olmayan konumu, başlangıç ve son noktadaki sınır koşulları ile $Q''_1(t_2)$ ve $Q''_{n-2}(t_{n-1})$ cinsinden A.2 ve A.3'teki gibi ifade edilebilir.

$$q_2 = q_1 + h_1 \dot{q}_1 + \frac{h_1^2}{3} \ddot{q}_1 + \frac{h_1^2}{6} Q''_1(t_2) \quad (\text{A.2})$$

$$q_{n-1} = q_n - h_{n-1} \dot{q}_n + \frac{h_{n-1}^2}{3} \ddot{q}_n + \frac{h_{n-1}^2}{6} Q''_{n-2}(t_{n-1}) \quad (\text{A.3})$$

A.2 ve A.3, A.1'de yerine yazılacak olursa, aşağıdaki sistem (A.4-8) elde edilir.

$$\begin{aligned} \left[\frac{2(h_1 + h_2)}{h_2} + \frac{h_1^2}{h_2} \left(\frac{1}{h_1} + \frac{1}{h_2} \right) \right] Q''_2(t_2) + Q''_2(t_3) \\ = \frac{6}{h_2} \left(\frac{q_3}{h_2} + \frac{q_1}{h_1} \right) - \frac{6}{h_2} \left(\frac{1}{h_1} + \frac{1}{h_2} \right) \cdot \left(q_1 + h_1 \dot{q}_1 + \frac{h_1^2}{3} \ddot{q}_1 \right) \\ - \frac{h_1}{h_2} \ddot{q}_1 \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} \left(\frac{h_2}{h_3} - \frac{h_1^2}{h_2 h_3} \right) Q''_2(t_2) + \frac{2(h_2 + h_3)}{h_3} Q''_3(t_3) + Q''_3(t_4) \\ = \frac{6}{h_2 h_3} \left(q_1 + h_1 \dot{q}_1 + \frac{h_1^2}{3} \ddot{q}_1 \right) + \frac{6q_4}{h_3^2} - \frac{6}{h_3} \left(\frac{1}{h_2} + \frac{1}{h_3} \right) q_3 \end{aligned} \quad (\text{A.5})$$

$$\begin{aligned} \frac{h_{i-1}}{h_i} Q''_{i-1}(t_{i-1}) + \frac{2(h_i + h_{i-1})}{h_i} Q''_i(t_i) + Q''_i(t_{i+1}) \\ = \frac{6}{h_i} \left[\frac{q_{i+1} - q_i}{h_i} - \frac{q_i - q_{i-1}}{h_{i-1}} \right], \quad i = 4, 5, \dots, n-3 \end{aligned} \quad (\text{A.6})$$

$$\frac{h_{n-3}}{h_{n-2}} Q''_{n-3}(t_{n-3}) + \frac{2(h_{n-2} + h_{n-3})}{h_{n-2}} Q''_{n-2}(t_{n-2}) \quad (\text{A.7})$$

$$\begin{aligned} &+ \left(1 - \frac{h_{n-1}^2}{h_{n-2}^2}\right) Q''_{n-2}(t_{n-1}) \\ &= \frac{6}{h_{n-2}^2} \left(q_n - h_{n-1} \dot{q}_n + \frac{h_{n-1}^2}{3} \ddot{q}_1 \right) \\ &\quad - \frac{6}{h_{n-2}} \left(\frac{1}{h_{n-2}} + \frac{1}{h_{n-3}} \right) q_{n-2} + \frac{6}{h_{n-2} h_{n-3}} q_{n-3} \end{aligned}$$

$$\frac{h_{n-2}}{h_{n-1}} Q''_{n-2}(t_{n-2}) + \left[\frac{2(h_{n-1} + h_{n-2})}{h_{n-1}} + \left(\frac{1}{h_{n-1}} + \frac{1}{h_{n-2}} \right) h_{n-1} \right] Q''_{n-2}(t_{n-1}) \quad (\text{A.8})$$

$$\begin{aligned} &= -\frac{6}{h_{n-1}} \left(\frac{1}{h_{n-1}} + \frac{1}{h_{n-2}} \right) \left(q_n - h_{n-1} \dot{q}_n + \frac{h_{n-1}^2}{3} \ddot{q}_1 \right) \\ &\quad + \frac{6q_n}{h_{n-1}^2} + \frac{6q_{n-2}}{h_{n-1} h_{n-2}} + a_n \end{aligned}$$

