

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİYEL ROBOTLARDA ZAMAN OPTİMAL YÖRÜNGE KONTROLÜ

YÜKSEK LİSANS TEZİ

Mehmet ÜNLÜEL

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

MAYIS 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİYEL ROBOTLARDA ZAMAN OPTİMAL YÖRÜNGE KONTROLÜ

YÜKSEK LİSANS TEZİ

**Mehmet ÜNLÜEL
(518111022)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şeniz ERTUĞRUL

MAYIS 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 518111022 numaralı Yüksek Lisans Öğrencisi **Mehmet ÜNLÜEL**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**ENDÜSTRİYEL ROBOTLARDA ZAMAN OPTİMAL YÖRÜNGE KONTROLÜ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

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

Jüri Üyeleri : **Doç. Dr. Pınar BOYRAZ**
İstanbul Teknik Üniversitesi

Y. Doç. Dr. Elif Erzan TOPÇU
Uludağ Üniversitesi

Teslim Tarihi : **28 Nisan 2015**
Savunma Tarihi : **26 Mayıs 2015**

ÖNSÖZ

Bu tez çalışmasının ortaya çıkmasında bana destek veren danışmanım Prof. Dr. Şeniz Ertuğrul'a robotik ve kontrol alanında beni yönlendirdiği için teşekkür ederim. Ayrıca aileme ve Türker Dilber'e ve Pınar Karamıkoğlu'na desteklerinden dolayı teşekkür ederim.

Mayıs 2015

Mehmet Ünlüel
Makina Mühendisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.2 Literatür Araştırması	2
1.3 Hipotez	4
2. UZAYSAL OPERATÖR CEBİRİ İLE KİNEMATİK VE DİNAMİK	
MODELLEME.....	9
2.1 Uzaysal Operatör Cebiri (UOA)	9
2.2 Seri Manipülatör Kinematiği.....	11
2.3 Seri Manipülatör Dinamiği.....	15
3. OPTİMAL HAREKET PLANLAMA	21
3.1 Giriş.....	21
3.2 Optimal Hareket Planlama Yöntemleri	21
3.2.1 Optimizasyon metotları.....	21
3.2.1.1 Kapsamlı arama.....	22
3.2.1.2 Analitik optimizasyon	22
3.2.1.3 Downhill simpleks metodu	22
3.2.1.4 Doğru arama metodu ile optimuma ulaşma	23
3.3 Konveks Optimizasyon	23
3.4 Zaman Optimal Yörünge Planlama.....	23
3.5 Zaman Optimal Yörünge Problemin Genelleştirilmesi.....	27
3.5.1 Yeni amaç fonksiyonlarının oluşturulması	27
3.5.2 Enerji optimal yörünge.....	28
3.5.3 İvme, tork değişim oranı kısıtlarının denkleme eklenmesi	28
3.5.3.1 Hız kısıtı	28
3.5.3.2 İvme kısıtı	29
3.5.3.3 Tork değişim kısıtı	30
3.5.4 Optimizasyon denklemlerinin genelleştirilmesi.....	30
3.6 Genelleştirilmiş Optimal Kontrol Probleminin Çözüm Yöntemleri	31
3.6.1 Doğrudan dönüşüm yöntemi	31
3.6.2 İkinci dereceden konik programlama (SOCP)	34
4. OPTİMAL YÖRÜNGE TAKİBİ.....	39
4.1 Robot Özellikleri	39
4.2 Kinematik ve Dinamik Parametrelerin Elde Edilmesi	40
4.3 Optimizasyon Tanımlarının Yazılıma Eklenmesi	41

4.3.1 Daire çizdirme uygulaması.....	42
4.3.2 Doğrusal yörüngede hareket uygulaması	49
5. SONUÇ VE ÖNERİLER.....	57
KAYNAKLAR.....	59
ÖZGEÇMİŞ	63

KISALTMALAR

DOF	: Serbestlik Derecesi
SGK	: Sanal Gerçeklik Kütüphanesi
UOC	: Uzaysal Operatör Cebiri
$SO(n)$	: n 'ye ait Ortogonal Grup
$so(n)$	: n 'ye ait Ters Simetrik Matris
KM	: Kütle Merkezi
SOCP	: İkinci Dereceden Konik Programlama (Second Order Cone Programming)

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 4.1: Robotun teknik özellikleri ve boyutları.....	39
Çizelge 4.2: Eksenel performans değerleri... ..	40

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Manipülâtör i'ye ait bağlantı k ile ilişkili vektörler.	9
Şekil 2.2 : Sabit platformda seri manipulatör.	11
Şekil 3.1 : $s \in [0, 1]$ için parçalı değişmez $\alpha(s)$	31
Şekil 3.2 : $s \in 0, 1$ için parçalı doğrusal bs	32
Şekil 3.3 : $s \in 0, 1$ için parçalı doğrusal olmayan $\tau i(s)$	32
Şekil 4.1 : Robotun eksenel gösterimi.	39
Şekil 4.2 : Ters kinematik hesaplama Matlab Simulink bloğu.	41
Şekil 4.3 : Yörünge planlaması Matlab Simulink bloğu.	41
Şekil 4.4 : Virtual Reality’de robotun izlediği yörünge’nin gösterilmesi.	42
Şekil 4.5 : Optimizasyon öncesi eklem konumunun zamana bağlı grafiği.	43
Şekil 4.6 : Optimizasyon sonrası eklem konumunun zamana bağlı grafiği.	43
Şekil 4.7 : Optimizasyon öncesi ve sonrası konumun zamana bağlı grafiği.	44
Şekil 4.8 : Optimizasyon öncesi eklem hızlarının zamana göre grafiği.	44
Şekil 4.9 : Optimizasyon sonrası eklem hızlarının zamana göre grafiği.	45
Şekil 4.10 : Optimizasyon öncesi ve sonrası hızların zamana göre grafiği.	45
Şekil 4.11 : Optimizasyon öncesi eklem ivmelerinin zamana göre grafiği.	46
Şekil 4.12 : Optimizasyon öncesi eklem ivmelerinin zamana göre grafiği.	46
Şekil 4.13 : Optimizasyon öncesi ve sonrası hızların zamana göre grafiği.	47
Şekil 4.14 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.	47
Şekil 4.15 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.	48
Şekil 4.16 : Zaman ve enerjinin birbirlerine göre değişimi.	48
Şekil 4.17 : Yörünge parametreleri girişi Matlab Simulink bloğu.	49
Şekil 4.18 : Virtual Reality’de robotun izlediği yörünge’nin gösterilmesi.	50
Şekil 4.19 : Optimizasyon öncesi eklem konumlarının zamana bağlı grafiği.	50
Şekil 4.20 : Optimizasyon sonrası eklem konumlarının zamana bağlı grafiği.	51
Şekil 4.21 : Optimizasyon öncesi ve sonrası konumların zamana bağlı grafiği.	51
Şekil 4.22 : Optimizasyon öncesi eklem hızlarının zamana bağlı grafiği.	52
Şekil 4.23 : Optimizasyon sonrası eklem hızlarının zamana bağlı grafiği.	52
Şekil 4.24 : Optimizasyon öncesi ve sonrası hızların zamana bağlı grafiği.	53
Şekil 4.25 : Optimizasyon öncesi eklem ivmelerinin zamana bağlı grafiği.	53
Şekil 4.26 : Optimizasyon sonrası eklem ivmelerinin zamana bağlı grafiği.	54
Şekil 4.27 : Optimizasyon öncesi ve sonrası ivmelerin zamana bağlı grafiği.	54
Şekil 4.28 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.	55
Şekil 4.29 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.	55

ENDÜSTRİYEL ROBOTLARDA ZAMAN OPTİMAL YÖRÜNGE KONTROLÜ

ÖZET

Optimal robot hareket planlaması robot sistemlerinin verimliliğini en yüksek seviyeye çıkarmak için çok önemlidir. Robot hareketlerinde dinamik ve kinematik hareketlerin istenen kriterlere göre iyileştirildiği optimizasyon ile robot kendi kapasitesini tamamen kullanır ve eyleyicilerinden tam olarak yararlanır.

İşlem kapasitesi yüksek bilgisayarların endüstride kolaylıkla satın alınabilir olmasıyla ve optimizasyon metotlarının etkinliğinin artmasıyla, bilgisayar destekli hareket planlaması günden güne daha fazla kullanılmaktadır. Bilgisayar destekli robot hareket planlaması, operatörlere yüklenen optimizasyon yükünü azaltır aynı zamanda robotların duruş zamanının azaltırken üretim kapasitesini artırır.

Kaynak ve yapıştırma gibi endüstriyel uygulamalarda robot sürekli olarak aynı hareketi tekrar eder. Bu sebeple yörünge planlaması uygulama özelinde en hızlı, en hassas, en düşük enerji gibi özel kriterlere göre yapılmalıdır. Yörünge zaman bağımlılığı belirli kısıtlamalar çerçevesinde optimize edilirken, robotların özel görevlere göre belirlenmiş yörüngeleri hareket planlama problemlerini ortaya çıkarmaktadır.

Kinematik ve dinamik denklemlerde UOC (uzaysal operator cebiri) metodu kullanılmıştır. Bu metot diğer kinematik ve dinamik hareket planlaması metotlarına göre değişen serbestlik derecelerine göre denklemlerin güncellenmesi çok kolay olan bir yöntemdir.

Optimal robot hareket planlama, robotlara ait eyleyici sınırlarını dikkate alarak zaman, enerji ve sarsılma gibi robotların uygulamalarında ihtiyaç duyulan kriterleri optimize eder. Ayrıca ilk ve son durumlar arasında uygulama uzayı veya eksenlerdeki yörüngelerin istenilen optimizasyon kriterine göre hesaplanmasını içerir. Hareket planlama problemlerinde, optimizasyon kriteri zaman ise zaman bağımlı optimal yörünge planlama, eğer optimizasyon kriteri enerji ise enerji bağımlı optimal yörünge planlama adını alır. Bu tez, genel hareket planlama problemlerini dikkate almasının yanı sıra özellikle zamana bağımlı optimal robot hareket planlaması ve zaman-enerji bağımlı optimal robot hareket planlaması problemlerine dayanmaktadır.

Optimal yörünge planlamasında optimizasyon kriterleri yanında hız, ivme ve tork gibi kısıt kriterleride önemlidir. Yörünge belirlenmesinde iki ayrı yoldan gidilebilir: Kinematik yörünge planlaması ve Dinamik yörünge planlaması. Kinematik yörünge planlaması kütle, atalet, kuvvet, tork gibi değerlerle ilgilenmez. Engelden sakınma optimizasyonu için genetik algortimalar kullanılmaktadır. Gidilen zamanın ve tüketilen enerjinin minimuma indirilmesi için adaptif genetik algortima, ikinci dereceden konik programlama, parçacık sürü optimizasyonu benzeri teknikler kullanılabilir.

Yörünge optimizasyonunda ise konveks optimizasyon metotlarından ikinci dereceden konik programlama metodu kullanılmıştır. Minimum zamana ulaşmak optimizasyon amacı olarak belirlenmiştir. Geliştirilen uygulamalarda robota dairesel yörünge takibi ve doğrusal yörünge takibi yaptırılmıştır. Dairesel yörünge takibinde robotun yörüngeyi tamamlama zamanı 3.6 saniyeden 1.5 saniyeye indirilmiştir.

Bu çalışmanın sonrasında benzetim ortamında denenmiş olan yazılımın çevrimdışı olarak bir robot üzerinde denenmesi düşünülmektedir. Bir sonraki çalışmada ise çevrimiçi olarak robota optimal yörünge planlamasının yapılması ve bunun için optimizasyon algoritmasının hızlandırılması için çalışması amaçlanmaktadır.

TIME OPTIMAL ROBOT PATH CONTROL FOR INDUSTRIAL ROBOTS

SUMMARY

Industrial robots are used in repeated tasks. The path followed by the robot differs with the change of product. In mass production line product change frequencies are longer than a month. In order to increase the efficiency of the robot, task duration must be decreased. Because of this reason robot motions must be planned. Robot motion is not only used for efficiency but also used for task specific requirements.

Robot motion planning is the most significant part of robot control at industrial robot systems. The motion have to be done by satisfying given conditions. Defined path have to be tracked in terms of given jerk, acceleration, velocity and position for each actuator of the robot.

Optimal robot motion planning has a significant role when maximizing the level of efficiency of the robot system. Dynamic and kinematic optimization is improved by optimizing the robot path according to the criteria for fully utilizing its capacity and take full performance of the actuator.

In industrial applications, last ten years it is easy to get high process capacity computer for controlling the robot. As the effectiveness of the optimization methods increases, computer-aided motion planning are widely used day by day. Robot producers are focused on the optimal motion planning. The mechanical properties of the same class robots are almost same. Major difference that can be added to robot is done by newer software.

Computer-aided robot motion planning, optimization reduces the work load on the operator which increases the production capacity of the robot while reducing downtime. For one cycle optimal motion planning can reduce about milliseconds or seconds. These milliseconds or seconds can increase the time efficiency of the motion about fifty percent.

In industrial applications such as welding, bonding and painting robot repeats the same motion at the mass production line. Trajectory planning application should be made as fast as possible or as precise as possible. In applications common used criterias are minimum energy, minumum time and minimum jerk. Predetermined path have to be optimized under certain restrictions according to the specific tasks in the optimization problems.

SOA spatial operator algebra methods for kinematic and dynamic equations are used. Generally Denavit Hartenberg method is use to solve kinematic and dynamic equations of robot. SOA method is an advance method which is very easy when it needs to update kinematic and dynamic equations of motion planning. This method has simple stucture when used in varying degrees of freedom. It is widely used when solving complex robot mechanisms.

Optimal robot motion planning is based on optimizing the robot actuator. It bases on time, energy and vibration criteria in practice while taking into account the limits of the actuators. Optimization saturates at least one actuator of the robot in a sample time of the optimization. It also includes the calculation according to the initial and last states in the application or by the desired optimization of the trajectory in operational space.

The motion planning problem is classified by the optimization criteria. The optimization method used in this thesis is named time-dependent optimal trajectory planning which minimizes the time. If the optimization criteria is energy, it will be named as energy dependent optimal trajectory planning. This thesis is based on the general motion planning problems as well as the optimization, especially considering the time-dependent optimal robot motion planning and time-energy-dependent optimal robot motion planning problem.

The optimization criteria in the optimal trajectory planning have to be included in the generalized equation. Objective function is defined in order to minimize or maximize the value of optimization criteria. System limits such as speed, acceleration, jerk and torque are defined in the optimization problem. Industrial robot arm has to complete the motion without surplusing the speed, acceleration, jerk and torque limits of the actuators.

Besides optimization criteria in the optimal trajectory planning speed, acceleration and torque constraints have significance. The determination of the path can be reached in two ways: Kinematic and dynamic trajectory planning trajectory planning. Kinematic trajectory planning does not deal with values mass, inertia, force and torque. In Dynamic trajectory planning values; mass, inertia, force and torque are taken into account.

Obstacle avoidance can be applied for optimization by using genetic algorithms. In order to complete path in minimum time or minimum energy adaptive genetic algorithms, the second order cone programming, particle swarm optimization techniques can be used.

In the trajectory optimization firstly optimization problem is converted to convex optimization problem beacuse of solving optimization Convex optimization problem is generalized in terms of notation used in this thesis. Generalized optimization problem is solved by using second order cone programming method. The optimization purposes that used in this thesis can be summarized as completing the path in minimum time.

Circular trajectory tracking of robot and linear trajectory tracking applications are investigated in simulation. Trajectory of the robot at circular path, time is reduced from 3.6 seconds to 1.5 seconds after the optimization method applied. While trajectory time decreases the energy of the actuator increases. At circular path pseudo energy that indicates mean value of actuators torques is increased 5 times higher than the energy before the optimization. In addition at the linear path tracking time of the robot is reduced from 4.0 seconds to 1.2 seconds. It can concluded from these results time optimality of the robot motion planning is successfully achieved.

The software is tested in the offline simulation environment later in this study are likely to be tested on a industrial robot arm. Simulation results indicates that the solution gathered in the study have reasonable values. Real system control will be a bit different. The time spended for the optimization calculation can be reduced. In a

subsequent study of the optimal trajectory planning of robot will be examined online and is intended to work to accelerate the optimization algorithm for it. Online time optimal control or online energy optimal control will be very useful when operating to the industrial robot arm.

1. GİRİŞ

Optimal robot hareket planlaması robot sistemlerinin verimliliğini en yüksek seviyeye çıkarmak için çok önemlidir. Robot hareketlerinde dinamik ve kinematik hareketlerin istenen kriterlere iyileştirildiği optimizasyon ile robot kendi kapasitesini tamamen kullanır ve eyleyicilerinden tam olarak yararlanır. Robotlarda dinamik denklemlerin doğrusal olmamasından dolayı [1], [2], [3] özellikle engellerin varlığında [4] hareket planlama son derece karmaşık ve zor bir görevdir [5]. Bu nedenle endüstriyel uygulamalarda, robot hareketlerinin planlanması genellikle manuel optimizasyon ve ayar ile gerçekleştirilir. İşlem kapasitesi yüksek bilgisayarların endüstride kolaylıkla satın alınabilir olmasıyla ve optimizasyon metotlarının etkinliğinin artmasıyla, bilgisayar destekli hareket planlaması günden güne daha fazla kullanılmaktadır. Bilgisayar destekli robot hareket planlaması, operatörlere yüklenen optimizasyon yükünü azaltır aynı zamanda robotların duruş zamanının azaltırken üretim kapasitesini artırır.

Kaynak ve yapıştırma gibi endüstriyel uygulamalarda robot sürekli olarak aynı hareketi tekrar eder. Bu sebeple yörünge planlaması uygulama özelinde en hızlı, en hassas, en düşük enerji gibi özel kriterlere göre yapılmalıdır. Yörünge zaman bağımlılığı belirli kısıtlamalar çerçevesinde optimize edilirken, robotların özel görevlere göre belirlenmiş yörüngeleri hareket planlama problemlerini ortaya çıkarmaktadır. Yörünge takibi adı verilen bu hareket planlama problemleri tezin ana bölümünü oluşturmaktadır.

1.1 Tezin Amacı

Optimal robot hareket planlama sorunları, optimizasyon problemleri sınıfından olan, optimum kontrol problemleri sınıfına aittir [6, 7]. Bölüm 3’de genel optimal kontrol metodları ve tezde kullanılan optimal kontrol metodu olan dışbükey optimizasyon anlatılmaktadır. Dışbükey optimizasyon teorisi ayrıntılı olarak incelenmiştir. 4. bölümde optimal yörünge planlamasının bir robota uygulanması ele alınacaktır.

Zamana bağımlı yörünge planlaması ve enerji-zaman bağımlı yörünge planlaması alt başlıklar halinde incelenip sonuçları karşılaştırılacaktır.

1.2 Literatür Araştırması

Optimal robot hareket planlama, robotlara ait eyleyici sınırlarını dikkate alarak zaman, enerji ve sarsılma gibi robotların uygulamalarında ihtiyaç duyulan kriterleri optimize eder. Ayrıca ilk ve son durumlar arasında uygulama uzayı veya eksenlerdeki yörüngelerin istenilen optimizasyon kriterine göre hesaplanmasını içerir. Hareket planlama problemlerinde, optimizasyon kriteri zaman ise zaman bağımlı optimal yörünge planlama, eğer optimizasyon kriteri enerji ise enerji bağımlı optimal yörünge planlama adını alır. Bu tez, genel hareket planlama problemlerini dikkate almasının yanı sıra özellikle zamana bağımlı optimal robot hareket planlaması ve zaman-enerji bağımlı optimal robot hareket planlaması problemlerine dayanmaktadır. Genellikle verilen ilk ve son durumlar arasındaki hareket konum ve hız açısından optimize edilir. Buna noktadan noktaya hareket planlama denir. Diğer durumlarda ara hareket öngürülmüş yörüngeyi içinden veya yanından geçerek veya yörünge noktalarının sayısı ile kısıtlandırılır [1], [2]. Kinematik robot hareket planlama problemleri eyleyici limitinin yerine, hız ve ivme gibi kinematik limitleri dikkate alan hareket planlama problemleridir. Hareket planlama problemleri, doğrusal olmayan robot dinamiği hesaplamaları ve eyleyici sınırlarını içerdiği için dinamik robot hareket planlama problemleri adını alır.

Dinamik robot hareket planlama problemleri tamamen doğrusal olmayan dinamik hesapları ve eyleyici sınırlarını içerir.

Robot hareket planlama problemlerini çözmek için olan metotlar doğrudan yaklaşım temelinde önerilmiştir [3], [4]. Bu yaklaşım çeşitli bilimsel çalışmalarda ve çeşitli yıllarda öne sürülmüştür [5], [6].

Optimal hareket planlama problemlerini çözmek için geliştirilen metotlar; dolaylı metotlar [6], [7], dinamik programlama ve ilgili araştırma metotları [5] ve doğrudan metotlar [8], [9] olmak üzere sınıflandırılır. Karmaşık robot sistemleri için, dolaylı yöntemler genellikle hesapsal olarak [10] elverişsiz veya sadece sınırlı değerler için

optimum şekilde [11] bir çözüm belirleyebilir. Doğrudan yöntemler doğrusal olmayan ve çözülmesi zor olan optimizasyon problemlerine yol açar [12].

Yörünge planlama, önceden tanımlanan geometrik yörüngeleri içerir. Bu tanımlanan yörüngeler göreve göre özel durumları, varsa engelden sakınmayı ve diğer yüksek dereceli genellikle geometrik şekilleri içerir. Engel varlığında planlama oldukça zordur ve kendiliğinden olan dışbükey olmayan problemidir. Bu problemler engellerin sayısı arttıkça daha karmaşık hale gelir [13], [14]. Planlama için olan metotlar; genel olarak parçalı eğrisel yörüngelerle [15], [16] veya dairesel eğrilere bağlı düz çizgilerle tanımlanabilir. Bu metotlar yörüngeyi optimize ederken arama metotları veya gradyan tabanlı optimizasyon metotlarını kullanarak yörüngeyi optimize eder [13].

