

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

STAUBLİ RX160 MANİPÜLATÖRÜN SENSÖRSÜZ KUVVET KONTROLÜ

YÜKSEK LİSANS TEZİ

Onur TAŞKIRAN

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

EYLÜL 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

STAUBLİ RX160 MANİPÜLATÖRÜN SENSÖRSÜZ KUVVET KONTROLÜ

YÜKSEK LİSANS TEZİ

**Onur TAŞKIRAN
(518121023)**

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Zeki Yağız BAYRAKTAROĞLU

EYLÜL 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 518121023 numaralı Yüksek Lisans Öğrencisi **Onur TAŞKIRAN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**STAUBLİ RX160 MANİPÜLATÖRÜN SENSÖRSÜZ KUVVET KONTROLÜ** ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Zeki Yağız BAYRAKTAROĞLU**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Ata MUĞAN**
İstanbul Teknik Üniversitesi

Yrd. Doç.Dr. Kadir ERKAN
Yıldız Teknik Üniversitesi

Teslim Tarihi : **26 Ağustos 2015**
Savunma Tarihi : **30 Eylül 2015**

Aileme,

ÖNSÖZ

Yapılan tez çalışmasında, Mekatronik Eğitim ve Araştırma Labaratuvarında bulunan Staubli Rx-160 manipülatörleri modellenip, kontrol edilmiştir. Tez boyunca göstermiş oldukları ilgi ve sabır için aileme teşekkürleri borç bilirim.

Tezimin başından itibaren benden bilgilerini ve yardımlarını esirgemeyen tez danışmanım olan Doç. Dr. Zeki Yağız Bayraktaroğluna teşekkür ederim.

Son olarak tez boyunca gerekli zamanlarda benden yardımlarını esirgemeyen tüm Mekatronik Eğitim ve Araştırma Merkezi çalışanlarına teşekkür ederim.

Ekim 2015

Onur Taşkiran
(Makine Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xix
SUMMARY	xxii
1. GİRİŞ	1
1.1 Tezin Amacı	2
1.2 Staubli RX-160 Manipülatör	3
1.2.1 Teknik özellikler ve ölçüler	3
1.2.2 Donanım.....	5
1.2.3 Yazılım.....	9
2. MODELLEME VE SİSTEM TANIMA	11
2.1 Geometrik Model	11
2.1.1 İleri geometrik model.....	11
2.1.1.1 Eklemlere eksen takımlarının atanması	12
2.1.1.2 Eksen takımları arasındaki dönme ve ötelenmeler	13
2.1.1.3 Euler açıları	16
2.1.2 Ters geometrik model	16
2.2 İleri Kinematik Model.....	21
2.2.1 Jakobiyen matris	23
2.3 Dinamik Model	27
2.3.1 Euler–Lagrange denklemleri.....	27
2.3.2 Newton-Euler denklemleri.....	30
2.3.3 Sistem tanıma.....	32
2.3.3.1 Sürtünme modeli	32
2.3.3.2 Yay modeli	37
3. KONTROL SİSTEMİ TASARIMI VE BENZETİMLER.....	45
3.1 Konum Kontrolü	45
3.1.1 Görev uzayında kontrolcü şeması	45
3.1.2 Yörünge planlama	49
3.1.3 Benzetim sonuçları.....	52
3.2 Sensörsüz Kuvvet Kontrolü	62
3.2.1 Hibrid konum/kuvvet kontrolü.....	67
3.2.1.1 Hibrid konum/kuvvet kontrolü benzetimi.....	69
3.2.2 Empedans kontrolü	75
3.2.2.1 Empedans kontrolü benzetimi.....	77
4. DENEYSELÇALIŞMALAR.....	83
4.1 Uygulama	83
4.2 Görev Uzayında Konum Kontrolü	84
4.2.1 Doğrusal yörüngede hareket - X doğrultusu	84