EC 90 flat Ø90 mm, brushless, 90 Watt



Part Numbers

Specifications		Operating Range		Comments	
Thermal data				Continuous operation In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient. = Thermal limit.	
17 Thermal resistance housing-ambient	1.91 K/W				
18 Thermal resistance winding-housing	2.6 K/W				
19 Thermal time constant winding	46.6 s				
20 Thermal time constant motor	283 s				
21 Ambient temperature	-40...+100°C				
22 Max. permissible winding temperature	+125°C				
Mechanical data (preloaded ball bearings)				Short term operation The motor may be briefly overloaded (recurring).	
23 Max. permissible speed	5000 rpm				
24 Axial play at axial load < 15 N	0 mm				
> 15 N	0.14 mm				
25 Radial play	preloaded				
26 Max. axial load (dynamic)	12 N				
27 Max. force for press fits (static) (static, shaft supported)	183 N				
28 Max. radial loading, 7.5 mm from flange	8000 N				
	60 N				

maxon Modular System

Overview on page 20 - 25

12 **Planetary Gearhead**
3 $\varnothing 52$ mm
600 g 4 - 30 Nm
Page 274

Encoder MILE
800 - 6400 CPT,
2 channels
Page 294

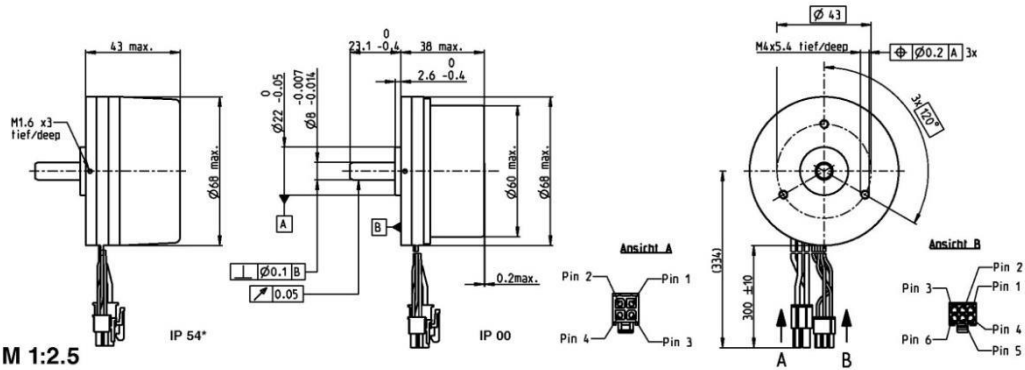
Cable	
Connection cable Universal, L = 500 mm	339380
Connection cable to EPOS2, L = 500 mm	354045

Recommended Electronics:	
ESCON 50/5	Page 32
ESCON Module 50/5	32
ESCON 70/10	32
DECS 50/5	32
DEC Module 50/5	32
EPOS2 24/5, 50/5, 70/10	33
EPOS2 P 24/5	33
EPOS3 70/10 EtherCAT	33
Notes	2

EC 60 flat Ø60 mm, brushless, 100 Watt

NEW

maxon flat motor



M 1:2.5

- Stock program
- Standard program
- Special program (on request)

Article numbers

IP 54* (with cover)
IP 00 (without cover)

412819	408057	412821
412823	411678	412825

Motor Data (provisional)

Values at nominal voltage

	V	12	12	24	24	48	48
1 Nominal voltage	V	12	12	24	24	48	48
2 No load speed	rpm	3710	3710	4250	4250	3970	3970
3 No load current	mA	671	671	419	419	187	187
4 Nominal speed	rpm	3290	3190	3850	3740	3580	3500
5 Nominal torque (max. continuous torque)	mNm	216	267	221	284	255	317
6 Nominal current (max. continuous current)	A	7.37	8.88	4.33	5.39	2.28	2.77
7 Stall torque	mNm	2820	2820	3740	3740	4190	4190
8 Starting current	A	93.5	93.5	78.2	78.2	43.8	43.8
9 Start efficiency	%	84	84	86	86	88	88