Yörünge takibi, belirlenmiş öngörülen geometrik yolu bir zaman-optimal yörüngeye dönüştürür. Zaman optimal yörünge, yörünge planlayıcı ya da özel görevler tarafından oluşturulan yörüngeleri robot dinamiği ve eyleyici sınırların hesaba katarak oluşturulur [17]. Optimal robot yörünge takibi problemleri robotik manipulatörün konumlarını kısıtlayan, doğrusal olmayan eşitlik kısıtlamalarına sahip optimal kontrol problemleri olarak formüle edilir [18]. Optimum robot yörünge hesaplamaları yapan yöntemler eyleyici kısıtlamalarıyla birlikte ileri sürülmüştür [19], [20]. Bu yöntemler doğrudan yöntemler [17], [21], dinamik programlama ve dolaylı yöntemler [22], [23] olarak sınıflandırılırlar. Genel yörünge izleme yöntemleri tek yörünge koordinatları ve onun zamana göre türevi tarafından önceden belirlenmiş yol boyunca olan hareketleri kullanır ve bu nedenle robotik manipulatörlerinin durum uzayı iki boyutlu veya yörünge koordinatları sözde zamana göre belirtilmiş ise tek boyutlu durum uzayına bile indirgenebilir [21], [24]. Dolaylı yöntemler birçok optimal yörünge kontrolü çalışmalarında ileri sürülmüştür [24]. Bu yöntemler ilerleyen zamanlarda iyileştirilmiştir [25]. Dinamik programlama yöntemleri dolaylı yöntemler ile karşılaştırıldığında amaç fonksiyonunun ve eyleyici torklarına ait sınırlarının daha genel olarak tanımlanmasına elverişlidir. Dolaylı yöntemler amaç fonksiyonlarında zaman veya zaman-enerji optimizasyonu dikkate almalarına rağmen daha iyi geliştirilmiştir [26]. Dinamik programlama metotları ve arama yöntemleri iki boyutlu durum uzayından dolayı yörünge takibi için kolay işlenirdir. Dolaylı yöntemlerle karşılaştırıldığında, dinamik programlama ve arama yöntemleri [27] hesapsal olarak daha fazla işlem içermesine rağmen daha genel objektif fonksiyonlara ve daha genel

kısıtlamalara izin verir. Doğrudan yöntemlerde dinamik programlama metodunda olduğu gibi genel amaç fonksiyonu ve genel kısıtların tanımlanması kolaydır [28], [29]. Ancak sonlu-boyutsal doğrusal olmayan optimizasyon problemlerinin çözülmesi zor olabilir. Doğrusal yöntemler, konveks doğrusal olmayan optimizasyon problemleri olması durumunda sadece genel optimal çözümlerin bulunduğunu gösterir.

Kinematik robot hareket planlama yöntemleri, dinamik robot hareket planlama yöntemlerinden farklı olarak, doğrusal olmayan robot dinamiğini hesaba katmaz veya basitleştirerek sadece optimal zamana yaklaşan çözümleri elde edilmesine imkan verir. Ancak çoğu robot dinamik hareket planlama algoritmaları çevrimdışı yöntemler iken, kinematik hareket planlama yöntemleri sıklıkla çevrimiçi kullanılmaya uygundur [30], [31]. Ayrıca çoğu kinematik hareket planlama yöntemleri genellikle belirlenmiş yörüngeleri dikkate alır. Bu nedenle, genel noktadan noktaya hareket planlama ve yörünge takibi yerine tipik olarak düz-çizgi noktadan noktaya hareket planlama [32], [33] veya düz-çizgi yolu nokta hareket planlaması [2], [34] gibi daha özel durumlar dikkate alınır.

1.3 Hipotez

Kinematik olarak noktadan noktaya hareket planlaması için geliştirilen yöntemlerden birçoğu uygulama ya da eklem uzayındaki düz çizgi, uygulama uzay koordinatları arasındaki iki noktayı veya birbirine bağlı eğriler tarafından belirlenen daha genel eğriler ile tanımlanır [35].

Eklem uzaylarındaki doğrusal interpolasyon ile noktadan noktaya hareket planlama kolaydır, oysaki uygulama uzay koordinatlarındaki düz-çizgi noktadan noktaya hareket planlama daha karmaşıktır [32], [36]. Bu konuda yapılan ilk çalışmalar, eklem koordinatları içinde yörünge süresi verilmiş ve basit sabit hız profili ile tanımlanmış düz çizgi boyunca olan hareketlerle başlamıştır. Düz-çizgi hareket planlama için olan bu yöntemler esas olarak kinematik durumlara odaklanır. Daha güncel yapılan çalışmalarda yöntemler, karmaşık hız profillerini ifade eder ve doğrudan çevrimdışı zaman-optimal noktadan noktaya hareketlerin hızı ivme ve jerk kısıtlamaları da içine alarak hesaplama yapar [30], [37].

İki nokta arasındaki yörünge parçalı eğriler ve genel eğriler tarafından da tanımlanabilir. Düz çizgi noktadan noktaya hareket planlamasıyla karşılaştırıldığında,

eğrilere dayalı noktadan noktaya hareket planlaması daha esnektir ve optimum zamana ulaşılması için daha fazla imkan sağlar. İki nokta arasındaki hareketi eğri tabanlı noktadan noktaya hareket planlamasını kullanmak normalize edilmiş eğriler esas olarak kinematik çözüm temeline dayanmaktadır [38], [39]. Ayrıca, hız profili eğri tanımından yörüngeyi izlerken yörünge süresi hesaba katılmalıdır. Konu ile ilgili yapılan çevrimdışı optimal yörünge bulunması çalışmasında eksen uzay eğrileri orta eksen uzay düğümleri ile kurgulanmış ve hız, ivme ve jerk limitleri hakkında yörünge süresini minimize etmek için zaman ölçekleme kullanılmıştır [40].İlerleyen yıllarda yapılan başka bir çalışmada ise hız ve ivme kısıtları dikkate alınarak çevrimiçi optimizasyon yapılmıştır [41].

Öngörölmüş yörüngeler boyunca hareket için, kinematik hareket planlama yöntemleri; yörünge boyunca yaklaşık hareketi ya düz bir çizgi hareketi ya da daha genel olan eğri temelli hareketler tarafından çözüme ulaştırılmıştır.

Doğrusal yörünge yoluna sahip nokta hareket planlamasında, öngörölmüş yörünge boyunca hareket düz çizgilerin noktadan noktaya hareketinin serisi şeklinde tanımlanmıştır. Bu yaklaşım hızın devamlılığını sağlamak için her bir yol noktasında durdurmayı gerektiren düzgün olmayan bir yörünge yaklaşımının bir sonucudur. Her bir yol noktasında durmaktan kaçınmak için, düz çizgi noktadan noktaya hareket planlaması yöntemleri geliştirilmiştir. Düzgün yörünge elde edilmesi amaçlanan bu yaklaşımda eklem uzayında veya uygulama uzayındaki koordinatları karışım veya köşeleme algoritmaları geliştirilmiştir [42], [43]. Yol noktasının içinden ya da yakınından geçen polinomal eğrilerle yol noktaları arasındaki düz çizgilerin katılımıyla, bu karıştırma ve köşeleme algoritmaları ardaşık düz çizgilerin arasındaki düzgün hareket geçişini gerçekleştirir [34]. Çoğu yol nokta hareket planlaması yöntemleri çevrimiçidir ve basit hız ve ivme kısıtları ile çalışır, çevrimdışı olanları ve robot dinamiğinden gelen ivme limitlerini hesaba katar [44]. Sıralı şekilde tanımlanan düz çizgi yörüngelerin spesifikasyonları optimizasyon için yetersizdir. Bu nedenle, bu yöntemler esas olarak maksimum hız ve ivme limitinden sorumlu olan yakın yol noktası ve hız profilleri arasındaki zamana bağlı optimizasyona yakınsamayı sağlar.

Yol noktaları arasında doğrusal interpolasyon yerine, yörünge eğrileri gibi daha genel eğimler tarafından tanımlanan eğri temelli hareket planlamasıdır. Düz çizgilere dayalı

yol nokta hareket planlamasıyla karşılaştırıldığında, eğri temelli hareket planlaması daha esnektir ve zaman bağılı optimum değere ulaşması daha kolaydır.

Çoğu robot hareket planlaması yöntemleri robot sistemlerin eyleyicilerinin dinamik hesaplamalarını ihmal eder. Akım kontrollü aktüatorlerin veya gerilim kontrollü eyleyicilerdeki sürtünmenin kontrolü bu sistemlere örnek olarak verilebilir. Aktüator dinamik hesaplamalarını ihmal eden kendini sürekli güncelleyem zaman optimal robot hareket planlama yöntemleri her bir zamandaki doyurulmuş (hız, ivme, jerk limitlerine ulaşmış) en az bir aktüator aracılığı ile ve aktüator torkları açma kapama kontrolü veya tekil açma kapama kontrolü ile çözüm üretir [45]. Benzer şekilde sarsıntının (jerk) hesaba katılmadığı kinematik zaman optimal robot hareket yöntemleri ivmelerde ani değişimler söz konusu olacaktır. Uygulamada zaman optimal robot hareketlerinin ani sıçramalara maruz kalması aktüatorler için çok fazla yorucu olabilir ve aktüatorlerin torklarının sınırlarda olması sonucunda hassas olmayan izleme davranışı, önemli ölçüde yapısal vibrasyonlar ve artan aşınmalar olabilir [46], [47]. Bu nedenle zaman optimal robot hareketleri veya zaman optimal hareket planlama yöntemleri kullanıldığı zaman sistemin gözlenmesine uygulanabilirlik için sıklıkla ihtiyaç duyulur.

Zaman optimal hareketleri modifiye eden yöntemler ya sisteme önfiltre eklenmesiyle ya da zaman ölçekleme yöntemlerini kullanarak istenmeyen durumların oluşmasına engel olur. Optimal hareketin geliştirilmesi için geribildirim dinamikleri yörünge önfiltreleri kullanılarak dengelenirken, eyleyici dinamikleri basit sürtünme modelleri kullanarak dengelenmiştir [48]. Yapılan bazı çalışmalarda, aktüator torklarında geribeslemeli kontrol ile yeterince geniş aralıkta çevrimiçi zaman ölçekleme kullanılmasıyla robot dinamiği modelleme hataları ve modellenmemiş aktüator dinamiğinden kaynaklanan hatalar telafi edilmiş olur [49], [50]. Zaman optimal hareketlerini hesaplayan yöntemlerin genellikle hayata geçirilmesi kolaydır ve hareket planlama yöntemlerinin değiştirilmesine ihtiyaç yoktur.

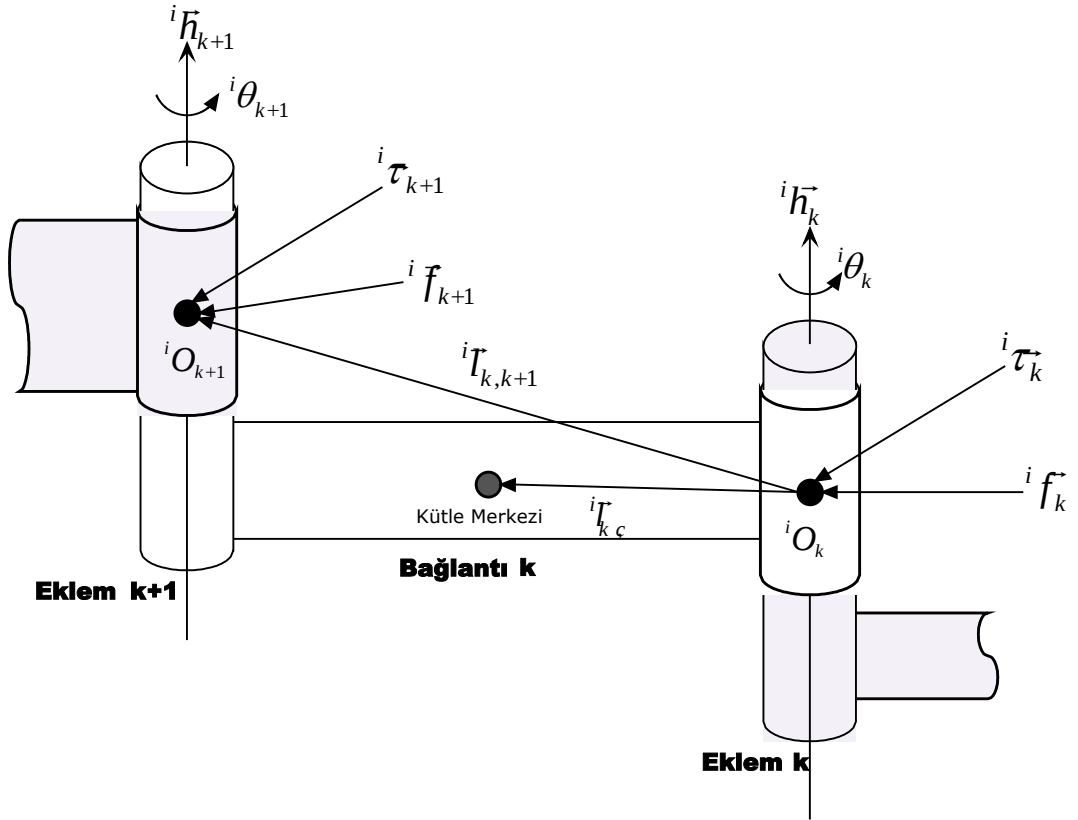
Zaman optimal hareket planlama problemlerinin değişimlerini içeren yöntemler amaç fonksiyonlarında enerji gibi ek koşullarla birleşir ya da aktüator dinamiklerini modeller [51], [52]. Zaman optimal hareket planlama metotları değiştirebilir olmasına rağmen, eğer hareket planlama çevrimdışı yerine getirilirse, bu yöntemler çevrimiçi yapılan hesaplamanın işlem yükünün arttırmaz. Ek terimlerin amaç fonksiyonlarına dahil edilen bu birleştirme genellikle anlaşılması kolay ve sık sık sadece basit

değişimleri gerektir. Bununla birlikte ek terimlerin birleştirilmesi karşılaştırıldığında, sonraki diğer yaklaşımlar genellikle daha direk ve etkilidir.

2. UZAYSAL OPERATÖR CEBİRİ İLE KİNEMATİK VE DİNAMİK MODELLEME

2.1 Uzaysal Operatör Cebiri (UOA)

Uzaysal Operatör Cebiri (UOC) çok serbestlik dereceli robot manipülatörlerin benzetimi için yüksek performanslı bir algoritmadır. UOC yöntemi koordinatları olmayan bir vektör notasyonu kullanılarak hesapların özyinelemeli olarak yapıldığı bir metod biçiminde çalışır. Bu yöntemin kullanılması için, bazı terimleri açıklanması gerekir. Üç boyutlu uzayda vektörler $\vec{x}$ ile belirtilir. Uzaysal koordinatlarda 6 serbestlik dereceli sistem $\vec{x}$ temsil edilir. Bunun dışında kalan vektörler $\underline{x}$ olarak temsil edilmiştir.



Şekil 2.1 : Manipülatör i'ye ait bağlantı k ile ilişkili vektörler.

Bir rijit bağlantı Şekil 2.1'de gösterilmiştir. I. manipülatör k. bağlantı açısal ve çizgisel hızlar ${}^i\vec{\omega}_k$ ve ${}^i\vec{v}_k$ ile sırasıyla temsil edilmektedir. Bunlar aşağıdaki gibi bir döner eklem için bağlantı k ve bağlantı k-1 aşağıda belirtildiği gibi yayılım göstermektedir:

$${}^i\vec{\omega}_k = {}^i\vec{\omega}_{k-1} + {}^i\vec{h}_k {}^i\dot{\theta}_k \quad (2.1)$$

$$\begin{aligned} {}^i\vec{v}_k &= {}^i\vec{v}_{k-1} + {}^i\vec{\omega}_{k-1} \times {}^i\vec{\ell}_{k-1,k} = {}^i\vec{v}_{k-1} - {}^i\vec{\ell}_{k-1,k} \times {}^i\vec{\omega}_{k-1} \\ &= {}^i\vec{v}_{k-1} - {}^i\hat{\ell}_{k-1,k} {}^i\vec{\omega}_{k-1} \end{aligned} \quad (2.2)$$

${}^i\vec{h}_k$ eklem k'ya ait olan dönme eksen vektörü ve ${}^i\hat{\ell}_{k-1,k} \triangleq ({}^i\vec{\ell}_{k-1,k} \times)$ ters bakışımli matristir. Matris içeriği aşağıda belirtildiği gibidir:

$${}^i\hat{\ell}_{k-1,k} = \begin{bmatrix} 0 & -{}^i\vec{\ell}_{(k-1,k)_z} & {}^i\vec{\ell}_{(k-1,k)_y} \\ {}^i\vec{\ell}_{(k-1,k)_z} & 0 & -{}^i\vec{\ell}_{(k-1,k)_x} \\ -{}^i\vec{\ell}_{(k-1,k)_y} & {}^i\vec{\ell}_{(k-1,k)_x} & 0 \end{bmatrix} \quad (2.3)$$

(2.2) ve (2.3) denklemlerini birleştirirsek:

$$\begin{bmatrix} {}^i\vec{\omega}_k \\ {}^i\vec{v}_k \end{bmatrix} = \begin{bmatrix} {}^3I & {}^30 \\ -{}^i\hat{\ell}_{k-1,k} & {}^3I \end{bmatrix} \begin{bmatrix} {}^i\vec{\omega}_{k-1} \\ {}^i\vec{v}_{k-1} \end{bmatrix} + \begin{bmatrix} {}^i\vec{h}_k \\ \vec{0} \end{bmatrix} {}^i\dot{\theta}_k \quad (2.4)$$

k bağlantısına ait uzaysal hız vektörü:

$${}^i\vec{V}_k \triangleq \begin{bmatrix} {}^i\vec{\omega}_k \\ {}^i\vec{v}_k \end{bmatrix} \quad (2.5)$$

ekleme ait hız yayılım matrisi ise:

$${}^i\phi_{k,k-1} \triangleq \begin{bmatrix} {}^3I & {}^30 \\ -{}^i\hat{\ell}_{k-1,k} & {}^3I \end{bmatrix} \quad (2.6)$$

ile gösterilir.

Bu bilgilerin ışığında k eklemine ait uzaysal dönme vektörü ${}^i\vec{H}_k$ ile simgelenir:

$${}^i\vec{H}_k \triangleq \begin{bmatrix} {}^i\vec{h}_k \\ \vec{0} \end{bmatrix} \quad (2.1)$$

Denklem (2.1) dönel eklemler içindir. Eğer eklem ötelemeli ise ${}^i\vec{H}_k$ aşağıdaki şekilde gösterilir:

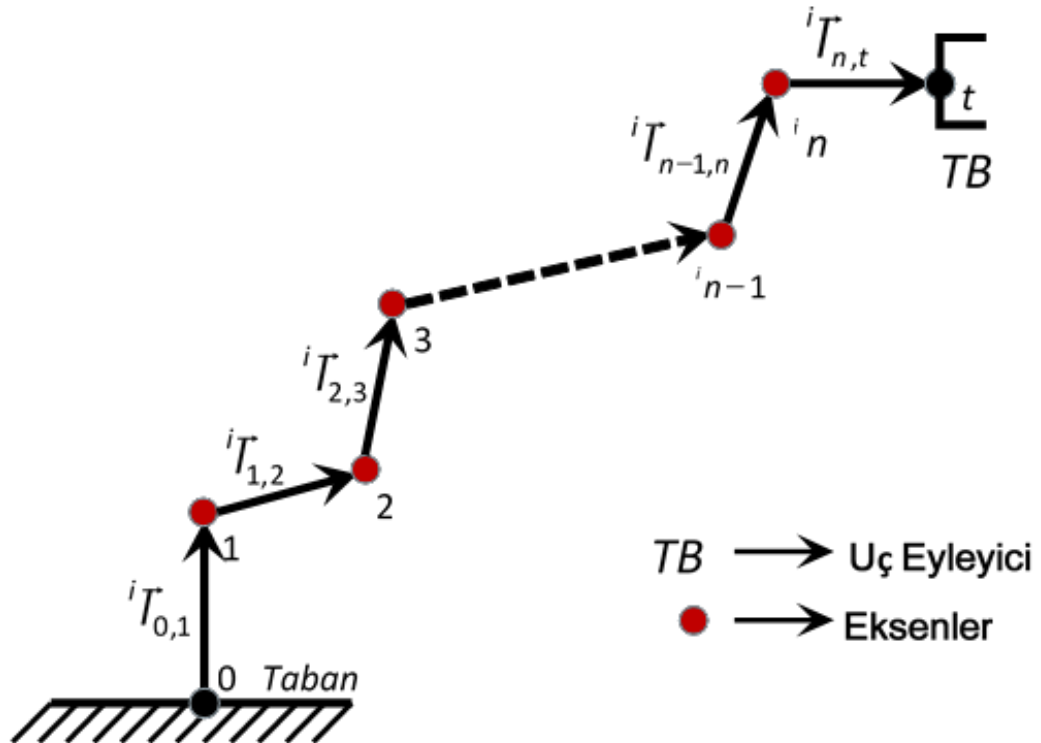
$${}^i\vec{H}_k \triangleq \begin{bmatrix} \vec{0} \\ {}^i\vec{h}_k \end{bmatrix} \quad (2.2)$$

Uzaysal hız denklemi, (2.1) ve (2.8) denklemleri cinsinden aşağıdaki şekilde ifade edilir:

$${}^i\vec{V}_k = {}^i\phi_{k,k-1} {}^i\vec{V}_{k-1} + {}^i\vec{H}_k {}^i\dot{\theta}_k \quad (2.3)$$

$(k - 1)$. bağlantıdan (k) . bağlantıya yayılıma ait uzaysal hız vektörü (açısal ve doğrusal hızlar) yukarıda gösterilen formülasyonda yukardaki gibi belirtilebilir.

2.2 Seri Manipülâtör Kinematığı



Şekil 2.2 : Sabit platformda seri manipulatör.

Seri robot topoloji sistemini temsil eden seri manipülatör kinematik olarak incelenmiş ve bu bölümde açıklanacaktır. Seri robotik manipülatöre ait her eklemin tabandan uç eyleyici noktasına kadar uzaysal hızların yayılımı sağlanacaktır. Kol için i notasyonu kullanılacak ve bu robot kolun sabit platformda olduğu kabulü ile işlemler yapılacaktır.