4.2.2 Doğrusal yörüngede hareket - Y doğrultusu	87
4.2.3 Doğrusal yörüngede hareket - X, Y ve Z doğrultusu	90
4.3 Görev Uzayında Sensörsüz Hibrid Konum/Kuvvet Kontrolü.....	93
4.3.1 Doğrusal hareket - Y doğrultusu, kuvvet kontrolü - Z doğrultusu	94
4.3.2 Doğrusal hareket - X ve Y doğrultusu, kuvvet kontrolü - Z doğrultusu ...	97
5. SONUÇ VE ÖNERİLER.....	101
KAYNAKLAR.....	103
EKLER.....	105
ÖZGEÇMİŞ.....	115

KISALTMALAR

LLI	: Alt Düzey Programlama Dili
PD	: Orantısal Türevsel Kontrol
PID	: Orantısal İntegral Türevsel Kontrol

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 1.1 : Staubli RX-160 robotunun teknik özellikleri	5
Çizelge 2.1 : Maksimum ve minimum açı değerleri.....	19
Çizelge 2.2 : Staubli RX-160 manipölatörünün sürtünme parametreleri	37

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Staubli RX-160 robotunun mekanik bölümleri.	3
Şekil 1.2 : RX-160 robotunun mekanik ölçüleri.	4
Şekil 1.3 : RX-160 Manipülatörünün çalışma uzayı.....	4
Şekil 1.4 : RX-160 Manipülatörünün donanımı.	5
Şekil 1.5 : RX-160 manipülatörüne ait iki farklı eklem mekanizması.	6
Şekil 1.6 : CS8C kontrolörü.....	7
Şekil 1.7 : El kumandası(pendant).	7
Şekil 1.8 : ATI marka kuvvet-tork sensörü.....	8
Şekil 1.9 : Transdüser ve kontrolör kutusu.	8
Şekil 1.10 : LLI çalışma yapısı.	9
Şekil 1.11 : LLI programına ait kontrol döngüleri.....	10
Şekil 2.1 : Staubli-RX160 manipülatörünün eksen takımları.	13
Şekil 2.2 : Kinematik ayrıştırma metodu.	17
Şekil 2.3 : Geometrik yaklaşım metodu.....	18
Şekil 2.4 : θ_2 ve θ_3 açılarını belirlemek için kullanılan geometrik yaklaşım metodu.	19
Şekil 2.5 : Birbirine komşu olan iki uzvun hız vektörleri.....	22
Şekil 2.6 : i. uzvun, robot uç noktasının hareketine olan etkisi.	25
Şekil 2.7 : Eklem hızları.....	26
Şekil 2.8 : Robot uç noktasına ait çizgisel ve açısal hızlar.	26
Şekil 2.9 : Newton denklemi ile elde edilen hareket.	30
Şekil 2.10 : Euler denklemi ile elde edilen hareket hareket.....	31
Şekil 2.11 : Newton-Euler denklemleri ile dinamik modeli elde edilen uzuv.....	32
Şekil 2.12 : Coulomb sürtünme modeli.	33
Şekil 2.13 : Coulomb ve viskoz sürtünme modeli.	34
Şekil 2.14 : Coulomb, viskoz ve statik sürtünme modeli	34
Şekil 2.15 : Sürtünme modelinde Stribeck etkisi.....	35
Şekil 2.16 : 5. ve 6.eklem arasındaki kinematik etkileşim.....	36
Şekil 2.17 : Yayın 2.eklemin hareketine bağlı oluşturduğu tork değerleri.	38
Şekil 2.18 : Yay modeli ile gerçek değerlerin karşılaştırılması.	38
Şekil 2.19 : Referans pozisyon yörüngesi.....	39
Şekil 2.20 : Referans hız yörüngesi.	39
Şekil 2.21 : Referans ivme yörüngesi	40
Şekil 2.22 : Merkezci ve Coriolis tork değerleri.....	40
Şekil 2.23 : Yer çekimi etkisi ile oluşan tork değerleri.	41
Şekil 2.24 : 2.eklemdeki yayın oluşturduğu tork değeri	41
Şekil 2.25 : Sürtünmeden dolayı eklemlerde oluşan tork değerleri.	42
Şekil 2.26 : Eylemsizlik matrisi ve ivme vektörünün çarpımı ile oluşan torklar.....	42
Şekil 2.27 : Referans hareketi yapabilmesi için eklemlere verilen torklar.	43
Şekil 3.1 : Eklemlerin kontrolüne dayanan görev uzayında ters dinamik kontrol....	46
Şekil 3.2 : Doğrudan görev uzayında ters dinamik kontrol.	47
Şekil 3.3 : Görev uzayında konum referans değerleri(üçüncü dereceden yörünge)...53	
Şekil 3.4 : Görev uzayında hız referans değerleri(üçüncü dereceden yörünge).....54	
Şekil 3.5 : Görev uzayında ivme referans değerleri(üçüncü dereceden yörünge).....54	
Şekil 3.6 : Referans eklem torkları(üçüncü dereceden yörünge).	55
Şekil 3.7 : Eklemlerin konum değerleri(üçüncü dereceden yörünge).....	55