Characteristics

10 Terminal resistance phase to phase	Ω	0.128	0.128	0.307	0.307	1.1	1.1
11 Terminal inductance phase to phase	mH	0.0615	0.0615	0.188	0.188	0.864	0.864
12 Torque constant	mNm/A	30.5	30.5	53.4	53.4	114	114
13 Speed constant	rpm/V	313	313	179	179	83.4	83.4
14 Speed/torque gradient	rpm/mNm	1.32	1.32	1.03	1.03	0.798	0.798
15 Mechanical time constant	ms	16.7	16.7	13	13	10.1	10.1
16 Rotor inertia	gcm ²	1210	1210	1210	1210	1210	1210

Specifications

- 17 Thermal resistance housing-ambient 4.31 (2.41) K/W
- 18 Thermal resistance winding-housing 3.8 K/W
- 19 Thermal time constant winding 40 s
- 20 Thermal time constant motor 155 (86.9) s
- 21 Ambient temperature -40...+100°C
- 22 Max. permissible winding temperature +125°C

Mechanical data (preloaded ball bearings)

- 23 Max. permissible speed 6000 rpm
- 24 Axial play at axial load < 15.0 N 0 mm
- 25 Radial play > 15.0 N 0.14 mm
- 26 Max. axial load (dynamic) 12 N
- 27 Max. force for press fits (static) 170 N
- 28 Max. radial loading, 7.5 mm from flange 6000 N

Other specifications

- 29 Number of pole pairs 7
- 30 Number of phases 3
- 31 Weight of motor 470 g

Values listed in the table are nominal.

Connection motor (Cable AWG 16)

- red Motor winding 1 Pin 1
- black Motor winding 2 Pin 2
- white Motor winding 3 Pin 3
- N.C. N.C. Pin 4

Connector Article number

Molex 39-01-2040

Connection Sensors (Cable AWG 26)

- grey Hall sensor 1 Pin 1
- grey Hall sensor 2 Pin 2
- grey Hall sensor 3 Pin 3
- grey GND Pin 4
- blue V_{bat} 4.5...18 VDC Pin 5
- N.C. N.C. Pin 6

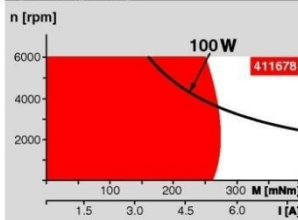
Connector Article number

Molex 430-25-0600

* Wiring diagram for Hall sensors see p. 29

* Protection level only when installed with flange-side seal.

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

maxon Modular System

Planetary Gearhead

Ø52 mm

4 - 30 Nm

Page 246



Recommended Electronics:

ESCON 50/5 Page 292

DEC Module 50/5 299

DEC 70/10 305

EPOS2 24/5 313

EPOS2 50/5 313

EPOS2 70/10 313

EPOS2 P 24/5 316

EPOS3 70/10 319

Notes 20

Overview on page 16 - 21

Encoder MILE

512 - 2048 CPT,

2 channels

Page 263



ÖZGEÇMİŞ



Ad-Soyad : Burak Boyacıoğlu
Doğum Tarihi ve Yeri : 20.08.1991 / Aydın
E-posta : boyaciogluburak@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2013, İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, Uçak Mühendisliği Bölümü

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN YAYINLAR

- Boyacıoğlu, B. & Ertuğrul, Ş. (2015). 6 Serbestlik Dereceli İnsansı Robot Kolun Engelden Kaçan Yörünge Planlaması. In *2. Türkiye Robotbilim Konferansı*. İstanbul.
- Boyacıoğlu, B. & Ertuğrul, Ş. (2016). Time-optimal Smoothing of RRT-given Path for Manipulators. In *13rd International Conference on Informatics in Control*. Lisbon (kabul edildi).