Şekil 2.2’de ifade edilen “ i_n ” i manipülatörüne ait n numaralı serbestlik derecesi anlamına gelmektedir. Uzaysal hızların yayılımı tabandan n numaralı serbestlik derecesine göre detaylandırılacaktır. Manipülatörün sabit platformda bağlı olması tabandaki açısal ve çizgisel hızların sıfır olduğu anlamına gelir.

$$\begin{aligned}
{}^i\vec{V}_0 &= \vec{0} \\
{}^i\vec{V}_1 &= {}^i\phi_{1,0} {}^i\vec{V}_0 + {}^i\vec{H}_1 {}^i\dot{\theta}_1 \\
{}^i\vec{V}_2 &= {}^i\phi_{2,1} {}^i\vec{V}_1 + {}^i\vec{H}_2 {}^i\dot{\theta}_2 \\
&\vdots \\
{}^i\vec{V}_{i_n} &= {}^i\phi_{i_n, i_{n-1}} {}^i\vec{V}_{i_{n-1}} + {}^i\vec{H}_{i_n} {}^i\dot{\theta}_{i_n}
\end{aligned} \tag{2.4}$$

Yayılım matrisinin durum geçiş özelliğinden faydalanarak, ${}^i\phi_{a,b} {}^i\phi_{b,c} = {}^i\phi_{a,c}$, (2.4)’da belirtilen matris aşağıdaki denklem yazılabilir:

$$\begin{aligned}
{}^i\vec{V}_0 &= \vec{0} \\
{}^i\vec{V}_1 &= {}^i\vec{H}_1 {}^i\dot{\theta}_1 \\
{}^i\vec{V}_2 &= {}^i\phi_{2,1} {}^i\vec{H}_1 {}^i\dot{\theta}_1 + {}^i\vec{H}_2 {}^i\dot{\theta}_2 \\
{}^i\vec{V}_3 &= {}^i\phi_{3,1} {}^i\vec{H}_1 {}^i\dot{\theta}_1 + {}^i\phi_{3,2} {}^i\vec{H}_2 {}^i\dot{\theta}_2 + {}^i\vec{H}_3 {}^i\dot{\theta}_3 \\
&\vdots \\
{}^i\vec{V}_{i_n} &= {}^i\phi_{i_n,1} {}^i\vec{H}_1 {}^i\dot{\theta}_1 + \dots + {}^i\phi_{i_n, i_{n-1}} {}^i\vec{H}_{i_{n-1}} {}^i\dot{\theta}_{i_{n-1}} + {}^i\vec{H}_{i_n} {}^i\dot{\theta}_{i_n}
\end{aligned} \tag{2.5}$$

(2.5)’de belirtilen denklemler düzenlenerek aşağıda belirtilen hız matrisleri yazılabilir:

$$\begin{bmatrix} \vec{iV}_1 \\ \vec{iV}_2 \\ \vdots \\ \vec{iV}_{i_n} \end{bmatrix} = \begin{bmatrix} {}^6I & {}^60 & \dots & {}^60 \\ {}^i\phi_{2,1} & {}^6I & \dots & {}^60 \\ \vdots & \vdots & \ddots & \vdots \\ {}^i\phi_{i_n,1} & {}^i\phi_{i_n,2} & \dots & {}^6I \end{bmatrix} \begin{bmatrix} \vec{iH}_1 & \vec{0} & \dots & \vec{0} \\ \vec{0} & \vec{iH}_2 & \dots & \vec{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \dots & \vec{iH}_{i_n} \end{bmatrix} \begin{bmatrix} {}^i\dot{\theta}_1 \\ {}^i\dot{\theta}_2 \\ \vdots \\ {}^i\dot{\theta}_{i_n} \end{bmatrix} \quad (2.6)$$

$$\underline{{}^iV} = \begin{bmatrix} \vec{iV}_1 \\ \vec{iV}_2 \\ \vdots \\ \vec{iV}_{i_n} \end{bmatrix} \quad {}^i\phi = \begin{bmatrix} {}^6I & {}^60 & \dots & {}^60 \\ {}^i\phi_{2,1} & {}^6I & \dots & {}^60 \\ \vdots & \vdots & \ddots & \vdots \\ {}^i\phi_{i_n,1} & {}^i\phi_{i_n,2} & \dots & {}^6I \end{bmatrix} \quad (2.7)$$

$${}^iH = \begin{bmatrix} \vec{iH}_1 & \vec{0} & \dots & \vec{0} \\ \vec{0} & \vec{iH}_2 & \dots & \vec{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \dots & \vec{iH}_{i_n} \end{bmatrix} \quad {}^i\dot{\theta} = \begin{bmatrix} {}^i\dot{\theta}_1 \\ {}^i\dot{\theta}_2 \\ \vdots \\ {}^i\dot{\theta}_{i_n} \end{bmatrix}$$

(2.6) ve (2.7)'deki denklemler aşağıda belirtilen formda sadeleştirilebilir.

$$\underline{{}^iV} = {}^i\phi \quad {}^iH \quad {}^i\dot{\theta} \quad (2.8)$$

Denklem (2.8) ile belirtilen robota ait uzaysal yayılım matrislerinin çıkarımı gösterilmektedir.

Bağlantı eklemi i_n 'e ait uzaysal hızların uç noktaya yayılımı basit şekilde ifade edilebilir.

$$\begin{aligned} {}^i\vec{\omega}_t &= {}^i\vec{\omega}_{i_n} \\ {}^i\vec{v}_t &= {}^i\vec{v}_{i_n} + {}^i\vec{\omega}_{i_n} \times {}^i\vec{\ell}_{i_n,t} = {}^i\vec{v}_{i_n} - {}^i\hat{\ell}_{i_n,t} \quad {}^i\vec{\omega}_{i_n} \end{aligned} \quad (2.9)$$

Denklem (2.9)'deki system matris formuna getirilirse:

$$\begin{bmatrix} {}^i\vec{\omega}_t \\ {}^i\vec{v}_t \end{bmatrix} = \begin{bmatrix} {}^3I & {}^30 \\ -{}^i\hat{\ell}_{i_n,t} & {}^3I \end{bmatrix} \begin{bmatrix} {}^i\vec{\omega}_{i_n} \\ {}^i\vec{v}_{i_n} \end{bmatrix} \quad (2.10)$$

Yayılm operatörü ${}^i\phi_{t, i_n}$ matris formunda gösterimi:

$${}^i\phi_{t, i_n} = \begin{bmatrix} {}^3I & {}^30 \\ -{}^i\hat{p}_{i_n, t} & {}^3I \end{bmatrix} \quad (2.11)$$

Denklem (2.10)'daki sistem aşağıda belirtilen formda yazılıabilir:

$${}^i\vec{V}_t = {}^i\phi_{t, i_n} {}^i\vec{V}_{i_n} \quad (2.12)$$

Denklem (2.12) 'deki ${}^i\vec{V}_{i_n}$ birimi ${}^i\underline{V}$ cinsinden yazılırsa:

$${}^i\vec{V}_t = {}^i\phi_t {}^i\underline{V} \quad (2.13)$$

Yayılmının ifadesi:

$${}^i\phi_t = \begin{bmatrix} {}^60 & {}^60 & \cdots & {}^i\phi_{t, i_n} \end{bmatrix} \quad (2.14)$$

(2.8) ve (2.13)'daki denklemlerden çıkarılan:

$${}^i\vec{V}_t = {}^i\phi_t {}^i\phi {}^iH {}^i\dot{\underline{\theta}} \quad (2.15)$$

Bu denklemlerden Jakobyen operatörü aşağıdaki gibi yazılır.

$${}^iJ \triangleq {}^i\phi_t {}^i\phi {}^iH \quad (2.16)$$

i Numarası ile belirtilen robota ait Jakobyen operatörü iJ ile gösterilir. Jakobyen operatör eklem uzayından çalışma uzayına geçiş sağlayan doğrusal operatördür. Denklem (2.15) ve (2.16) kullanılmasıyla uç noktadaki hız vektörü aşağıdaki gibi yazılabilir:

$${}^i\vec{V}_t = {}^iJ {}^i\dot{\underline{\theta}} \quad (2.17)$$

Eğer eksen uzayında olan hızlar biliniyorsa Jakobyen operatörü kullanılarak çalışma uzayındaki hızlar bulunabilir. Başka bir deyişle uç noktadaki açısal ve çizgisel hızlar eksen hızların kullanılarak hesaplanabilir.

Altı boyutlu uzaysal sistemde hızları açısal ve doğrusal olarak aşağıdaki şekilde yazabiliriz:

$$\begin{bmatrix} {}^i\vec{\omega}_t \\ {}^i\vec{v}_t \end{bmatrix} = \begin{bmatrix} {}^iJ_\omega \\ {}^iJ_v \end{bmatrix} {}^i\dot{\underline{\theta}} \quad (2.18)$$

Bu denklemde ${}^iJ_\omega$ ve iJ_v simgeleri $3 \times n$ boyutunda matrisleri tanımlamak için kullanılır. Burada n serbestlik derecesini belirtir.

$$\begin{aligned} {}^i\vec{\omega}_t &= {}^iJ_\omega {}^i\dot{\underline{\theta}} \\ {}^i\vec{v}_t &= {}^iJ_v {}^i\dot{\underline{\theta}} \end{aligned} \quad (2.19)$$

Bu tanımlama ile açısal ve çizgisel hızların eksen hızlarından kaynaklanan genel formülasyonu görülmektedir.

Ters kinematik, robotun uç noktasının uzaysal hızından eksen hızlarına geçiş yapılması anlamına gelmektedir. Jakobyen matrisinin tam rank olarak hesaplanabilmesi için jakobyen matrisinin sözde (pseudo) tersi olarak ifade edilen matris hesaplanması gerekmektedir.

$${}^i\dot{\underline{\theta}} = {}^iJ^\# {}^i\vec{V}_t \quad (2.20)$$

${}^iJ^\#$ Terimi jakobyen matrisinin sahte tersinin alınmış halidir.

Jakobyen matrisinin tersinin doğrudan hesaplanabilmesi için tam rankının olması ve kare matris olması gerekmektedir. Jakobyen matrisinin sahte tersinin alınması serbestlik derecesinin ne olduğuna bakılmaksızın çözüm oluşturulmasını sağlar. Aksi takdirde matrise jakobyen matrisi her zaman kare olmayacağı için tersi alınmasında mümkün olmamaktadır.

2.3 Seri Manipülatör Dinamiği

Dinamik analiz yapılabilmesi için açısal ve çizgisel hızların hesaplanması gerekmektedir.

$${}^i\dot{\vec{\omega}}_k = {}^i\dot{\vec{\omega}}_{k-1} + {}^i\vec{h}_k {}^i\ddot{\theta}_k + {}^i\vec{\omega}_{k-1} \times {}^i\vec{\omega}_k \quad (2.21)$$

$${}^i\dot{\vec{v}}_k = {}^i\dot{\vec{v}}_{k-1} - {}^i\vec{l}_{k-1,k} \times {}^i\dot{\vec{\omega}}_{k-1} + {}^i\vec{\omega}_{k-1} \times ({}^i\vec{\omega}_{k-1} \times {}^i\vec{l}_{k-1,k}) \quad (2.22)$$

(2.27) ve (2.28) kullanılarak

$${}^i\ddot{\vec{V}}_k = {}^i\phi_{k,k-1} {}^i\ddot{\vec{V}}_{k-1} + {}^i\vec{H}_k \ddot{\theta}_k + {}^i\vec{a}_k \quad (2.23)$$

i numaralı kola ait k eksenindeki uzaysal ivmelenme ${}^i\vec{a}_k$ ile gösterilir:

$${}^i\vec{a}_k = \begin{bmatrix} {}^i\vec{\omega}_{k-1} \times {}^i\vec{\omega}_k \\ {}^i\vec{\omega}_{k-1} \times ({}^i\vec{\omega}_{k-1} \times {}^i\vec{l}_{k-1,k}) \end{bmatrix} \quad (2.24)$$

$\underline{{}^i a}$ bütün eksenlere ait uzaysal ivmelenmelerin birleştirilmiş halidir:

$$\underline{{}^i a} = \begin{bmatrix} {}^i\vec{a}_1 \\ \vdots \\ {}^i\vec{a}_{n_i} \end{bmatrix} \quad (2.25)$$

i numaralı kola ait k numaralı eksene gelen uzaysal kuvvet ${}^i\vec{F}_k$ ile gösterilir:

$${}^i\vec{F}_k = {}^i\phi_{k+1,k}^T {}^i\vec{F}_{k+1} + {}^iM_k {}^i\ddot{\vec{V}}_k + {}^i\vec{b}_k \quad (2.26)$$

$${}^i\vec{F}_k = \begin{bmatrix} {}^i\vec{t}_k \\ {}^i\vec{f}_k \end{bmatrix} \quad (2.27)$$

${}^i\vec{t}_k$ eksen k 'ya ait olan tork vektörü ${}^i\vec{f}_k$ ise eksen k 'ya etkiyen kuvvet vektörüdür.

$$\underline{{}^i F} = \begin{bmatrix} {}^i\vec{F}_1 \\ \vdots \\ {}^i\vec{F}_{n_i} \end{bmatrix} \quad (2.28)$$

$\underline{{}^i F}$ bütün eksenlere etkiyen kuvvetlerin bir arada gösterilmiş halidir.

Eksenlere uygulanan torklar dönme eksenindeki kuvvetlerin etkisiyle oluşmaktadır.

$$\underline{{}^i \tau} = {}^iH^T \underline{{}^i F} \quad (2.29)$$

${}^i\mathfrak{S}_k$ k eklemine ait atalet matrisini ifade eder. i , Üç boyutlu uzayda her bir düzleme ait atalet matrisleri bu system içine eklenir.

$${}^i\mathfrak{S}_k = \begin{bmatrix} {}^i\mathfrak{S}_{xx} & {}^i\mathfrak{S}_{xy} & {}^i\mathfrak{S}_{xz} \\ {}^i\mathfrak{S}_{yx} & {}^i\mathfrak{S}_{yy} & {}^i\mathfrak{S}_{yz} \\ {}^i\mathfrak{S}_{zx} & {}^i\mathfrak{S}_{zy} & {}^i\mathfrak{S}_{zz} \end{bmatrix} \quad (2.30)$$

Bir ekleme ait atalet denklemleri:

$$\begin{aligned} {}^i\mathfrak{S}_{xx} &= \iiint (y^2 + z^2) \rho(x, y, z) dx dy dz \\ {}^i\mathfrak{S}_{yy} &= \iiint (x^2 + z^2) \rho(x, y, z) dx dy dz \\ {}^i\mathfrak{S}_{zz} &= \iiint (x^2 + y^2) \rho(x, y, z) dx dy dz \\ {}^i\mathfrak{S}_{xy} &= {}^i\mathfrak{S}_{yx} = - \iiint xy \rho(x, y, z) dx dy dz \\ {}^i\mathfrak{S}_{xz} &= {}^i\mathfrak{S}_{zx} = - \iiint xz \rho(x, y, z) dx dy dz \\ {}^i\mathfrak{S}_{yz} &= {}^i\mathfrak{S}_{zy} = - \iiint yz \rho(x, y, z) dx dy dz \end{aligned} \quad (2.31)$$

Yukarıda hesaplanan denklemler rijit gövdenin farklı eksenlere göre hesaplanmış atalet momentleridir. ${}^i\mathfrak{S}_{xx}$, ${}^i\mathfrak{S}_{yy}$, ${}^i\mathfrak{S}_{zz}$ terimleri x, y, z eksenlerinde temel atalet momentlerini ifade eder. ${}^i\mathfrak{S}_{xy}$, ${}^i\mathfrak{S}_{xz}$ gibi diğer terimler vektörel çarpımları ifade eder. Eğer eksenlere göre kütle dağılımı simetrik ise vektörel çarpım terimleri ${}^i\mathfrak{S}_{xy}$ ${}^i\mathfrak{S}_{yz}$ ve diğerleri sıfırdır.

iM_k k eklemine ait kütle matrisidir:

$${}^iM_k = \begin{bmatrix} {}^i\mathfrak{S}_k & {}^i m_k {}^iL_{k,c} \\ - {}^i m_k {}^iL_{k,c} & {}_3I {}^i m_k \end{bmatrix} \quad (3.32)$$

I (3x3) boyutlarında birim matrisidir. m_k k eklemine ait kütle vektörüdür. $\mathfrak{S}_k$ k eklemine ait atalet matrisidir.

iM bütün eklemlere ait kütle matrisi

$${}^iM = \begin{bmatrix} {}^iM_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & {}^iM_{n_i} \end{bmatrix} \quad (3.33)$$

$\vec{{}^ib_k}$ k eklemine ait uzaysal kuvvetleri:

$$\vec{{}^ib_k} = \begin{bmatrix} {}^i\vec{\omega}_k \times {}^i\mathfrak{S}_k {}^i\vec{\omega}_k \\ {}^im_k {}^i\vec{\omega}_k \times ({}^i\vec{\omega}_k \times {}^i\vec{l}_{k,c}) \end{bmatrix} \quad (3.34)$$

$\vec{l}_{k,c}$ orjinden k ekseninin kütle merkezine ait eklem vektörüdür. $\underline{{}^ib}$ bütün eklemlere gelen kuvvet vektörlerinin bir arada olduğu matrisdir:

$$\underline{{}^ib} = \begin{bmatrix} \vec{{}^ib_1} \\ \vdots \\ \vec{{}^ib_{n_i}} \end{bmatrix} \quad (3.35)$$

Yukarıda belirtilen bütün denklemlerde:

$$\underline{{}^i\tau} = {}^i\mathcal{M} {}^i\ddot{\theta} + {}^i\underline{C} + {}^i\mathcal{M}_b \vec{{}^iV}_b + {}^iJ^T {}^i\vec{F}_t \quad (3.36)$$

${}^i\mathcal{M}$ i numaralı kola ait genelleştirilmiş kütle matrisidir:

$${}^i\mathcal{M} = {}^iH^T {}^i\Phi^T {}^iM {}^i\Phi {}^iH \quad (3.37)$$

${}^i\underline{C}$ terimi Coriolis ve yer çekimini ifade etmektedir:

$$\underline{{}^iC} = {}^iH^T {}^i\Phi^T ({}^iM {}^i\Phi \underline{{}^ia} + \underline{{}^ib}) \quad (3.38)$$

${}^i\mathcal{M}_b$ tabanla dinamik etkileşimi içeren kütle matrisidir:

$${}^i\mathcal{M}_b = {}^iH^T {}^i\Phi^T {}^iM {}^i\Phi {}^i\Phi_b \quad (3.39)$$

Bu bölümde endüstriyel robotlarda optimal yörünge planlaması yapılırken kullanılacak olan kinematik ve dinamik denklemler anlatılmıştır. Endüstriyel manipülatörler genel olarak sabit platformda kullanılırlar. Doğrusal olmayan problemin kolaylaştırılması için sistemde önerilen robotun sabit platformda olduğu kabul

edilmiştir. ${}^i\underline{C}$, ${}^i\mathcal{M}$ terimleri tek bir platform için sıfıra eşittir. Platform sabitlenmiş olduğu içinde ${}^i\vec{V}_b$ terimi sıfır matrisidir.

Bu tezde 3. bölümde, optimal hareket planlama ve optimal yörünge takibi üzerinde yapılan çalışmalar anlatılacaktır.

3. OPTİMAL HAREKET PLANLAMA

3.1 Giriş

Optimal hareket planlama robotların çalışmasında çok önemli bir yere sahiptir. Optimal hareket planlamada genel amaç belirlenen optimizasyon kriteri ışığında başlangıç bitişe kadar olan yörüngenin oluşturulmasıdır. Optimizasyon kriterleri gidilen mesafenin en aza indirilmesi, sarsmanın en aza indirilmesi, gidilecek zamanın en aza indirilmesi, harcanan enerjinin en aza indirilmesi, engelden sakınma, kinematik ve dinamik limitlerde çalışma olarak ayrılabilir. Bu kriterlere ek olarak eklem uzayında ya da çalışma uzayında olması farklı yaklaşımlar getirmektedir. Eklem uzayında her bir eklem için yörüngeler tek tek belirlenir. Çalışma uzayı ise sadece uç eyleyicinin pozisyonu ile ilgilenir. Çalışma uzayında belirlenen yörüngelerin tekil noktalardan geçmemesi gerekir. Her bir dönme eksenini kendi hız ve ivme sınırlarını aşmamalıdır. Optimal yörünge planlamasında optimizasyon kriterleri yanında hız, ivme ve tork gibi kısıtlar önemlidir. Yörüngenin belirlenmesinde iki ayrı yoldan gidilebilir: Kinematik yörünge planlaması ve Dinamik yörünge planlaması. Kinematik yörünge planlaması kütle, atalet, kuvvet, tork gibi değerlerle ilgilenmez. Engelden sakınma optimizasyonu için çeşitli algortimalar kullanılmaktadır. Gidilen zamanın ve tüketilen enerjinin en aza indirilmesi için adaptif genetik algortima, ikinci dereceden konik programlama, parçacık sürü optimizasyonu benzeri teknikler kullanılabilir.

3.2 Optimal Hareket Planlama Yöntemleri

3.2.1 Optimizasyon metotları

Optimizasyon bir süreç ya da fonksiyondaki girişleri ve parametreleri ayarlayarak bazı çıktıların en az ya da en çok olması için algoritmalar kullanarak sonuç bulmaya çalışır. Burdaki süreç ya da fonksiyona maliyet fonksiyonu, amaç fonksiyonu ya da değerlendirme fonksiyonu adı verilir. İstenen değerin en küçük ya da en büyük olması gerektiğinin belirlendiği yer maliyet fonksiyonudur. Bir fonksiyona ait tüm değerlerin

araştırılması ve en küçüğünün bulunması optimizasyon problemlerinin temelini oluşturur. Araştırmada bulunan en az ya da maksimum değerin yerel mi yoksa genel mi olduğunun bulunması ayrı bir durumdur. Bazı yöntemler yerel bazı yöntemler global en az bulunması üzerine yoğunlaşmıştır.

3.2.1.1 Kapsamlı arama

Robot kolun hareket edebileceği bütün yolların çıkarılmasında kullanılabilir. Bu yöntem sadece sınırlı sayıda girdi bulunduğu zaman uygulanabilir. Global en az arandığı zaman sonuç vermesi uzun zaman alır.

3.2.1.2 Analitik optimizasyon

Calculus yöntemi maliyet fonksiyonunun en az değerinin bulunması için fonksiyon tek bir değer etrafında toplanır. Uçdeğer, maliyet fonksiyonunun birinci türevinin alınıp sıfıra eşitlenmesiyle indirgenen tek bir değere göre bulunur. Eğer ikinci türev sıfırdan büyük ise uçdeğer en az aksi halde uç değer en fazla olacaktır. (Practical Genetic algortihms Randy L Haupt) Bu yöntem tek bir en az bulunması için hızlı bir yöntemdir fakat global en az değerinin bulunması için uzun araştırma gerektirir. Çok değerli fonksiyonlar için *Lagrange* tarafından bulunan Lagrange çarpanları uç değer kısıt denklemlerine göre bulunması için etkili bir yöntemdir.

$$\max f(x, y) = \max[f(x, y) + k(g(x, y) - c)] \quad (3.1)$$

Calculus yönteminin diğer yönteme göre dezavantajı sadece yerel değerlerle uğraşması ve birden fazla parametreye göre işlem yapamamasıdır.