Şekil 3.8 : Eklemlerin hız değerleri(üçüncü dereceden yörünge).	56
Şekil 3.9 : Eklemlerin ivme değerleri(üçüncü dereceden yörünge).	56
Şekil 3.10 : Robot uç noktası konumu(üçüncü dereceden yörünge).	57
Şekil 3.11 : Robot uç noktası referans kartezyen konumu(beşinci dereceden yörünge)	57
Şekil 3.12 : Robot uç noktası referans dönme konumu(beşinci dereceden yörünge).58	
Şekil 3.13 : Robot uç noktası referans hızları(beşinci dereceden yörünge).	58
Şekil 3.14 : Robot uç noktası referans ivmeleri(beşinci dereceden yörünge).	59
Şekil 3.15 : Referans eklem torkları(beşinci dereceden yörünge).	59
Şekil 3.16 : Eklem konumları(beşinci dereceden yörünge).	60
Şekil 3.17 : Eklem hızları(beşinci dereceden yörünge).	60
Şekil 3.18 : Eklem ivmeleri(beşinci dereceden yörünge).	61
Şekil 3.19 : Robot uç noktası kartezyen konumları(beşinci dereceden yörünge).	61
Şekil 3.20 : Tepki kuvvetinin sensör kullanılmadan hesaplanması..	63
Şekil 3.21 : Referans eklem konumları(üçüncü dereceden yörünge).	64
Şekil 3.22 : Referans eklem hızları(üçüncü dereceden yörünge).	64
Şekil 3.23 : Geri besleme tork sinyali(üçüncü dereceden yörünge).	65
Şekil 3.24 : Robot dinamik modeli kullanılarak elde edilen eklem torkları (üçüncü dereceden yörünge).	65
Şekil 3.25 : Robotun çevre ile teması sonucu eklemlerde oluşan tepki torku(üçüncü dereceden yörünge).	66
Şekil 3.26 : Tahmin edilen çevre kuvvet-momentleri (üçüncü dereceden yörünge). 66	
Şekil 3.27 : Hibrid konum/kuvvet kontrolü.	68
Şekil 3.28 : Robot uç noktası referans kartezyen konumu(beşinci dereceden yörünge).	69
Şekil 3.29 : Robot uç noktası referans dönme konumu(beşinci dereceden yörünge).70	
Şekil 3.30 : Robot uç noktası referans hızları(beşinci dereceden yörünge).	70
Şekil 3.31 : Robot uç noktası referans ivmeleri(beşinci dereceden yörünge).	71
Şekil 3.32 : Robotun çevreye uygulamasının istendiği referans kuvvet-moment değerleri(beşinci dereceden yörünge)..	71
Şekil 3.33 : Hibrit kuvvet-hareket kontrolü sırasında eklemlere uygulanan tork değerleri(beşinci dereceden yörünge).	72
Şekil 3.34 : Hibrid hareket kontrolü sırasında eklemlerde oluşan konumlar(beşinci dereceden yörünge).	72
Şekil 3.35 : Hibrid hareket kontrolü sırasında eklemlerde oluşan hızlar(beşinci dereceden yörünge).	73
Şekil 3.36 : Hibrid hareket kontrolü sırasında eklemlerde oluşan ivmeler(beşinci dereceden yörünge).	73
Şekil 3.37 : Hibrid hareket kontrolü sırasında robot uç noktasının çevresine uyguladığı kuvvet-moment değerleri(beşinci dereceden yörünge).	74
Şekil 3.38 : Hibrid hareket kontrolü sırasında robot eklemlerinde oluşan tepki torkları(beşinci dereceden yörünge).	74
Şekil 3.39 : Hibrid hareket kontrolü sırasında robot uç noktasının kartezyen konumları(beşinci dereceden yörünge).	75
Şekil 3.40 : Empedans kontrolü	77
Şekil 3.41 : Robot uç noktası referans kartezyen konumu(beşinci dereceden yörünge).	77
Şekil 3.42 : Robot uç noktası referans dönme konumu(beşinci dereceden yörünge).78	
Şekil 3.43 : Robot uç noktası referans hızları(beşinci dereceden yörünge).	78
Şekil 3.44 : Robot uç noktası referans ivmeleri(beşinci dereceden yörünge).	79