3.2.1.3 Downhill simpleks metodu

Simpleks 1947'de G.B. Dantzip tarafından geliştirilen optimizasyonda kullanılan bir metotodur. Bu yöntemde türev alma işlemleri yoktur. Doğrusal programlamaya dayalı bu yöntem iterasyonlardan oluşur. Simpleks tablosu düzenlenerek belirlenmiş yöntemler ışığında optimal çözüme ulaşıncaya kadar işlemler tekrar edilir. Belirlenen tolerans aralığına gelinceye devam simpleks iterasyonları devam eder. Hesaplama hızı konusunda bu yöntem oldukça zayıftır. Sonuç bulmadaki kararlılığı dolayısıyla tercih edilebilir. Simpleks metodu geliştirilerek değiştirilmiş simpleks metodu (kompleks metot) adını almıştır. Eşitsizlik kısıtlarını bu yönteme eklemek mümkün olmaktadır.

3.2.1.4 Doğru arama metodu ile optimuma ulaşma

Optimizasyon metotları arasında en genel kullanıma sahip olan yöntem budur. Bu algoritmada maliyet fonksiyonu üzerinde rastgele bir nokta ve gidiş yönü seçilerek maliyet fonksiyonunun arttığı yere gidiş devam eder. Bir sonraki adımda ters yönde tekrarlanır. Yerel en az bulunabilmesi için denklem 3.2’de γ_n arama yönü ile belirlenen aralıklarda denklem hesaplanır.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \gamma_n \nabla f(x_n, y_n) \quad (3.2)$$

3.3 Konveks Optimizasyon

Konveks optimizasyon problemleri tek bir uç noktası (en az, en fazla) olan problemlerdir. Konveks olmayan yöntemlerde birden fazla uç nokta söz konusudur. Konveks optimizasyonda global en az ya da maksimum noktası kolaylıkla bulunabilirken eğer konveks olmayan optimizasyon problemi söz konusu ise çözüm yerel bir uç noktanın bulunmasıyla global uç noktaya ulaşmadan tamamlanabilir. Doğrusal amaç fonksiyonu, doğrusal eşitlik ve eşitsizliklere sahip olan bir optimal planlama problemi konvektir. Eğer doğrusal olmayan amaç fonksiyonu, doğrusal olmayan eşitlik ve eşitsizlikler söz konusu ise problem konveks yada konveks olmayan optimizasyon problemi olabilir. Bu sistemlerde konveksliğin araştırılması gerekmektedir. Bu tezde yörünge planlaması konveks optimizasyon metotlarından ikinci dereceden konveks optimizasyon yöntemi kullanılarak yapılacaktır.

3.4 Zaman Optimal Yörünge Planlama

Robotlarda yörünge planlanmasında zaman optimal yörünge planlaması önemli problemlerden biridir. Üretimde verimliliğin artırılması amacıyla kullanılan robotlarda verimliliğin artırılması hızın artırılması ile mümkün olacaktır. Başka bir deyişle bir işlemin yerine getirilmesi için gereken sürenin en aza indirilmesi en genel isteklerden birisidir. Amaç üretim zamanı en aza indirilerek elde edilen karın artırılmasıdır. Bu sebepten dolayı robotun kendi çalışma kapasitesinin en yüksek hızında çalışması istenir. Burda ortaya çıkan en kısa işlem zamanında robotun en yüksek hızda çalışırken işleme ait istekleri ve belirlenen sınır şartlarının göz önünde bulundurulmasıdır. Burda sınır şartlarından kast edilen robot mümkün olan en yüksek hızda işlemi

gerçekleştirdiğinde yörüngeden sapmalar ya da sistemde olan sarsılmalar, titreşimler operasyonun izin verdiği seviyeleri aşmamalıdır.

Robotlarda optimal yörünge hesaplanabilmesi için iki farklı yaklaşım söz konusudur: Kinematik ve dinamik yaklaşım. Eğer robotun yük kapasitesi sisteme uygulanan yüke çok fazla, sistemin kapasitesinin çok altında bir değer ile çalışma söz konusu ile kinematik yaklaşımın kullanılması işleri kolaylaştıracaktır. Kinematik yaklaşımda öncelikle istenilen yörünge noktalara bölünerek belirtilir. Burada eklemlere ait hız, ivme, jerk limitleri dikkate alınarak sistemin hızının en yüksek olacağı hareket ortaya çıkartılır. Diğer bir yaklaşım olan dinamik yaklaşımda ise sistemin kütlesi ve uygulanan kuvvetler ve tork limitleri göz önünde bulundurularak robotun yörüngesi optimal olarak hesaplanmaktadır.

Zaman optimal yörünge problemlerinde iki farklı yaklaşımla sonuca gitmek mümkün olacaktır. Doğrudan yaklaşımda hareket planlama problemlerini çözmek için sistem durum uzayı şeklinde yazılır ve çözüm aranır. Robot dinamiğinin doğrusal olmamasından dolayı bu yöntem oldukça karmaşıktır. Diğerisi ise birleştirilmiş yaklaşımdır. Yörünge planlaması yani kinematik yaklaşım için çözüm üretilir ve bu çözüm yörünge takibi olarak ifade edilen dinamik yaklaşım ile birleştirilir. Birleştirilmiş yöntemde hesaplama yükü kolaylaşır.

N serbestlik derecesine sahip robot manipülatörde eksen açıları q şeklinde ve eksenlere ait torklar τ şeklinde yazılabilir.

$${}^i\tau = {}^i\mathcal{M} {}^i\ddot{\theta} + {}^i\mathcal{C}\dot{\theta} + {}^iF_s {}^i\theta sgn(\dot{\theta}) + G(\theta) \quad (3.3)$$

Burada ${}^i\mathcal{M}$ kütle matrisi ve ${}^i\mathcal{C}$ Coriolis ve merkezci etkileri iF_s Coloumb sürtünme torklarını ve $G(q)$ ise yerçekimi ve eksen açılarına bağlı torkları ifade eder. Viskoz sürtünme kuvveti bu sistemde ihmal edilmiştir.

Eksen koordinatlarındaki zamana bağlı bir yörünge $q(t)$ skalar yörünge koordinatlarındaki $q(s)$ 'e bağlı bir yörünge haline dönüştürülmesi yörüngeyi zamandan bağımsız kılacaktır. 0'dan T'ye olan bir yörünge $s(0) = 0 \leq s(t) \leq 1 = s(T)$ şeklinde ifade edilebilir.

Burada zamandan bağımsız yörünge türevi alınarak eksen hızları ve yörüngeleri türev hesaplarındaki zincir kuralı uygulanarak (3.6)'da göüldüğü gibi yeniden çıkartılır.

$$\dot{\theta}(s) = \theta'(s)\dot{s} \quad (3.4)$$

$$\ddot{\theta}(s) = \theta'(s)\ddot{s} + \theta''(s)\dot{s}^2 \quad (3.5)$$

$$\dot{s} = \frac{ds}{dt} \quad \ddot{s} = \frac{d^2s}{dt^2} \quad \theta'(s) = \frac{\partial \theta(s)}{\partial s} \quad \theta''(s) = \frac{\partial^2 \theta(s)}{\partial s^2} \quad (3.6)$$

$${}^i\tau(s) = m(s)\ddot{s} + c(s)\dot{s} + g(s) \quad (3.7)$$

$$m(s) = {}^i\mathcal{M}(\theta(s))\theta'(s) \quad (3.8)$$

$$c(s) = {}^i\mathcal{M}(\theta(s))\theta'(s) + {}^i\mathcal{C}(\theta(s), \theta'(s))\theta'(s) \quad (3.9)$$

$$g(s) = {}^iF_s(\theta(s))\text{sgn}(\theta'(s)) + G(\theta(s)) \quad (3.10)$$

Robot manipulatorlarda zaman optimal yörünge izleme problemlerinde eksenlere ait torkların alt ve üst limitlerinin gösterimi aşağıdaki gibidir.

$$\min T$$

$$\tau(t) = m(s(t))\ddot{s}(t) + c(s(t))\dot{s}(t)^2 + g(s(t)) \quad (3.11)$$

$$s(0) = 0 \quad (3.12)$$

$$s(T) = 1 \quad (3.13)$$

$$\dot{s}(0) = \dot{s}_0 \quad (3.14)$$

$$\dot{s}(T) = \dot{s}_T \quad (3.15)$$

$$\dot{s}(t) \geq 0 \quad (3.16)$$

$$\underline{\tau}(s(t)) \leq \tau(t) \leq \bar{\tau}(s(t)), \quad t \in [0, T] \quad (3.17)$$

$\underline{\tau}$ eklemlere ait tork değerlerinin alt limitini ve $\bar{\tau}$ ise eklemlere ait değerlerinin üst limitlerini s' 'e bağlı olarak gösterir. Birçok durumda $\dot{s}_0$ ve $\dot{s}_T$ sıfıra eşit olduğu kabul edilebilir.

(3.11)-(3.17) arasındaki denklemlerden yola çıkarak herhangi bir yerel zaman optimal çözümün global çözüm olduğu öne sürülemez. Sebeplere dayanarak probleme ait yerel optimum çözümlerin global olarak optimum olduğu görülmektedir. Bu çalışmada problemlerin formülasyonunda farklı bir gösterim metodu uygulanacaktır. (Diederik vereshure Convex time optimal) gösteriminden faydalanarak yeni bir yaklaşım kullanıcaktır.

3.11 'de kullanılan amaç fonksiyonuna ait integrasyon değişkeninin t 'den s 'e dönüştürülerek yeniden düzenlenmesi;

$$T = \int_0^T 1 dt = \int_{s(0)}^{s(T)} \frac{1}{\dot{s}} ds = \int_0^1 \frac{1}{\dot{s}} ds \quad (3.18)$$

Sonrasında optimizasyon değişkenleri s 'e bağlı olarak yazılırsa;

$$a(s) = \ddot{s} \quad (3.19)$$

$$b(s) = \dot{s}^2 \quad (3.20)$$

$$b'(s) = 2a(s) \quad (3.21)$$

$$\dot{b}(s) = b'(s)\dot{s} \quad (3.22)$$

$$\dot{b}(s) = \frac{d(\dot{s})}{dt} = 2\dot{s} = 2a(s)\dot{s} \quad (3.23)$$

Bu dönüşümlerden faydalanarak amaç fonksiyonunun yeniden düzenlenmesiyle;

$$\min \int_0^1 \frac{1}{\sqrt{b(s)}} ds \quad (3.24)$$

$$\tau(s) = m(s)a(s) + c(s)b(s) + g(s) \quad (3.25)$$

$$b(0) = \dot{s}_0 \quad (3.26)$$

$$b(1) = \dot{s}_T \quad (3.27)$$

$$b'(s) = 2a(s) \quad (3.28)$$

$$b(s) \geq 0 \quad (3.29)$$

$$\underline{\tau}(s) \leq \tau(s) \leq \bar{\tau}(s), \quad s \in [0,1] \quad (3.30)$$

(3.24) ile (3.30) arasındaki denklemler elde edilir. Burada kullanılan değişkenler doğrudan optimizasyon değişkenleri olarak kullanılabilir. Burada problem ve sınır şartları konveks olarak tanımlanmıştır. (3.25) ile (3.30) arasındaki değişkenler doğrusaldır.

(3.24) ile (3.30) denklemleri ile tanımlanan problem optimal kontrol problemi olarak tanımlanabilir. Cebirsel katsayılı diferansiyel formda tanımlanan denklemde s zaman parameteresini, $a(s)$ kontrol girişini, $b(s)$ diferansiyeli, $\tau(s)$ cebirik durumu, (3.28) doğrusal sistem dinamiğini, (3.25), (3.29), (3.30) doğrusal bağımlı sınırlamaları ve (3.26), (3.27) ise başlangıç ve bitiş koşullarını ifade eder.

Yeniden tanımlama ile oluşturulan (3.24)- (3.30) denklemleri tek bir diferansiye duruma sahiptir. (3.25) denklemi kullanılarak cebirik durumlar kolaylıkla sadeleştirilebilir. Zaman denklemden açık olarak bulunmamaktadır. Bu sebepten dolayı (3.24)- (3.30) denklemlerinin konveks olduğu ve kolay bir amaç fonksiyonu ile belirlendiği görülmektedir. Ayrıca eşitsizlikler optimal kontrol probleminin konveksliğini bozmadan sisteme dahil edilebilir.

Sisteme termal enerji, hız, ivme ve tork değişim oranı kısıtlarının problemin konveksliğinin değiştirilmeden eklenmesi gerçekleştirilecektir.

3.5 Zaman Optimal Yörünge Problemin Genelleştirilmesi

3.5.1 Yeni amaç fonksiyonlarının oluşturulması

Zaman ve yörünge birbiriyle olan ilişkisi değerlendirilerek zaman optimal yörünge planlaması yapılmaktadır. Amaç fonksiyonu değiştirilerek enerji optimal yörünge planlaması, zaman-enerji optimal yörünge planlaması ve minimum sarsılmalı yörünge planlaması şeklinde yapılabilir.

3.5.2 Enerji optimal yörünge

Robotlarda tüketilen enerji hesaplaması en doğru olarak sistemden çekilen akımdan yola çıkılarak gerçekleştirilebilir. Fakat sistemin çektiği akım sürtünme dolayısıyla ısı ve enerji dönüşüm kayıplarını içerdiği için buradan yola çıkılarak yörünge planlaması yapmak mümkün olmayacaktır. En basit şekilde hesaplanabilir enerji optimal yörünge planlaması sistemin anlık torklarının toplamının hesaplanarak sisteme dahil edilmesidir.

Bir i eklemine ait değerinin integrali:

$$\int_0^T \tau_i(s)^2 dt = \int_0^1 \frac{\tau_i(s)^2}{\dot{s}} ds = \int_0^1 \frac{\tau_i(s)^2}{\sqrt{b(s)}} ds \quad (3.31)$$

Bir i eklemine ait tork değerinde değişimin mutlak değeri:

$$\int_0^T |\dot{\tau}_i(s)| dt = \int_0^1 \frac{|\tau'_i(s)\dot{s}|}{\dot{s}} ds = \int_0^1 |\tau'_i(s)| ds \quad (3.32)$$

$$\dot{s} \geq 0 \text{ for } t \in [0, T] \quad (3.33)$$

3.5.3 İvme, tork değişim oranı kısıtlarının denkleme eklenmesi

3.5.3.1 Hız kısıtı

Belirlenen uygulamalara göre optimal yörünge denklemlerine hız limitlerinin alt $-\bar{q}(s)$ ve üst $\bar{q}(s)$ limitleri her bir eklem için aşağıda belirtildiği şekilde eklenir.

$$-\bar{q}(s) \leq \dot{q}(s) \leq \bar{q}(s) \quad (3.34)$$

$$(\dot{q}(s))^2 = (q'_i(s)\dot{s})^2 = (q'_i(s))^2 b(s) \leq (\bar{q}(s))^2 \quad (3.35)$$

Eklem uzayından uygulama uzayına geçilip öteleme parametreleri (v_x, v_y, v_z) ve dönme parametreleri $(\omega_x, \omega_y, \omega_z)$ dikkate alınarak aynı kısıt fonksiyonu sisteme uygulandığında;

$$-\bar{v}(s) \leq v_x(s) \leq \bar{v}(s) \quad (3.36)$$

$$(v_x(s))^2 \leq (\bar{v}(s))^2 \quad (3.37)$$

Uygulama uzayındaki hız parametreleri eklem uzayındaki hız parametrelerine robotun Jakobyen matrisi ile tanımlarsak $i = 1 \dots 6$ ve $j = 1 \dots n$ olmak kaydıyla

$$J(q) = (J_{ij}(q)) \quad (3.38)$$

$$(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T = J(q)\dot{q} \quad (3.39)$$

(3.37)'deki denklem düzenlenip yeniden yazılırsa

$$\left(\sum_{i=1}^n J_{1i}(q(s)) q_i'(s) \right)^2 b(s) \leq (\bar{v}_x(s))^2 \quad (3.40)$$

(3.34)'den (3.40)'a kadar olan denklemlerden yola çıkarak

$$b(s) \leq \bar{b}(s) ; s \in [0,1] \quad (3.41)$$

3.5.3.2 İvme kısıtı

Hareket planlamada en önemli unsurlardan biri olan ivmenin optimizasyon kısıtlarına eklenmesi gerekir. Her bir eklem için ivme alt ve üst sınırları belirtildiği gibi tanımlanır. $\underline{\ddot{q}}_i(s)$ i numaralı ekleme ait ivme alt sınırı ve $\overline{\ddot{q}}_i(s)$ i numaralı ekleme ait ivme üst sınırıdır.

$$\underline{\ddot{q}}_i(s) \leq \ddot{q}(s) \leq \overline{\ddot{q}}_i(s) \quad (3.42)$$

$$\underline{\ddot{q}}_i(s) \leq q_i'(s) + q_i''(s)s \leq \overline{\ddot{q}}_i(s) \quad (3.43)$$

$$\underline{\ddot{q}}_i(s) \leq q_i'(s)a(s) + q_i''(s)b(s) \leq \overline{\ddot{q}}_i(s) \quad (3.44)$$

Hız kısıtlarında belirtildiği gibi, ivme kısıtlarında da eklem uzayından uygulama uzayına geçmek mümkündür. Eklem uzayı ve uygulama uzayı kısıtları aşağıda belirtildiği gibi yazılabilir.

$$\underline{f}(s) \leq f(s)a(s) + h(s)b(s) \leq \overline{f}(s) \quad (3.45)$$

3.5.3.3 Tork deęişim kısıtı

Tork deęişim oranı ile ilgili tanımlamalar yapılabilir ve kısıtlar yazılabilir fakat burada kullanılacak denklemlerin konveks olmamasından dolayı optimizasyona dahil etmek mümkün deęildir.

3.5.4 Optimizasyon denklemlerinin genelleştirilmesi

Denklem (3.31)'den (3.33)' e kadar olan yeniden tanımlanan amaç fonksiyonuna (3.34)'den (3.44)' e kadar olan kısıt denklemleri eklenerek optimizasyon problemi genelleştirilmiştir.

$$\min \int_0^1 \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma_1}{\sqrt{b(s)}} \left(\sum_{i=0}^n \frac{\tau_i(s)^2}{\bar{\tau}_i^2} \right) + \gamma_1 \left(\sum_{i=0}^n \frac{|\tau_i'(s)|}{|\bar{\tau}_i|} \right) \right] \quad (3.46)$$

$$\tau(s) = m(s)a(s) + c(s)b(s) + g(s) \quad (3.47)$$

$$b(1) = \dot{s}_T^2, \quad (3.48)$$

$$b(1) = \dot{s}_T^2, \quad (3.49)$$

$$b(s) \geq 0, \quad (3.50)$$

$$b(s) \leq \bar{b}(s) \quad (3.51)$$

$$\underline{f}(s) \leq f(s)a(s) + h(s)b(s) \leq \bar{f}(s), \quad (3.52)$$

$$\underline{\tau}(s) \leq \tau(s) \leq \bar{\tau}(s) \quad (3.53)$$

$$s \in [0,1] \quad (3.54)$$

Burada $\bar{\tau}_i$ $i = 1$ 'den n 'e kadar tanımlanmıştır. Burada $s \in [0,1]$ 'e tanımlı bölge ile belirtilen yörünge planlama optimizasyon problemi, amaç fonksiyonunun konveks yapıda olması, eşitsizliklerin ve sınırların doğrusal olmasından dolayı konvektir.

3.6 Genelleştirilmiş Optimal Kontrol Probleminin Çözüm Yöntemleri

(3.45)'den (3.54)'e kadar belirtilen optimal kontrol probleminin çözümü için kullanılan yöntemler iki ana başlıkta incelenebilir: Dolaylı yöntemler ve d yöntemler. Birinci dereceden optimallik barındıran bir kontrol problemin çözümünde dolaylı yöntemler kullanılmaktadır. Dolaylı yaklaşım çoklu sınır şartlarına sahip bir problemin optimal adayı (ekstremum) değerleri bulunur. Hesaplanan her bir ekstremum değerinin yerel en az, yerel maksimum ya da aradeğer olup olmadığı incelenir. Amaç fonksiyonunu en küçük yapan ekstremum değeri optimal çözümlerden biridir. Doğrudan dönüşüm yöntemi üç kısma ayrılır: Doğrudan eşzamanlı yöntem, doğrudan ardışık yöntem, doğrudan çoklu atış yöntemi.

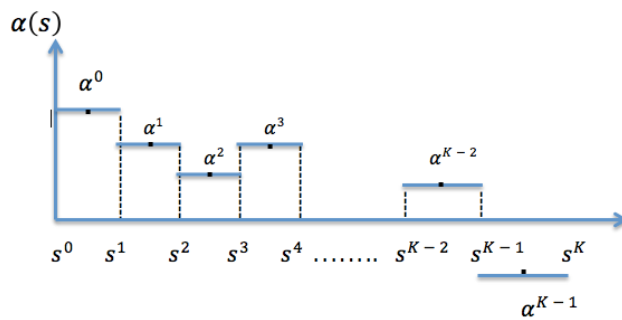
3.6.1 Doğrudan dönüşüm yöntemi

Bir kontrol probleminin ayrıklaştırılması ve doğrusal olmayan programlama tekniklerinin uygulanması ile sonlu boyutta optimizasyon problemi elde edilir.

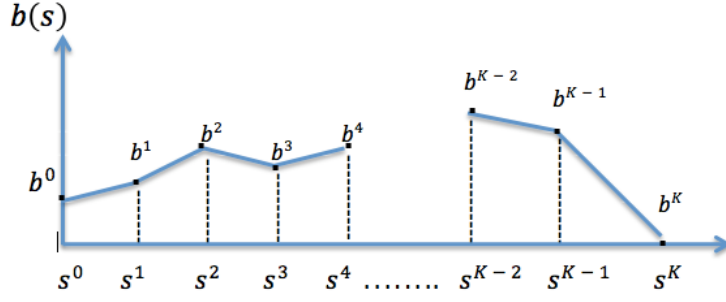
Bu yöntemle çözümde üç ana yaklaşım kullanılmaktadır: Doğrudan eşzamanlı yöntem, doğrudan ardışık yöntem (tekli atış), doğrudan çoklu atış. Genelleştirdiğimiz (3.45)'den (3.54)'e kadar belirtilen formüllere doğrudan dönüşüm yöntemi uygulanacaktır. $[0,1]$ arasında tanımlı yörünge koordinatı "s" ayrıklaştırılacaktır. K adette parçaya ayrılarak sistemde $K+1$ s noktası bulunacaktır.

$$s^0 = 0 \leq s^l \leq 1 = s^K, l = 0 \dots K \quad (3.55)$$

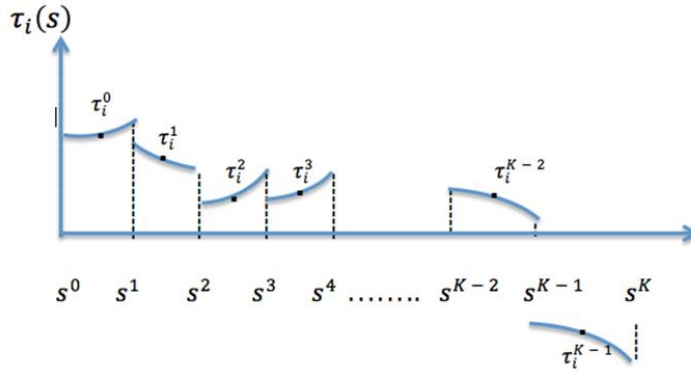
Ayrıca $b(s)$, $a(s)$ and $\tau_i(s)$ denklemleri sınırlı sayıda bölümlere ayrılarak b^k , a^k , τ_i^k olarak tanımlanmıştır. Değişken sayıları ve değişkenlerin noktaları doğrudan dönüşüm yönteminin farklılıklarına göre belirlenir.



Şekil 3.1 : $s \in [0, 1]$ için parçalı değişmez $\alpha(s)$.



Şekil 3.2 : $s \in [0, 1]$ için parçalı doğrusal $b(s)$.



Şekil 3.3 : $s \in [0, 1]$ için parçalı doğrusal olmayan $\tau_i(s)$.

Burada önerilen yöntemlerden bir tanesi üzerinde durulacaktır. (3.45)'den (3.54)'e kadar tanımlanan optimal kontrol probleminde $a(s)$ kontrol girişi olarak belirlenmiş parçalı değişmez yönteme dahil olduğu kabul edilmiştir (Şekil 3.1). (3.28)'de verilen denklemdeki kabule göre $b(s)$ parçalı doğrusal yönteme (Şekil 3.2) ve (3.25)'de verilen denkleme göre ise $\tau_i(s)$ parametresi parçalı doğrusal olmayan yönteme (Şekil 3.3) ait olduğu belirtilmiştir. Bu tanımlamalardan yola çıkarak $b(s)$ yeniden tanımlandığında

$$b(s) = b^k + \left(\frac{b^{k+1} - b^k}{s^{k+1} - s^k} \right) (s - s^k) \quad (3.56)$$

$s \in [s^k, s^{k+1}]$ aralığında tanımlıdır. s^k ile belirlenen noktaların ortasında $b(s^k) = b^k a^k$ ve τ_i^k bu değerler hesaplanır. Bu orta nokta $s^{k+1/2} = (s^k + s^{k+1})/2$ denklemi ile tanımlanır. Buradaki belirlenen noktada değişkenler $a^k = a(s^{k+1/2})$ and $\tau^k = \tau(s^{k+1/2})$ şeklinde belirlenir. Tanımlanan $k = 0 \dots K - 1$ aralığında $b(s)$ fonksiyonu parçalı doğrusal yöntemle çözülür. $b(s)$ 'in (3.45)'de tanımlanan ilk iki terimi integre edilirse;

$$\int_0^1 \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma_1}{\sqrt{b(s)}} \left(\sum_{i=0}^n \frac{\tau_i(s)^2}{\bar{\tau}_i^2} \right) \right] ds \quad (3.57)$$

$$= \sum_{k=0}^{K-1} \int_{s^k}^{s^{k+1}} \left[\frac{1}{\sqrt{b(s)}} + \frac{\gamma_1}{\sqrt{b(s)}} \left(\sum_{i=0}^n \frac{\tau_i(s)^2}{\bar{\tau}_i^2} \right) \right] ds \quad (3.58)$$

$$\approx \sum_{k=0}^{K-1} \left[1 + \gamma_1 \left(\sum_{i=0}^n \frac{\tau_i(s)^2}{\bar{\tau}_i^2} \right) \right] \int_{s^k}^{s^{k+1}} \frac{1}{\sqrt{b(s)}} ds \quad (3.59)$$

Tekil noktalardan kurtulmak için $b(s) = 0$ olduğu ve $1/\sqrt{b(s)}$ olan bölgeler ayrı değerlendirilecektir. (3.55) denlemi kullanılarak elde edilen (3.58) denkleminin sağ tarafı yeniden düzenlenirse;

$$\sum_{k=0}^{K-1} \left[1 + \gamma_1 \left(\sum_{i=0}^n \frac{\tau_i(s)^2}{\bar{\tau}_i^2} \right) \right] \sqrt{b^{k+1}} + \sqrt{b^k} \frac{2 \Delta s^k}{\Delta s^k} \quad (3.60)$$

$\Delta s^k = s^{k+1} - s^k$ 'e eşit olduğu görülmektedir. Buradan yola çıkarak (3.54)'de tanımlanan genelleştirilmiş amaç denkleminde her bir i eklemi için

$$\int_0^1 \gamma_2 \left[\frac{|\tau_i'(s)|}{|\bar{\tau}_i|} \right] ds \approx +\gamma_2 \left(\sum_{k=0}^{K-1} \frac{|\Delta \tau_i^k|}{|\bar{\tau}_i|} \right) \quad (3.61)$$

i numaralı ekleme ait tork farkını $\Delta \tau_i^k = \tau_i^k - \tau_i^{k-1}$ ($k = 1 \dots K - 1$ 'e her bir eklem için tanımlı) Daha kısa şekilde yazarsak $b^{k+1/2} = (b^k + b^{k+1})/2$ şeklinde bir denklem elde ederiz.

(3.45)'ten (3.54)'e a kadar olan genelleştirilmiş optimizasyon denklemleri a^k , b^k ve τ^k parametreleri ile yeniden tanımlandığında,

$$\min_{a^k, b^k, \tau^k} \sum_{k=0}^{K-1} \frac{2 \Delta s^k \left(1 + \gamma_1 \left(\sum_{i=0}^n \frac{\tau_i(s)^2}{\bar{\tau}_i^2} \right) \right)}{\sqrt{b^{k+1}} + \sqrt{b^k}} + \gamma_2 \sum_{k=1}^{K-1} \left(\sum_{i=1}^n \left| \frac{\Delta \tau_i^k}{|\bar{\tau}_i|} \right| \right) \quad (3.62)$$

$$\tau^k = m(s^{k+1/2})a^k + c(s^{k+1/2})b^{k+1/2} + g(s^{k+1/2}) \quad (3.63)$$

$$b^0 = \dot{s}_0^2 \quad (3.64)$$

$$b^K = \dot{s}_T^2, \quad (3.65)$$

$$(b^{k+1} - b^k) = 2a^k \Delta s^k, \quad (3.66)$$

$$b^k \geq 0, \quad b^K \geq 0, \quad (3.67)$$

$$b^k \leq \bar{b}(s^k) \text{ ve } b^K \leq \bar{b}(s^K), \quad (3.68)$$

$$\underline{f}(s^{k+1/2}) \leq f(s^{k+1/2})a^k + h(s^{k+1/2})b^{k+1/2}, \quad (3.69)$$

$$f(s^{k+1/2})a^k + h(s^{k+1/2})b^{k+1/2} \leq \bar{f}(s^{k+1/2}) \quad (3.70)$$

$$\underline{\tau}(s^{k+1/2}) \leq \tau^k \leq \bar{\tau}(s^{k+1/2}) \quad (3.71)$$

(3.61)'den (3.70)'e kadar olan denklemlerin konveks olmalarından dolayı herhangi yerel optimum değeri genel olarak optimum değerine eşittir. Problem herhangi bir doğrusal olmayan çözücü ile çözülebilir. Buradaki denklemi ikinci dereceden konik programlama (SOCP) halinde yazıldığı zaman verimli bir şekilde çözümü mümkün olacaktır.

3.6.2 İkinci dereceden konik programlama (SOCP)

İkinci dereceden konik programlamanın standart formu

$$\min_x f^T x \quad (3.72)$$

$$\text{Koşul } Fx = g, \quad (3.73)$$

$J = 1$ 'den m 'e tanımlı

$$\|M_j x + n_j\|_2 \leq p_j^T x + q_j \quad (3.74)$$

(3.61)'den (3.71)'e kadar olan denklemlerin bu forma dönüştürülmesi gerekmektedir. Öncelikle (3.61)'den (3.71)'e kadar olan denklemlerin eşlenik bir doğrusal amaç fonksiyonu şeklinde tanımlanması gerekir. Yeni tanımlanacak formülde kullanılacak d^k for $k = 0 \dots K - 1$ ve $e^k \in \mathcal{R}^n$ $k = 1 \dots K - 1$ (3.61) denkleminin yeniden tanımlanmasıyla

$$\sum_{k=0}^{K-1} 2 \Delta s^k d^k + \gamma_2 \sum_{k=1}^{K-1} 1^T e^k \quad (3.75)$$

ve (3.61)'den (3.71)'e kadar sisteme ait eşitsizlik sınırlarınında sisteme dahil edilmesi gerekir.

$$\frac{(1+\gamma_1 \sum_{i=1}^n (\tau_i^k)^2 / \bar{\tau})}{\sqrt{b^{k+1}} + \sqrt{b^k}} \leq d^k, \quad (k = 0 \dots K - 1), \quad (3.76)$$

ve

$$-e^k \leq \begin{pmatrix} \Delta \frac{\tau_1^k}{|\bar{\tau}_1|} \\ \Delta \frac{\tau_n^k}{|\bar{\tau}_n|} \end{pmatrix} \leq e^k, \quad (k = 1 \dots K - 1). \quad (3.77)$$

Denklemleri elde edilir. Ayrıca ikinci dereceden konik programlamaya ait kısıtlamalar c^k parametresi ($k = 0 \dots K$) eklenerek (3.76)'da yerine iki farklı eşlenik denklem şeklinde belirtilebilir.

$$\frac{(1+\gamma_1 \sum_{i=1}^n (\tau_i^k)^2 / \bar{\tau})}{c^{k+1} + c^k} \leq d^k, \quad (k = 0 \dots K - 1), \quad (3.78)$$

$$c^k \leq \sqrt{b^k}, \quad (k = 0 \dots K) \quad (3.79)$$

(3.78) ve (3.79)'da bulunan eşitsizlikler ikinci dereceden konik kısıtlamaların genel gösterimi şeklinde ifade edildiğinde (3.74) denklemi,

$$\frac{\left(1 + \frac{\gamma_1 \sum_{i=1}^n (\tau_i^k)^2}{\bar{\tau}}\right)}{c^{k+1} + c^k} \leq d^k \quad (3.80)$$

$$\left\| \begin{array}{c} 2 \\ 2\sqrt{\gamma_1} \tau_1^k / \bar{\tau}_1 \\ \vdots \\ 2\sqrt{\gamma_1} \tau_n^k / \bar{\tau}_n \end{array} \right\|_{c^{k+1} + 2c^k - d^k} \leq c^{k+1} + c^k + d^k \quad (3.81)$$

$$k = 0 \dots K - 1 \quad (3.82)$$

$$c^k \leq \sqrt{b^k} \quad (3.83)$$

$$\left\| \frac{2c^k}{b^k - 1} \right\|_2 \leq b^k + 1, \quad (3.84)$$

$$k = 0 \dots K. \quad (3.85)$$

İkinci dereceden konik programlamanın standart formunda yazıldığında;

$$\min_{a^k, b^k, \tau^k, c^k, d^k, e^k} \sum_{k=0}^{K-1} 2 \Delta s^k d^k + \gamma_2 \sum_{k=1}^{K-1} 1^T e^k \quad (3.86)$$

$$\tau^k = m(s^{k+1/2})a^k + c(s^{k+1/2})b^{k+1/2} + g(s^{k+1/2}) \quad (3.87)$$

$$b^0 = \dot{s}_0^2 \quad (3.88)$$

$$b^K = \dot{s}_T^2, \quad (3.89)$$

$$(b^{k+1} - b^k) = 2a^k \Delta s^k, \quad (3.90)$$

$$\underline{\tau}(s^{k+1/2}) \leq \tau^k \leq \bar{\tau}(s^{k+1/2}) \quad (3.91)$$

$$\left\| \begin{array}{c} 2 \\ 2\sqrt{\gamma_1}\tau_1^k/\tau_1 \\ \vdots \\ 2\sqrt{\gamma_1}\tau_n^k/\tau_n \\ c^{k+1} + c^k - d^k \end{array} \right\|_2 \leq c^{k+1} + c^k + d^k \quad (3.92)$$

$$\underline{f}(s^{k+1/2}) \leq f(s^{k+1/2})a^k + h(s^{k+1/2})b^{k+1/2} \quad (3.93)$$

$$f(s^{k+1/2})a^k + h(s^{k+1/2})b^{k+1/2} \leq \bar{f}(s^{k+1/2}) \quad (3.94)$$

$$k = 0 \dots K - 1 \quad (3.95)$$

$$\left\| \frac{2c^k}{b^k - 1} \right\|_2 \leq b^k + 1, \quad (3.96)$$

$$b^k \geq 0, \quad (3.97)$$

$$b^k \leq \bar{b}(s^k), \quad (3.98)$$

$$-e^k \leq \left(\Delta \frac{\tau_1^k}{|\bar{\tau}|} \right) \leq e^k \quad (k = 0 \dots K - 1) \quad (3.99)$$

(3.85) ile (3.99) denklemlerinde belirtilen a^k , τ^k , d^k değişkenleri $k = 0 \dots K - 1$ aralığında, a , b^k , c^k değişkenleri $k = 0 \dots K$ aralığında, e^k for $k = 1 \dots K - 1$ aralığında tanımlıdır.

c^k , d^k ve e^k , parametreleri optimizasyon çözüm kütüphanesinde tanımlanmalıdır. (3.85) ve (3.99) arasında tanımlanan denklemler kolaylıkla Matlab'ın optimizasyon çözüm kütüphanesi SEDUMI'ye eklenbilmektedir. Öncelikle b^k , değerleri için bir çözüm üretilir ve $s(t)$ yörünge koordinatı ve zaman arasındaki $t(s)$ bağıntı denklem (3.100)'den hesaplanır.

$$t(s) = \int_0^s \frac{1}{\sqrt{b(u)}} du \quad (3.100)$$

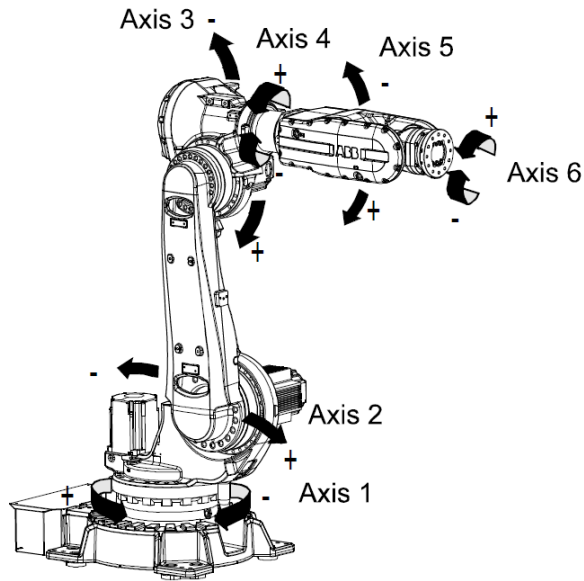
4. OPTİMAL YÖRÜNGE TAKİBİ

4.1 Robot Özellikleri

Kullanılan robot ABB tarafından üretilen IRB 6600 ailesinin bir parçası olan IRB 6620'dir. IRB 6620 Endüstriyel uygulamalar için geniş bir çalışma alanı ile esnek ve çevik bir robottur. Robot farklı montaj seçeneklerine sahiptir. Tabandan, tavandan, eğik olarak montajı mümkündür.

Çizelge 4.1 : Robotun teknik özellikleri ve boyutları.

Özellikler	Kolon A
Ulaşma Mesafesi	2.2 m
Yükleme Kapasitesi	150 kg
Eksen Sayısı	6
Koruma Sınıfı	IP 54
Montaj	Tavan, Taban, Eğik
Konum Tekrarlanabilirliği	3.1 mm



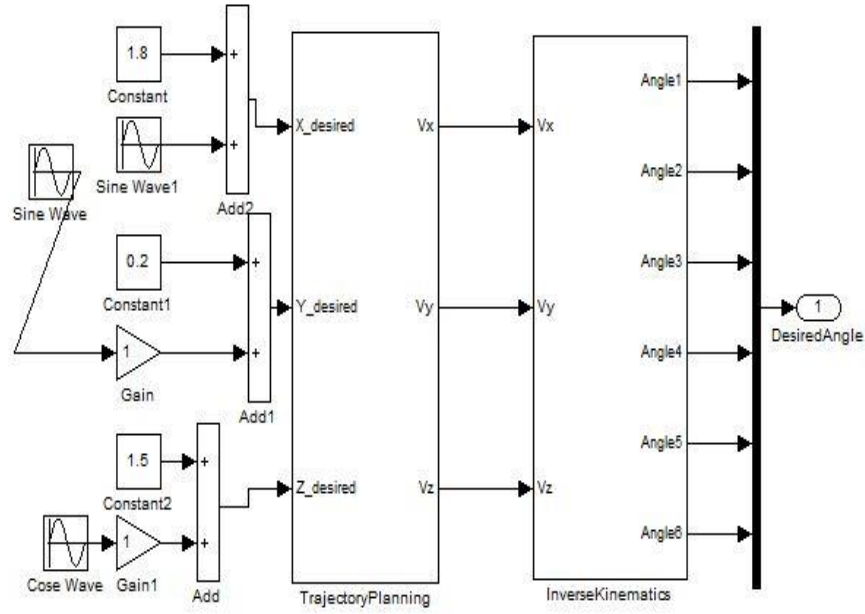
Şekil 4.1 : Robotun eksenel gösterimi.

Çizelge 4.2 : Eksenel performans değerleri

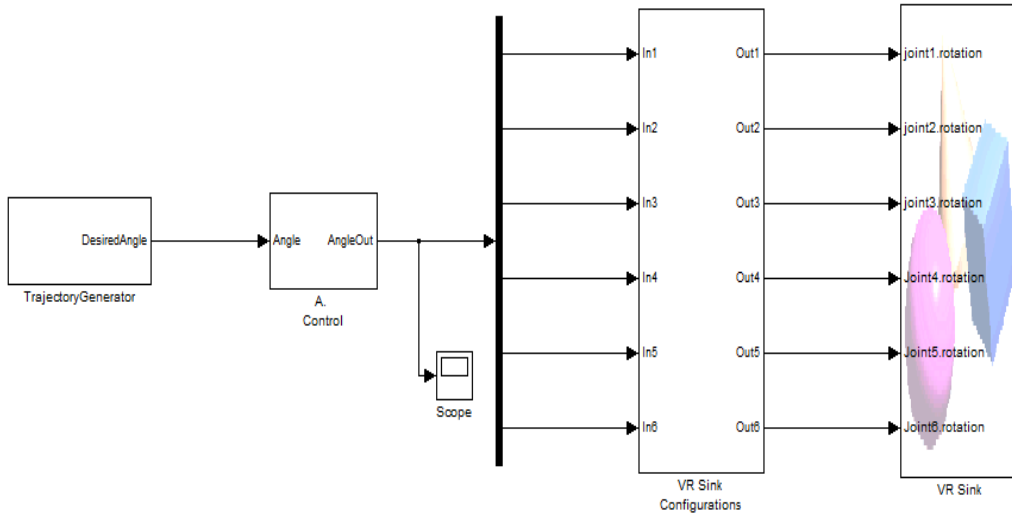
Eksen Numarası	Hareket Kapasitesi (°)	Maksimum Eksen Hızı (°/s)	Maksimum Tork Değerleri (Nm)
1	+170° to -170°	100°/s	4400 Nm
2	+140° to -65°	90°/s	15230 Nm
3	+70° to -180°	90°/s	2770 Nm
4	+300° to -300°	150°/s	736 Nm
5	+130° to -130°	120°/s	736 Nm
6	+300° to -300°	190°/s	383 Nm

4.2 Kinematik ve Dinamik Parametrelerin Elde Edilmesi

Oncelikle dinamik ve kinematik analizde robotun izlemesi gereken yörünge tanımlanmaktadır. Burada dinamik ve kinematik analiz işlemlerinin yapılması için Bölüm 2’de anlatılan uzaysal operatör cebiri kullanılmıştır. Dinamik ve kinematik analiz belirtilen formüllere göre *Matlab* programında hazırlanmıştır. Burada öncelikle robota ait kütle, atalet, geometrik boyutlar tanımlanmalıdır. Özellikleri belirtilen robota izlemesi gereken yörünge tanımlanır. Şekil 4,5’de gösterilen simülink dosyayı içerisinde “Trajectory Generator” bloğuna giriş yapılır. Burada robotun uç noktasının (uygulama uzayı) takip etmesi istenilen x,y,z yörüngeleri sisteme giriş olarak verilir. Örnek olarak gösterilecek harekette robotun bir daire çizdirilmesi istendiği için “X_Desired” parametresine sinüs girişi ve Y_Desired” parametresine cosinüs girişi verilerek robota daire çizdirilmektedir. Robotun her bir ekleminin açısı, ivmesi ve hızı ters kinematik çözümle elde edilmektedir. Ayrıca dinamik analiz ilede eklemlere ait tork değerlerine ulaşılmaktadır. Robotun izlediği yörünge “VR Sink” bloğundan ulaşılan şekil 4.3 te görülen *Virtual Reality* programında izlenebilir. Hesaplanan bu değerler “test.mat” dosyasına aktarılmaktadır. Çevrimdışı olarak hesaplanan zaman optimal ve enerji optimal yörünge planlama metoduna çevrimdışı olarak hesaplanan bu parametreler aktarılmaktadır.



Şekil 4.2 : Ters kinematik hesaplama Matlab Simulink bloğu.



Şekil 4.3 : Yörünge planlaması Matlab Simulink bloğu.

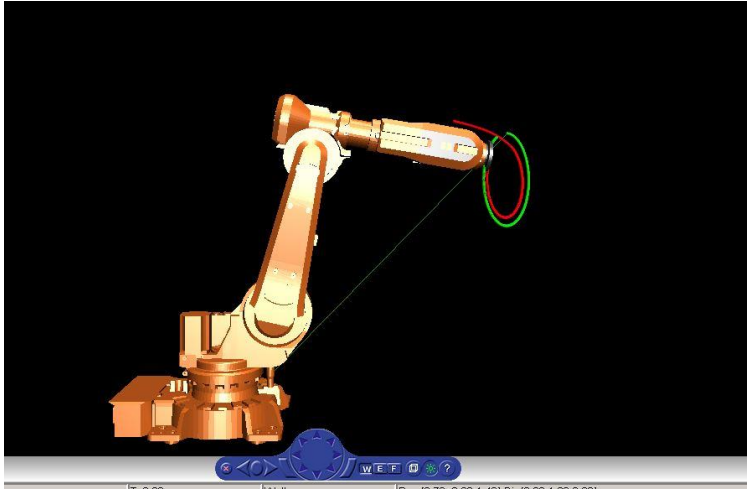
4.3 Optimizasyon Tanımlarının Yazılıma Eklenmesi

Her bir ekleme ait hız alt ve üst limitleri, ivme alt ve üst limitleri ve tork alt ve üst limitleri sistemde tanımlanır. “test.mat” dosyasından tanımlanan yörünge için her bir ekleme ait konum, hız, ivme değerleri yüklenir. Ayrıca sistemin dinamik değerleri de denklemde belirtilir. Yörünge $s(t)$ ’nin orta noktaları bulunur. Amaç fonksiyonu parametreleri sistemde tanımlanır. Optimizasyon fonksiyonundaki kısıtlar sisteme

eklenir. Optimizasyon fonksiyonu çözdürülür ve zaman optimal yörünge planlama grafikleri elde edilir.

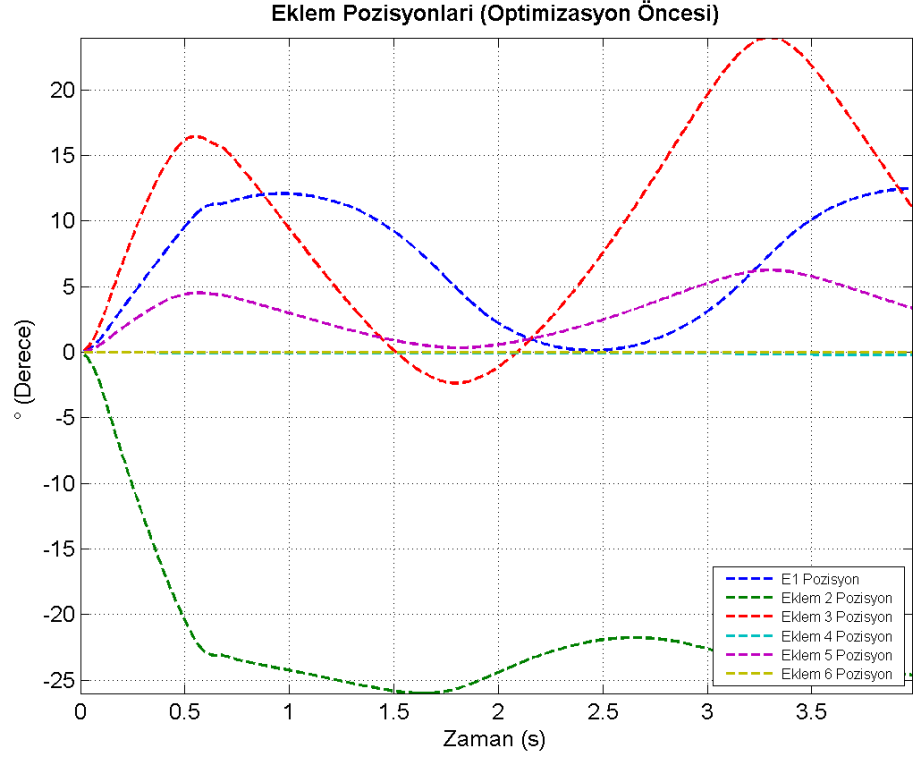
4.3.1 Daire çizdirme uygulaması

Bu uygulamada amaçlanan belirtilen robotun uç noktasına bir daire çizdirilecektir. Bu daire çizimi için dinamik ve kinematik hareket denklemlerinin oluşturulduğu *Matlab* programında geliştirilen yazılımda şekil 4.6’ da gösterildiği “Trajectory Generator” bloğu içerisindeki *Y_Desired* parametresine sinüs dalgası ve “*Z_Desired*” parametresine kosinüs dalgası girişi yapılarak robotun daire şeklindeki bir yörüngeyi takip etmesi amaçlanmaktadır. Robotun takip ettiği yörünge Şekil 4.7’de gösterilmektedir. “GeneralGui.m” çalıştırıldığında “test.mat” adında değişkenleri içeren bir dosya oluşur. Bu oluşan “test.mat” içerisinde eklem uzayında konum, hız ve ivme bilgilerini içermektedir. Ayrıca coriolis, kütle ve yerçekimi ivmesi matrislerini de içermektedir. Yerçekimi ivmesi matrisleri sıfır matrisidir, çünkü bu teoride yerçekimi matrisi coriolis matrisinin içerisinde yer almaktadır. Optimizasyon fonksiyon tanımına bağlı kalmak için yazılım parametrelerine *g* matrisi eklenmiştir.

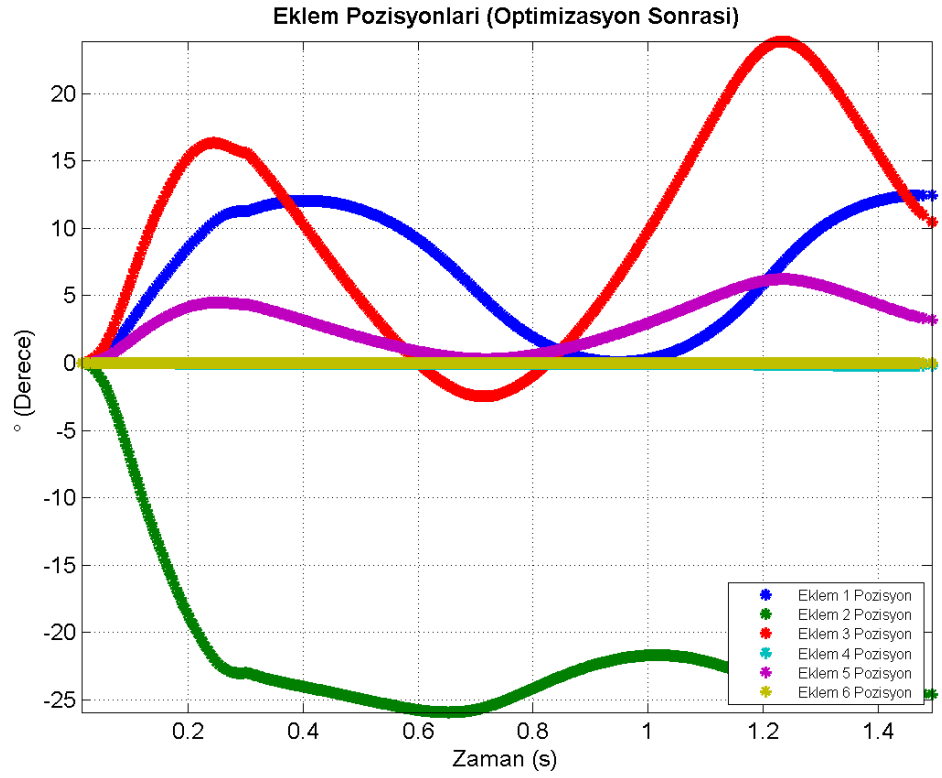


Şekil 4.4 : Virtual Reality’de robotun izlediği yörüngeyi gösterilmesi.

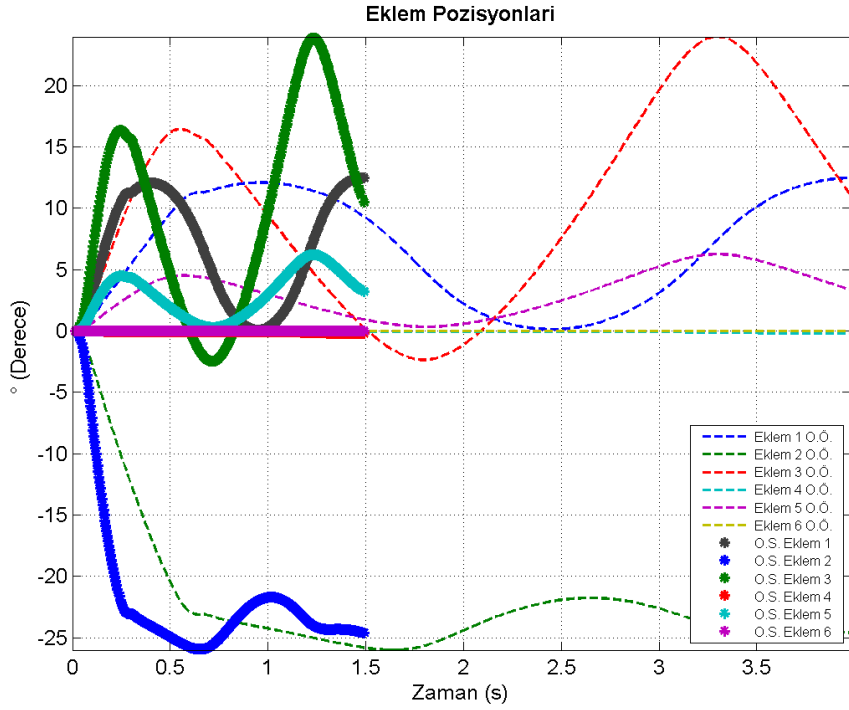
Oluşturulan yörünge için optimizasyon öncesi ve sonrası grafikler incelenmiştir. Her bir eklemin optimizasyon öncesi yörüngeyi zamana bağlı, optimizasyon sonrası yörüngeyi zamana bağlı ve her ikisinin de aynı grafikte gösterilmiş hali Şekil 4.5, Şekil 4.6, Şekil 4.7’de gösterilmektedir.



Şekil 4.5 : Optimizasyon öncesi eklem konumunun zamana bağlı grafiği.

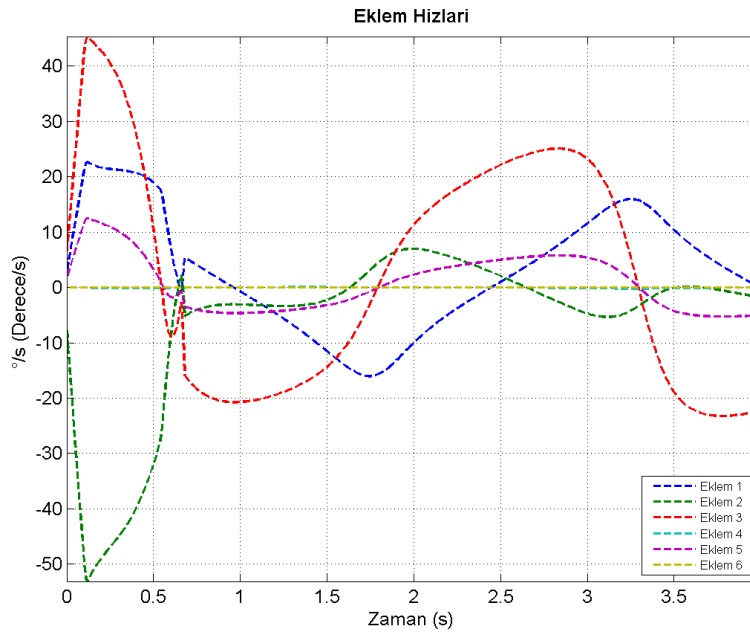


Şekil 4.6 : Optimizasyon sonrası eklem konumunun zamana bağlı grafiği.

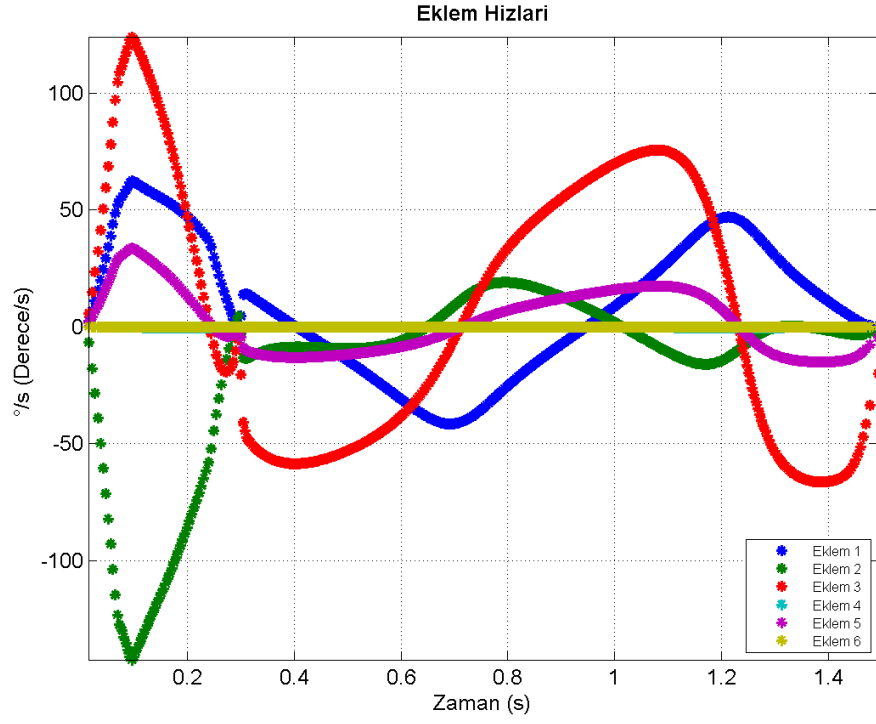


Şekil 4.7 : Optimizasyon öncesi ve sonrası konumun zamana bağlı grafiği.

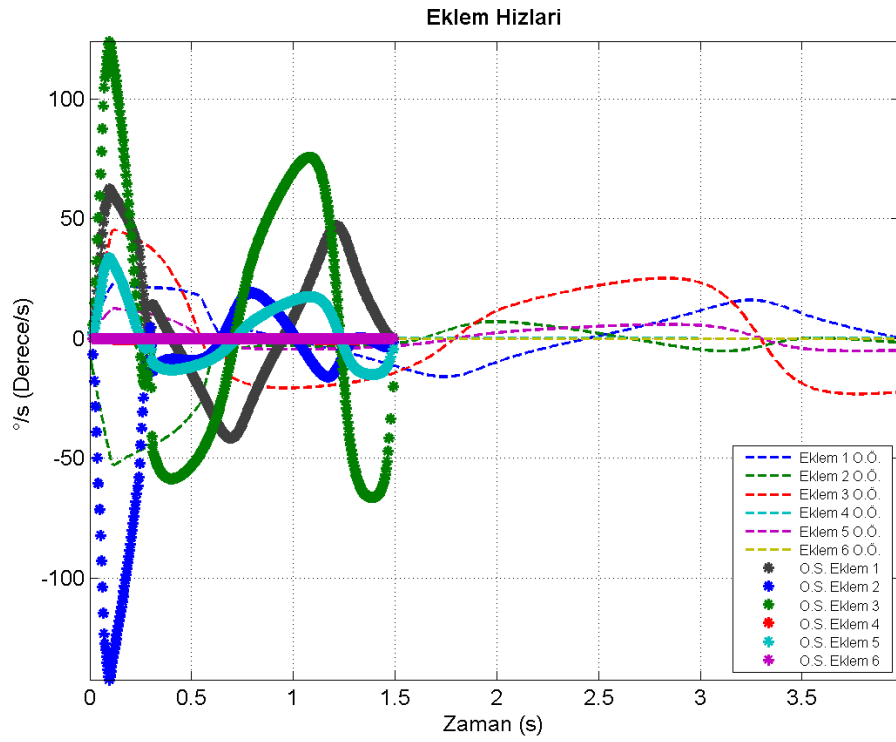
Eklem hızlarının zamana bağlı grafikleri optimizasyon öncesi, sonrası ve her ikisinin de gösterimi Şekil 4.8, Şekil 4.9, Şekil 4.10 ‘da gösterilmiştir.



Şekil 4.8 : Optimizasyon öncesi eklem hızlarının zamana göre grafiği.

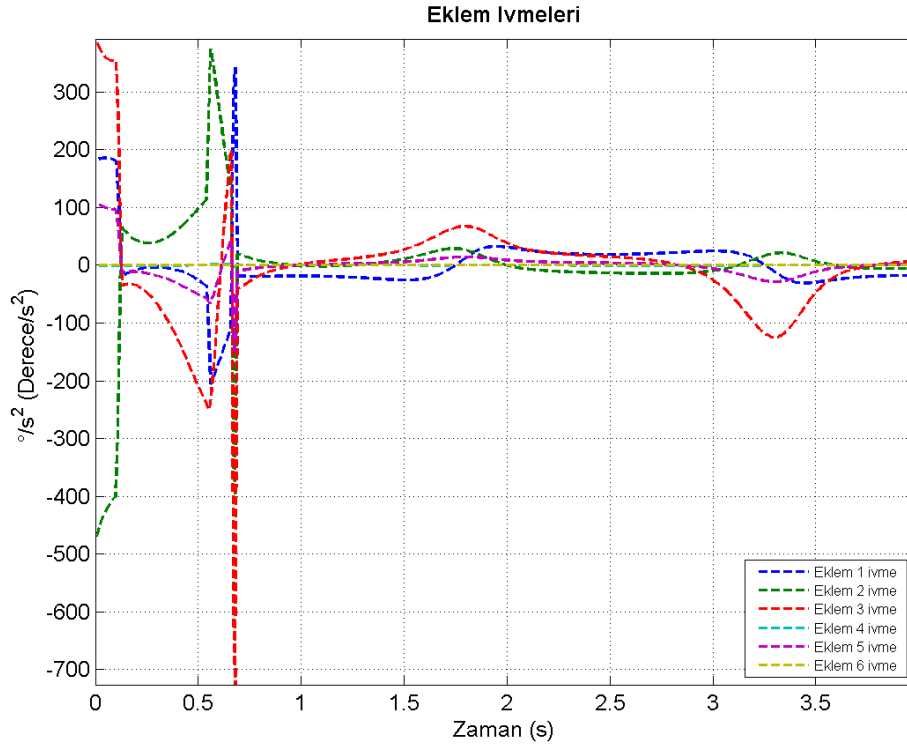


Şekil 4.9 : Optimizasyon sonrası eklem hızlarının zamana göre grafiği.

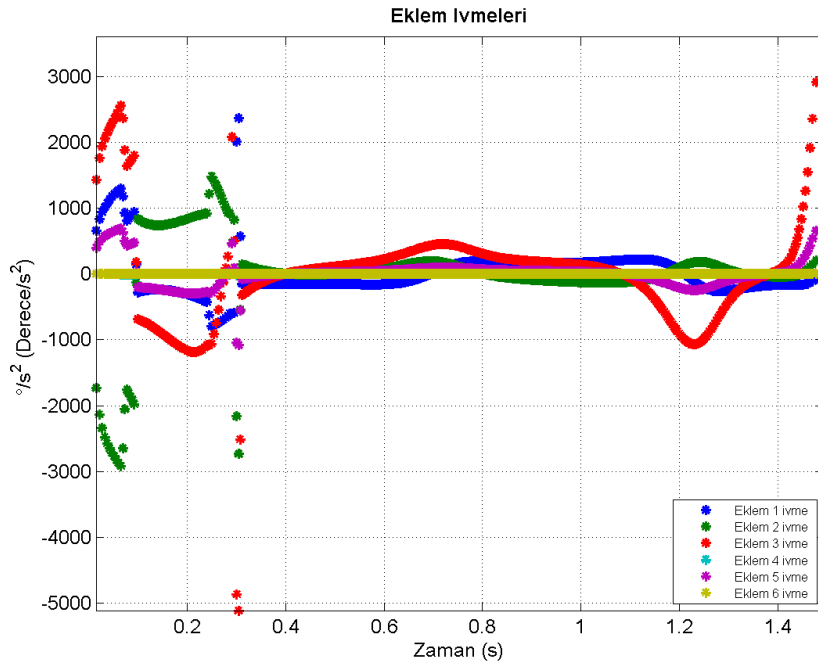


Şekil 4.10 : Optimizasyon öncesi ve sonrası hızların zamana göre grafiği.

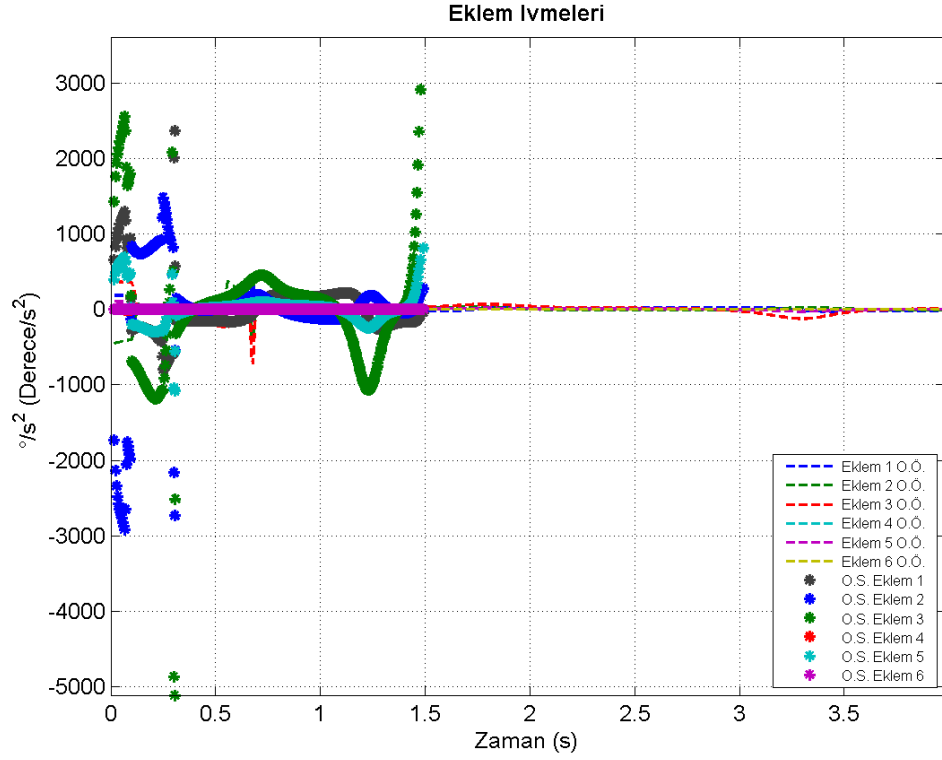
Şekil 4.11,Şekil 4.12, Şekil 4.13’de optimizasyon öncesi, sonrası ve her ikisinin ivmelerinin zamana göre değişimi gösterilmiştir.



Şekil 4.11 : Optimizasyon öncesi eklem ivmelerinin zamana göre grafiği.

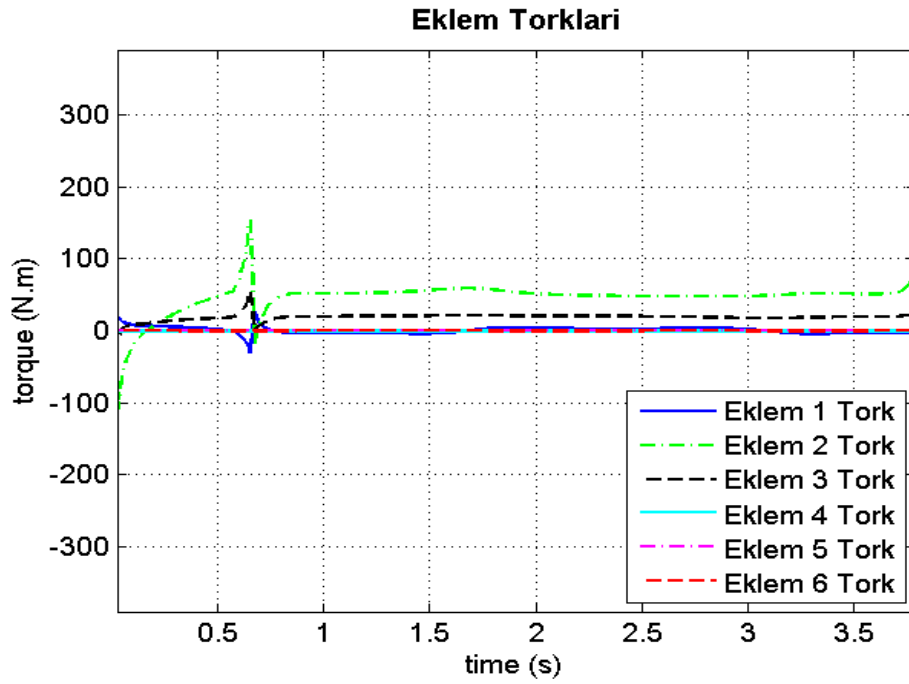


Şekil 4.12 : Optimizasyon öncesi eklem ivmelerinin zamana göre grafiği.

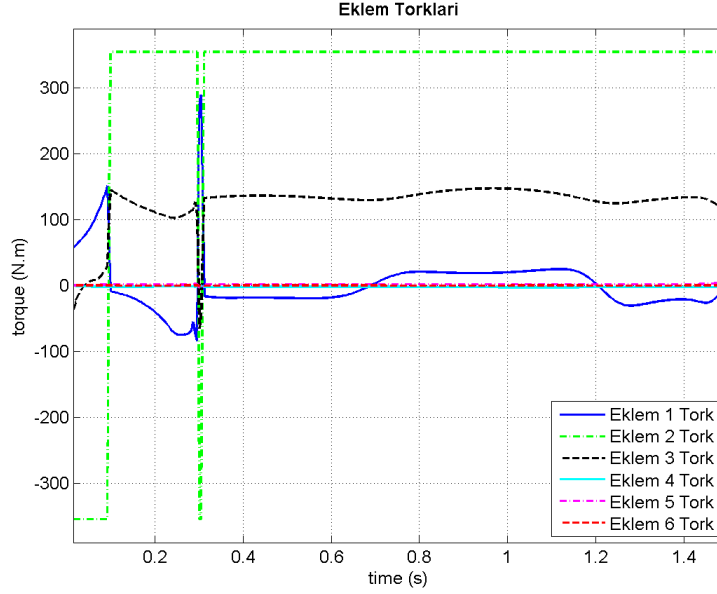


Şekil 4.13 : Optimizasyon öncesi ve sonrası hızların zamana göre grafiği.

Optimizasyon öncesi ve sonrası eklem torkları Şekil 4.14 ve Şekil 4.15’de gösterilmiştir.

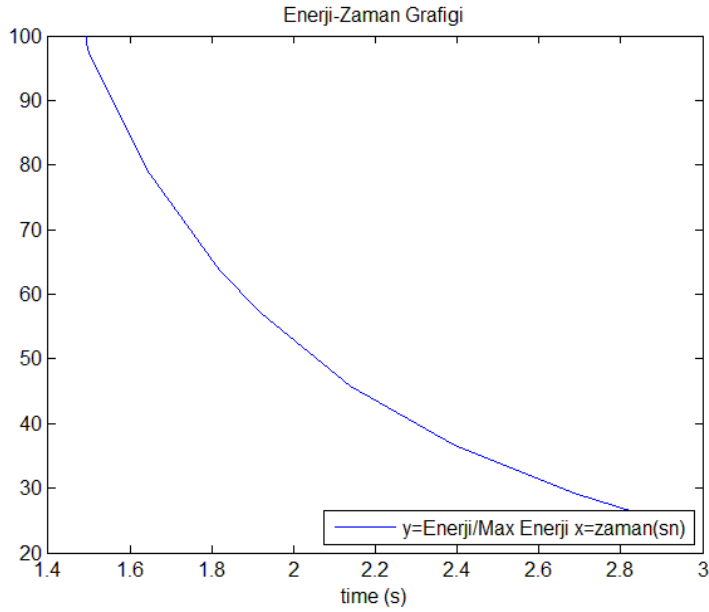


Şekil 4.14 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.



Şekil 4.15 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.

Eklem torkları zamana bağlı integre edilerek sözde enerji değeri elde edilmiştir. Yörünge zamanına göre toplam enerji optimizasyon sonrası enerji değerine oran olarak enerji zaman grafiği elde edilmiştir. Yörünge süresi optimal süre olan 1.5 saniyede tamamlandığında tüketilen enerji, 3 saniye sürece tamamlandığında harcanan enerjinin yaklaşık 5 katıdır. Bu enerji artışında ani ivme değişiklikleri etkilidir. Gerçekte harcanan enerjinin elde edilmesi için sistemden çekilen akımların ölçülmesi gerekir.

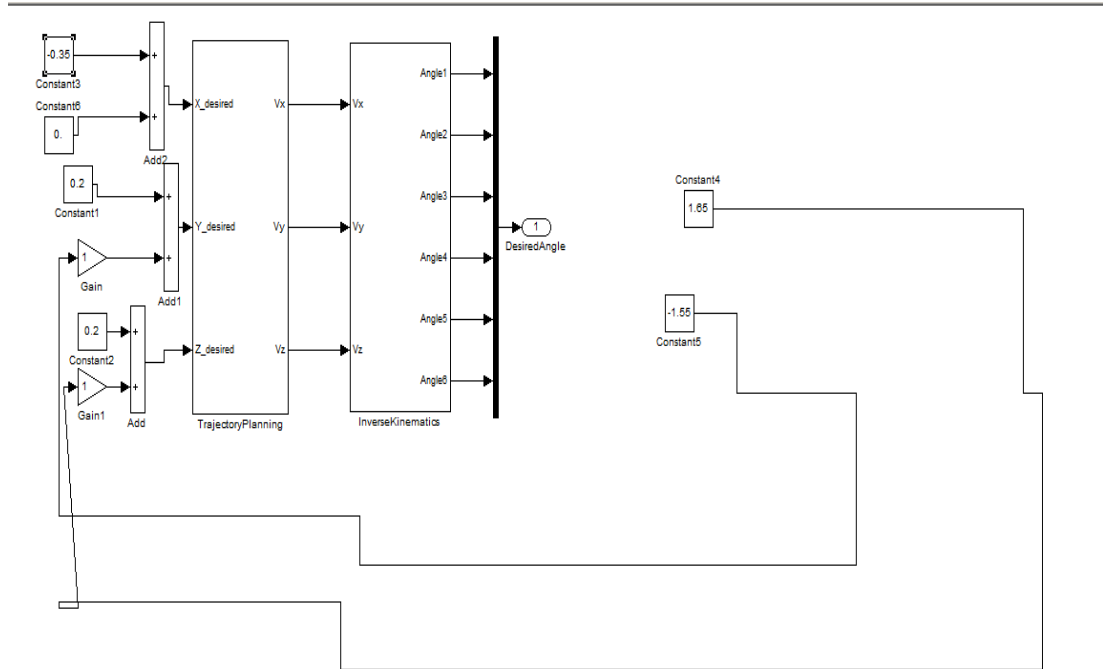


Şekil 4.16 : Zaman ve enerjinin birbirlerine göre değişimi.

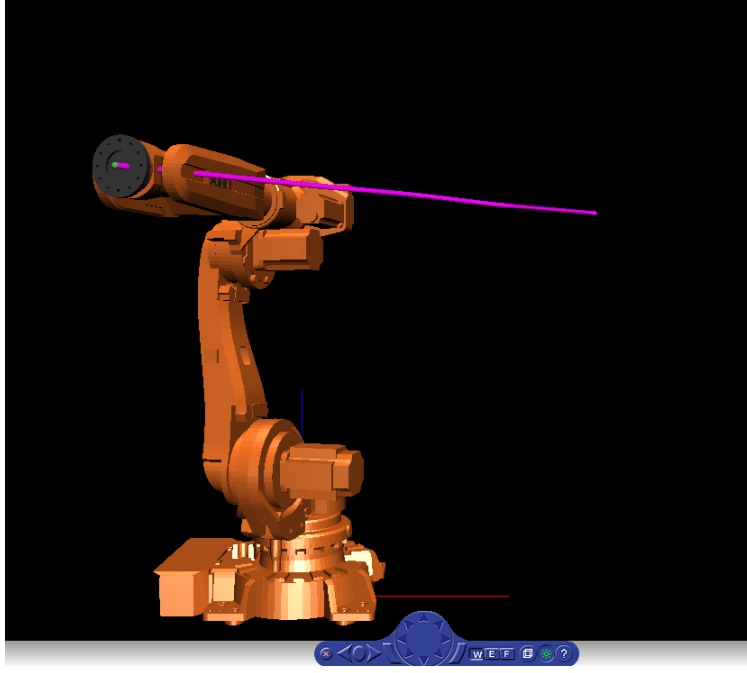
4.3.2 Doğrusal yörüngede hareket uygulaması

Bu uygulamada amaçlanan belirtilen robotun uç başlangıç noktasından sistemde belirtilen noktaya kadar doğrusal bir yörünge ile hareket ettirilecektir. Bu doğrusal yörünge için dinamik ve kinematik hareket denklemlerinin oluşturulduğu *Matlab* programında geliştirilen yazılımda Şekil 4.17’de gösterildiği “Trajectory Generator” bloğu içerisindeki Y_Desired parametresine ve “Z_Desired” parametresine şekilde gösterilen konum girişi yapılarak robotun doğrusal bir yörünge ile belirtilen noktalara ulaşması amaçlanmaktadır. Robotun takip ettiği yörünge Şekil 4.20’de gösterilmektedir. “GeneralGui.m” çalıştırıldığında “test.mat” adında değişkenleri içeren bir dosya oluşur.

Bu oluşan “test.mat” içerisinde eklem uzayında konum, hız ve ivme bilgilerini içermektedir. Ayrıca coriolis, kütle ve yerçekimi ivmesi matrislerini de içermektedir. Yerçekimi ivmesi matrisleri sıfır matrisidir, çünkü bu teoride yerçekimi matrisi coriolis matrisinin içerisinde yer almaktadır. Optimizasyon fonksiyon tanımına bağlı kalmak için yazılım parametrelerine g matrisi eklenmiştir.

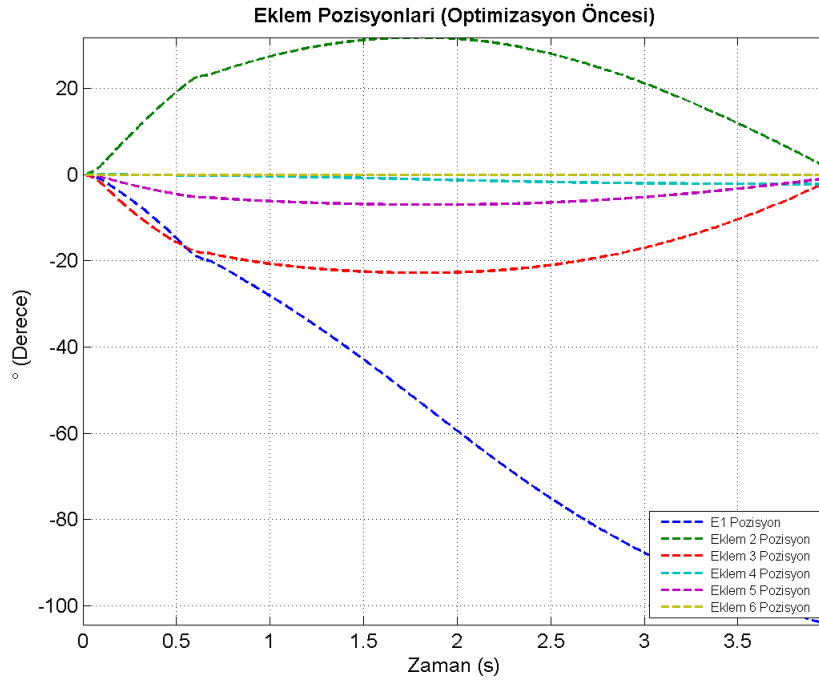


Şekil 4.17 : Yörünge parametreleri girişi Matlab Simulink bloğu.

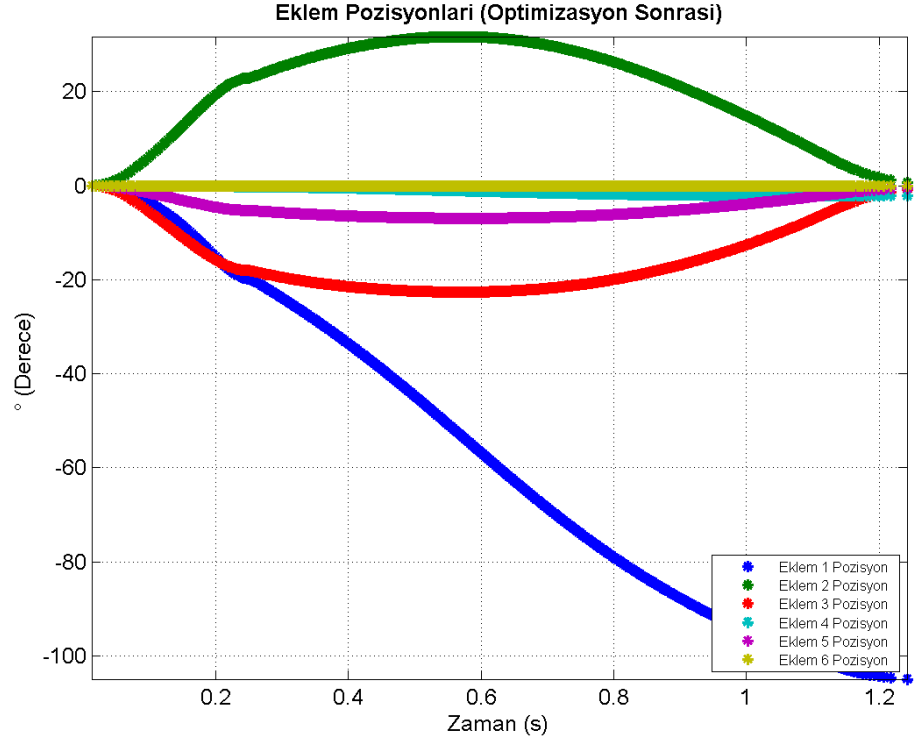


Şekil 4.18 : Virtual Reality’de robotun izlediği yörüngeyi gösterilmesi.

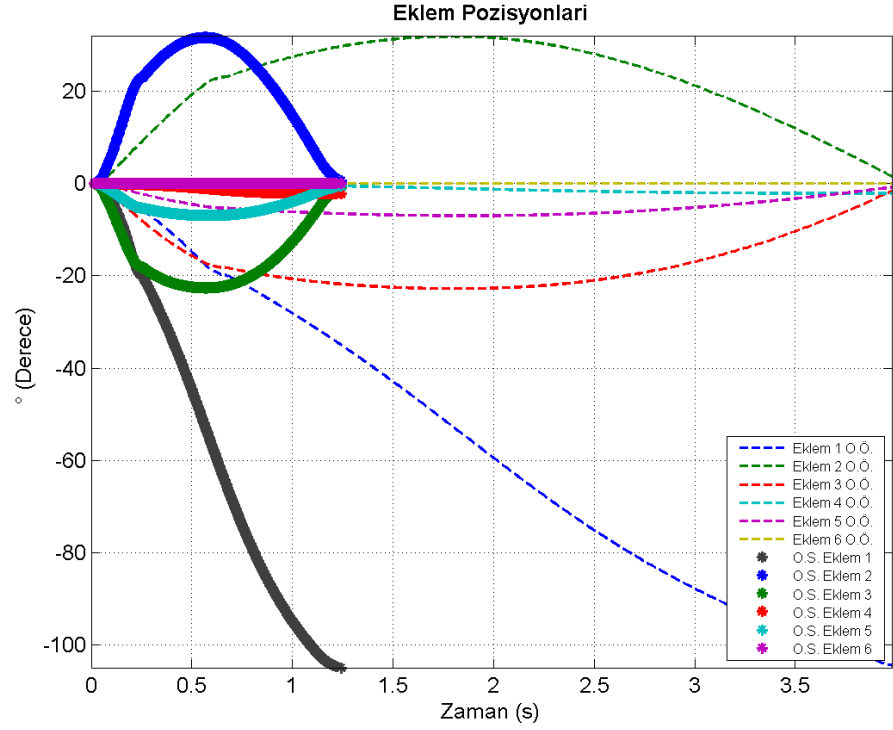
Oluşturulan yörünge için optimizasyon öncesi ve sonrası grafikler incelenmiştir. Her bir eklemin optimizasyon öncesi yörüngeyi zamana bağlı, optimizasyon sonrası yörüngeyi zamana bağlı ve her ikisinin aynı grafikte gösterilmiş hali Şekil 4.19, Şekil 4.20 ve Şekil 4.21’de gösterilmektedir.



Şekil 4.19 : Optimizasyon öncesi eklem konumlarının zamana bağlı grafiği.

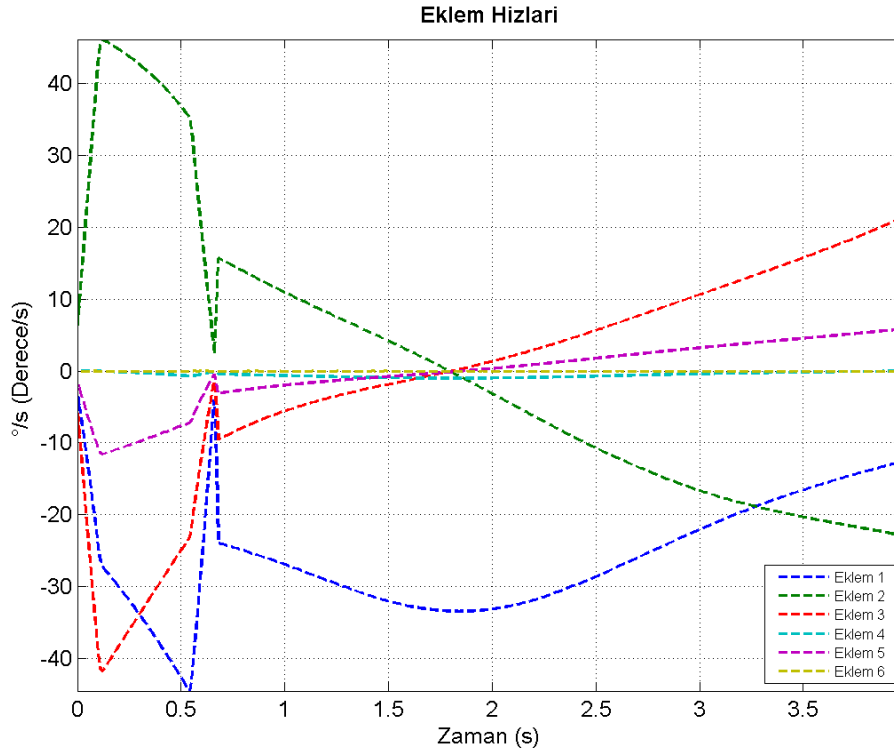


Şekil 4.20 : Optimizasyon sonrası eklem konumlarının zamana bağlı grafiği.

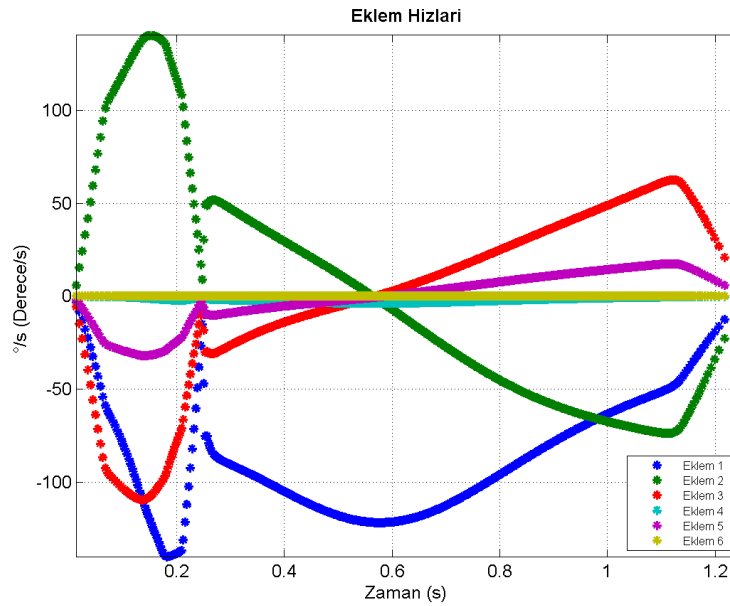


Şekil 4.21 : Optimizasyon öncesi ve sonrası konumların zamana bağlı grafiği.

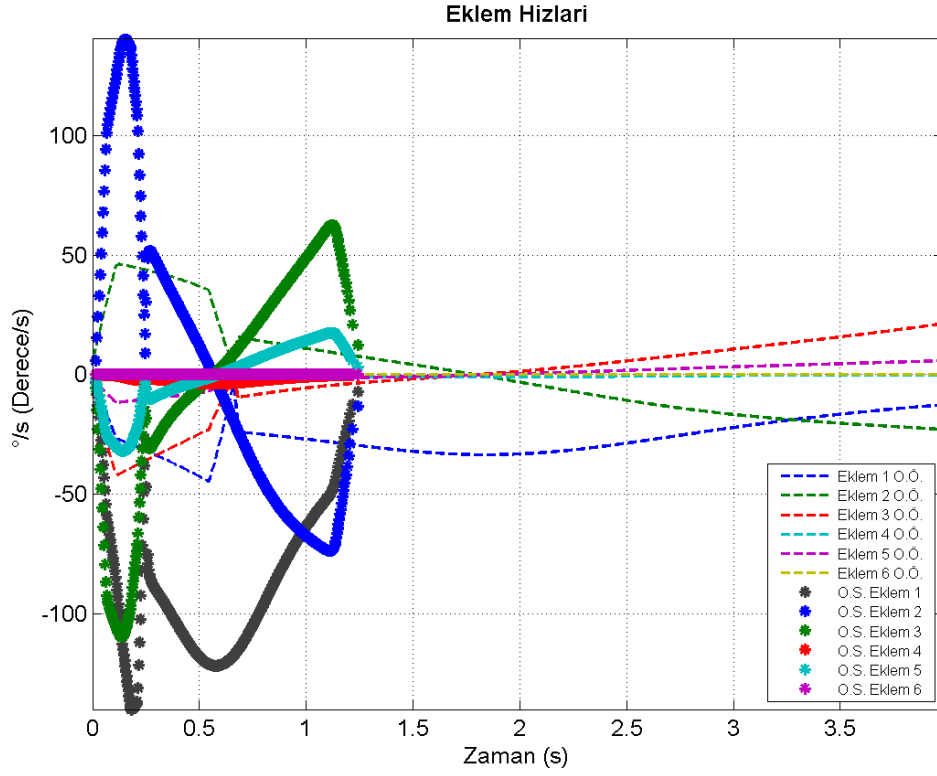
Eklem hızlarının zamana bağlı grafikleri optimizasyon öncesi, sonrası ve her ikisinin de gösterimi Şekil 4.22, Şekil 4.23 ve Şekil 4.24 ‘de gösterilmiştir.



Şekil 4.22 : Optimizasyon öncesi eklem hızlarının zamana bağlı grafiği.

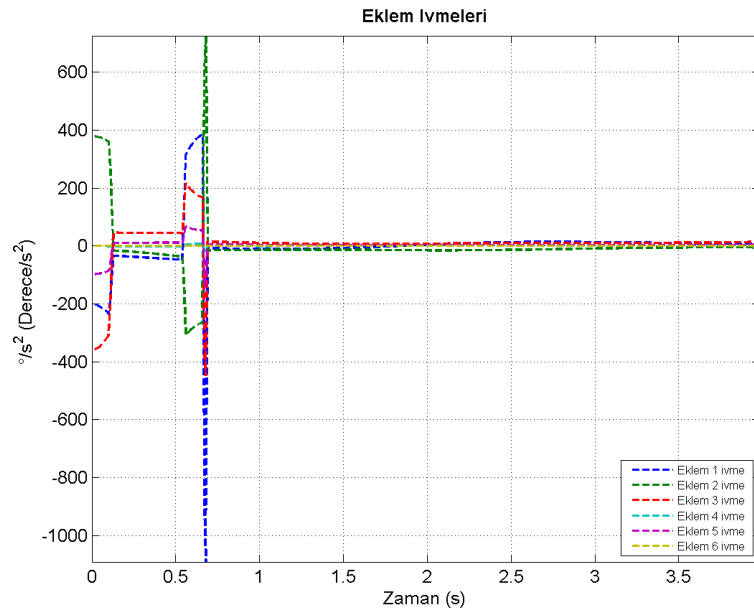


Şekil 4.23 : Optimizasyon sonrası eklem hızlarının zamana bağlı grafiği.

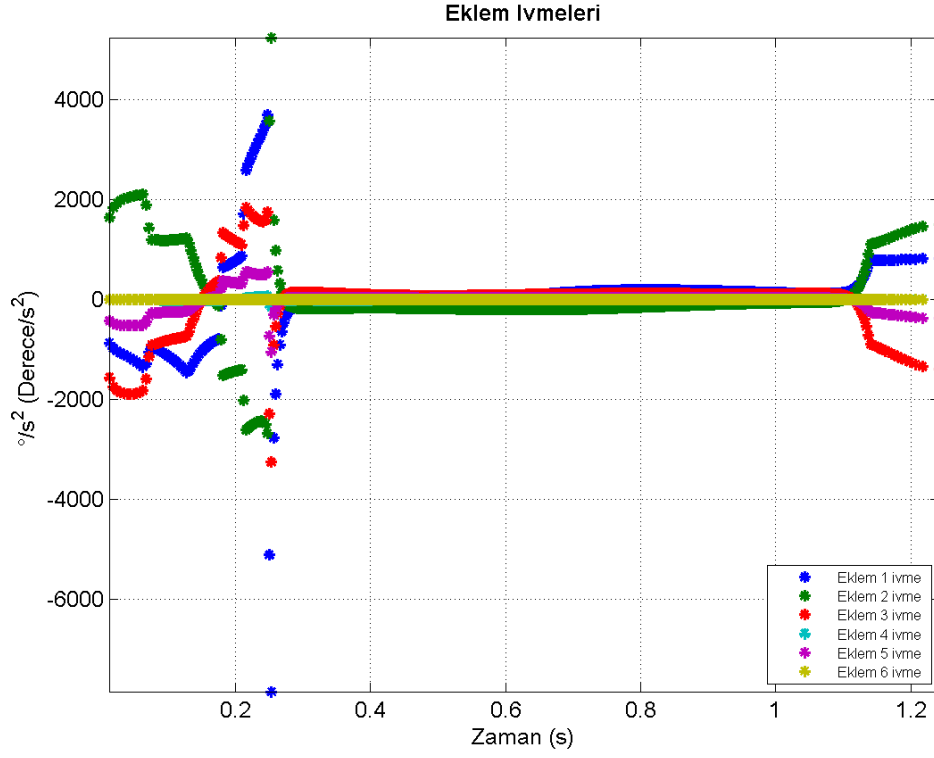


Şekil 4.24 : Optimizasyon öncesi ve sonrası hızların zamana bağlı grafiği.

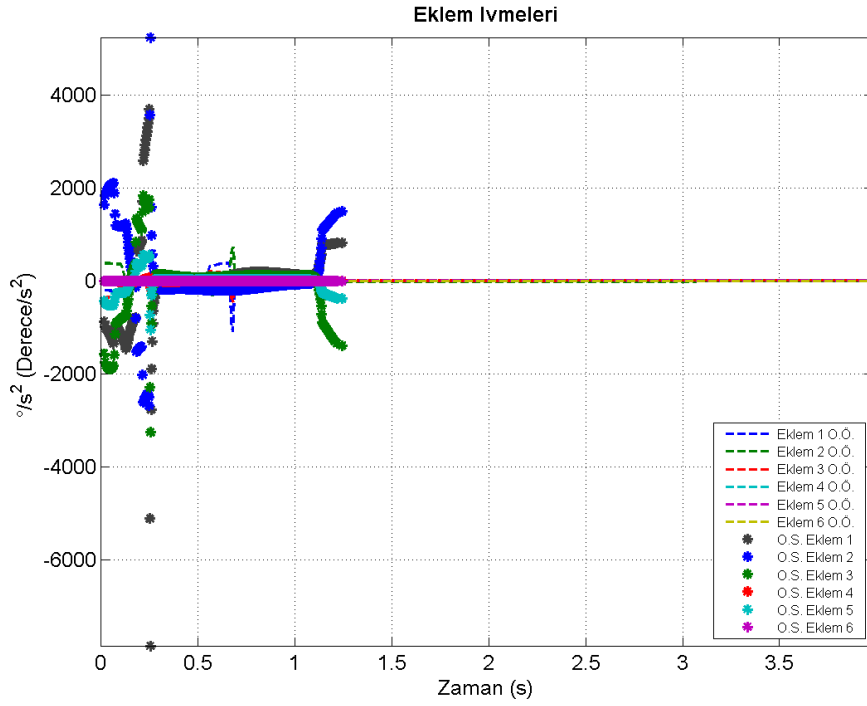
Şekil 4.25, Şekil 4.26 ve Şekil 4.27 'de optimizasyon öncesi, sonrası ve her ikisinin ivmelerinin zamana göre değişimi gösterilmiştir.



Şekil 4.25 : Optimizasyon öncesi eklem ivmelerinin zamana bağlı grafiği.

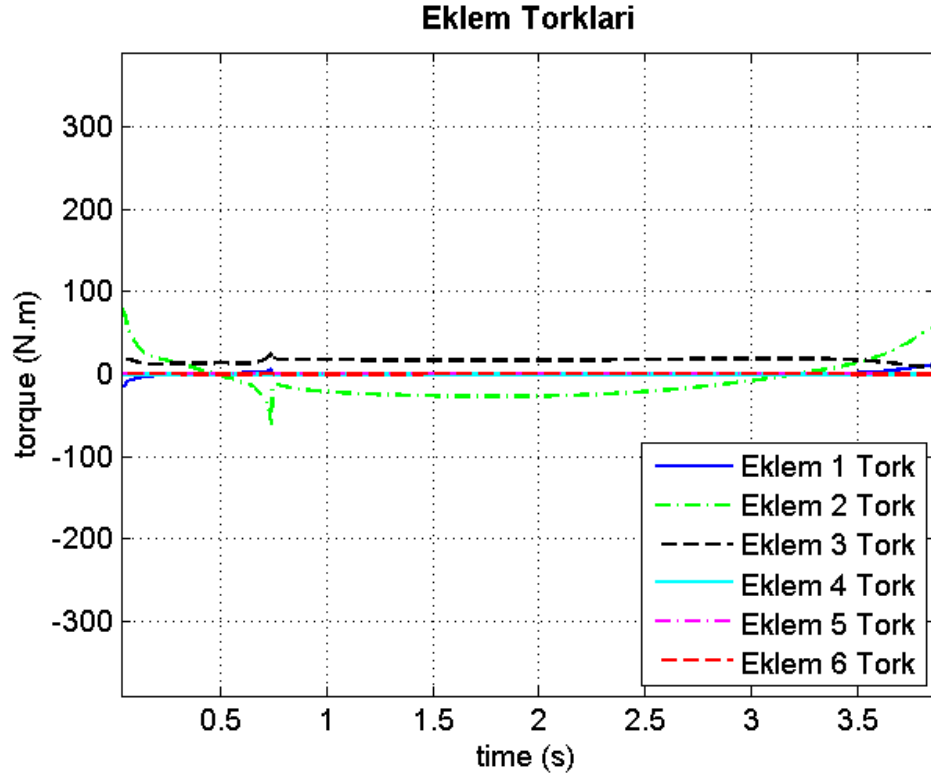


Şekil 4.26 : Optimizasyon sonrası eklem ivmelerinin zamana bağlı grafiği.

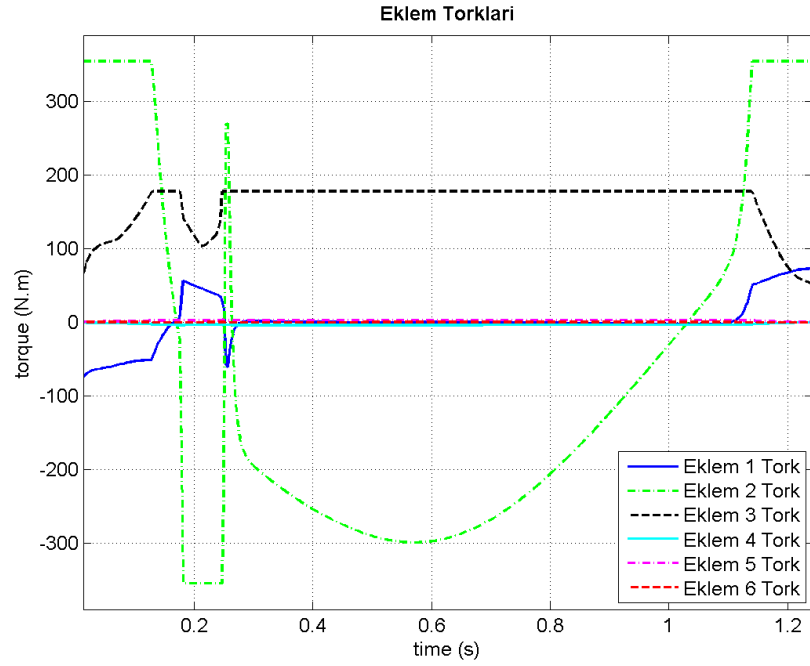


Şekil 4.27 : Optimizasyon öncesi ve sonrası ivmelerin zamana bağlı grafiği.

Optimizasyon öncesi ve sonrası eklem torkları Şekil 4.28 ve Şekil 4.29'da gösterilmiştir.



Şekil 4.28 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.



Şekil 4.29 : Optimizasyon sonrası eklem torklarının zamana bağlı grafiği.

5. SONUÇ VE ÖNERİLER

Bu tezde robotlarda dinamikaik ve dinamik yörünge denklemlerinin oluşturulması ve bu oluşturan denklemlerde robotun yörüngesinin tanımlaması yapılmıştır. Tanımlanan yörüngelere hareketin en az zamana göre optimizasyonu ve zaman-enerji optimizasyonu üzerine çalışmalar yapılmıştır. Kinematik ve dinamik denklemlerde UOC uzaysal operator cebiri metodu kullanılmıştır. Bu metot diğer kinematik ve dinamik hareket planlaması metotlarına göre değişen serbestlik derecelerine göre denklemlerin güncellenmesi çok kolay olan bir yöntemdir. Yörünge optimizasyonunda ise konveks optimizasyon metotlarından ikinci dereceden konik programlama metodu kullanılmıştır. Minimum zamana ulaşmak optimizasyon amacı olarak belirlenmiştir. Geliştirilen uygulamalarda robota dairesel yörünge takibi ve doğrusal yörünge takibi yaptırılmıştır. Dairesel yörüngenin takibinde robotun yörüngeyi tamamlama zamanı 3.6 saniyeden 1.5 saniyeye indirilmiştir. Doğrusal yörünge takibinde ise robotun yörüngeyi tamamlama zamanı 3.7 saniyeden 1.25 saniyeye indirilmiştir. Ayrıca robotun harcadığı enerjinin zamana göre grafiğide incelenmiştir. Dairesel yörünge takibinde yörünge optimal süre olan 1.5 saniyede tamamlandığında tüketilen enerji, 3.6 saniyede tamamlandığında harcanan enerjinin yaklaşık 5 katıdır. Burada enerji artışını etkileyen en önemli parametre ivmelerde meydana gelen ani değişikliklerdir. Bu çalışmanın devamında benzetim ortamında denenmiş olan yazılımın çevrimdışı olarak bir robot üzerinde denenmesi düşünülmektedir. Bir sonraki çalışmada ise çevrimiçi olarak robota optimal yörünge planlamasının yapılması ve bunun için optimizasyon algoritmasının hızlandırılması için çalışması amaçlanmaktadır.

KAYNAKLAR

- [1] **Paul, R.P.**, (1981). Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators: Richard Paul.
- [2] **Lloyd, J. ve Hayward, V.**, (1993). "Trajectory generation for sensor-driven and time-varying tasks", The International journal of robotics research, 12: 380-393.
- [3] **Weinreb, A. ve Bryson, A.**, (1985). "Optimal control of systems with hard control bounds", Automatic Control, IEEE Transactions on, 30: 1135-1138.
- [4] **Sahar, G. ve Hollerbach, J.M.**, (1986). "Planning of minimum-time trajectories for robot arms", The International journal of robotics research, 5: 90-100.
- [5] **Geering, H. Guzzella, L. Hepner, S. ve Onder, C.**, (1985). "Time-optimal motions of robots in assembly tasks": IEEE.
- [6] **Meier, E.-B.**, (1987). An efficient algorithm for bang-bang control systems applied to a two-link manipulator: University Microfilms.
- [7] **Yang, H.S. ve Slotine, J.-J.E.**, (1994). "Fast Algorithms for Near-Minimum-Time Control of Robot Manipulators Communication", The International journal of robotics research, 13: 521-532.
- [8] **Wen, J. ve Desrochers, A.**, (1986). "Sub-time-optimal control strategies for robotic manipulators": IEEE.
- [9] **Shiller, Z. ve Dubowsky, S.**, (1989). "Robot path planning with obstacles, actuator, gripper, and payload constraints", The International journal of robotics research, 8: 3-18.
- [10] **Dubowsky, S. Norris, M. ve Shiller, Z.**, (1986). "Time optimal trajectory planning for robotic manipulators with obstacle avoidance: A CAD approach": IEEE.
- [11] **Shiller, Z. ve Dubowsky, S.**, (1988). "Global time optimal motions of robotic manipulators in the presence of obstacles": IEEE.
- [12] **Bobrow, J.E.**, (1988). "Optimal robot plant planning using the minimum-time criterion", Robotics and Automation, IEEE Journal of, 4: 443-450.
- [13] **Slotine, J.-J. ve Yang, H.S.**, (1989). "Improving the efficiency of time-optimal path-following algorithms", Robotics and Automation, IEEE Transactions on, 5: 118-124.

- [14] **Zlajpah, L.,** (1996). "On time optimal path control of manipulators with bounded joint velocities and torques": IEEE.
- [15] **Shin, K.G. ve McKay, N.D.,** (1985). "Minimum-time control of robotic manipulators with geometric path constraints", *Automatic Control, IEEE Transactions on*, 30: 531-541.
- [16] **Shin, K.G. ve McKay, N.D.,** (1986). "A dynamic programming approach to trajectory planning of robotic manipulators", *Automatic Control, IEEE Transactions on*, 31: 491-500.
- [17] **Pfeiffer, F. ve Johanni, R.,** (1987). "A concept for manipulator trajectory planning", *Robotics and Automation, IEEE Journal of*, 3: 115-123.
- [18] **De Luca, A. Lanari, L. ve Oriolo, G.,** (1991). "A sensitivity approach to optimal spline robot trajectories", *Automatica*, 27: 535-539.
- [19] **Betts, J.T. ve Huffman, W.P.,** (1993). "Path-constrained trajectory optimization using sparse sequential quadratic programming", *Journal of Guidance, Control, and Dynamics*, 16: 59-68.
- [20] **Bobrow, J.E. Dubowsky, S. ve Gibson, J.,** (1985). "Time-optimal control of robotic manipulators along specified paths", *The International journal of robotics research*, 4: 3-17.
- [21] **Phu, H. Bock, H. ve Schölder, J.,** (1995). "Extremal solutions of some constrained control problems*", *Optimization*, 35: 345-355.
- [22] **Shiller, Z.,** (1996). "Time-energy optimal control of articulated systems with geometric path constraints", *Journal of Dynamic Systems, Measurement, and Control*, 118: 139-143.
- [23] **Shin, K.G. ve McKay, N.D.,** (1986). "Minimum-time trajectory planning for industrial robots with general torque constraints": IEEE.
- [24] **Constantinescu, D.,** (1998). *Smooth time optimal trajectory planning for industrial manipulators*, ed^{ed}s.: University of British Columbia.
- [25] **Costantinescu, D. ve Croft, E.,** (2000). "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths", *Journal of robotic systems*, 17: 233-249.
- [26] **Macfarlane, S. ve Croft, E.A.,** (2003). "Jerk-bounded manipulator trajectory planning: design for real-time applications", *Robotics and Automation, IEEE Transactions on*, 19: 42-52.
- [27] **Whitney, D.E.,** (1969). "Resolved motion rate control of manipulators and human prostheses", *IEEE Transactions on man-machine systems*.
- [28] **Taylor, R.H.,** (1979). "Planning and execution of straight line manipulator trajectories", *IBM Journal of Research and Development*, 23: 424-436.
- [29] **Volpe, R.,** (1993). "Task space velocity blending for real-time trajectory generation": IEEE.

- [30] **Paul, R.,** (1979). "Manipulator Cartesian path control", Systems, Man and Cybernetics, IEEE Transactions on, 9: 702-711.
- [31] **Kroger, T. Tomiczek, A. ve Wahl, F.M.,** (2006). "Towards on-line trajectory computation": IEEE.
- [32] **Angeles, J. Alivizatos, A. ve Zsombor-Murray, P.J.,** (1988). "The synthesis of smooth trajectories for pick-and-place operations", Systems, Man and Cybernetics, IEEE Transactions on, 18: 173-178.
- [33] **Gosselin, C. ve Hadj-Messaoud, A.,** (1993). "Automatic planning of smooth trajectories for pick-and-place operations", Journal of Mechanical Design, 115: 450-456.
- [34] **Cao, B. ve Dodds, G.I.,** (1994). "Time-optimal and smooth joint path generation for robot manipulators": IET.
- [35] **Bazaz, S.A. ve Tondur, B.,** (1997). "Online computing of a robotic manipulator joint trajectory with velocity and acceleration constraints": IEEE.
- [36] **Brady, M.,** (1982). Robot motion: Planning and control: MIT press.
- [37] **Craig, J.J.,** (2005). Introduction to robotics: mechanics and control: Pearson/Prentice Hall Upper Saddle River, NJ, USA:.
- [38] **Kim, B.K. ve Shin, K.G.,** (1985). "Minimum-time path planning for robot arms and their dynamics", Systems, Man and Cybernetics, IEEE Transactions on: 213-223.
- [39] **Chen, Y. ve Desrochers, A.,** (1988). "Time-optimal control of two-degree of freedom robot arms": IEEE.
- [40] **Leahy Jr, M. ve Saridis, G.,** (1987). "Compensation of unmodeled PUMA manipulator dynamics": IEEE.
- [41] **Cao, B. Doods, G. ve Irwin, G.W.,** (1994). "Time-optimal and smooth constrained path planning for robot manipulators": IEEE.
- [42] **Shiller, Z. Chang, H. ve Wong, V.,** (1996). "The practical implementation of time-optimal control for robotic manipulators", Robotics and Computer-Integrated Manufacturing, 12: 29-39.
- [43] **Moreno-Valenzuela, J.,** (2006). "Time-scaling of trajectories for point-to-point robotic tasks", ISA transactions, 45: 407-418.
- [44] **Kumagai, A. Kohli, D. ve Perez, R.,** (1996). "Near-minimum time feedback controller for manipulators using on-line time scaling of trajectories", Journal of Dynamic Systems, Measurement, and Control, 118: 300-308.
- [45] **Gasparetto, A. ve Zanutto, V.,** (2007). "A new method for smooth trajectory planning of robot manipulators", Mechanism and machine theory, 42: 455-471.
- [46] **Tarkiainen, M. ve Shiller, Z.,** (1993). "Time optimal motions of manipulators with actuator dynamics": IEEE.

- [47] **Rodriguez, G. and Jain, A.** (1991). Kinematics and Dynamics of Under-Actuated Manipulators, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation Sacramento, California - April 1991*.
- [48] **Siciliano, B., Sciavicco, L. etc.** (2000). *Modelling and Control of Robot Manipulators*, Springer-Verlag London Limited.
- [49] **Yesiloglu, S. M.** (2007). *High Performance Dynamical Modeling of Complex Topology Systems*, Ph. D. Thesis, Istanbul Technical University/ Graduate School of Science Engineering and Technology, Istanbul.

ÖZGEÇMİŞ

Ad Soyad : Mehmet Ünlüel
Doğum Yeri ve Tarihi : Kayseri, 1985
Adres : Sarıyer / İstanbul
E-Posta : munluel@gmail.com
Lisans : İstanbul Teknik Üniversitesi Makine Mühendisliği