Şekil 3.45 : Referans eklem torkları(beşinci dereceden yörünge).	79
Şekil 3.46 : Eklem konumları(beşinci dereceden yörünge).	80
Şekil 3.47 : Eklem hızları(beşinci dereceden yörünge)..	80
Şekil 3.48 : Eklem ivmeleri(beşinci dereceden yörünge).	81
Şekil 3.49 : Robot uç noktası kartezyen konumları(beşinci dereceden yörünge).	81
Şekil 4.1 : LLI kullanıcı arayüz program akış şeması.	83
Şekil 4.2 : Uç noktaya ait referans ve geri besleme çizgisel konumları...	84
Şekil 4.3 : Uç noktaya ait referans ve geri besleme dönme konumları.....	85
Şekil 4.4 : Uç noktaya ait referans ve geri besleme hızları.....	85
Şekil 4.5 : Hareket esnasında eklemlere uygulanan torklar..	86
Şekil 4.6 : Hareket esnasında eklemlerin yapmış olduğu konum hareketleri.....	86
Şekil 4.7 : Hareket esnasında eklemlerin yapmış olduğu hız hareketleri..	87
Şekil 4.8 : Uç noktaya ait referans ve geri besleme çizgisel konumları..	87
Şekil 4.9 : Uç noktaya ait referans ve geri besleme dönme konumları.....	88
Şekil 4.10 : Uç noktaya ait referans ve geri besleme hızları.....	88
Şekil 4.11 : Hareket esnasında eklemlere uygulanan torklar.	89
Şekil 4.12 : Hareket esnasında eklemlerin yapmış olduğu konum hareketleri.	89
Şekil 4.13 : Hareket esnasında eklemlerin yapmış olduğu hız hareketleri.	90
Şekil 4.14 : Uç noktaya ait referans ve geri besleme çizgisel konumları.	90
Şekil 4.15 : Uç noktaya ait referans ve geri besleme dönme konumları.	91
Şekil 4.16 : Uç noktaya ait referans ve geri besleme hızları.....	91
Şekil 4.17 : Hareket esnasında eklemlere uygulanan torklar.	92
Şekil 4.18 : Hareket esnasında eklemlerin yapmış olduğu konum hareketleri.....	92
Şekil 4.19 : Hareket esnasında eklemlerin yapmış olduğu hız hareketleri.	93
Şekil 4.20 : Görev uzayında uç noktanın yapmış olduğu çizgisel hareket.	94
Şekil 4.21 : Görev uzayında uç noktanın yapmış olduğu hızlar.	94
Şekil 4.22 : Görev uzayında yapılan hareket sırasında eklem konumları.....	95
Şekil 4.23 : Hareket sırasında sensörlerden ölçülen eklem torkları.....	95
Şekil 4.24 : Uç noktanın çevresi ile teması sırasında eklemlerde oluşan tepki torkları.	96
Şekil 4.25 : Kuvvet-Tork sensöründen ölçülen ve tahmin edilen tepki kuvveti.....	96
Şekil 4.26 : Görev uzayında uç noktanın yapmış olduğu çizgisel hareket.	97
Şekil 4.27 : Görev uzayında uç noktanın yapmış olduğu hızlar.	97
Şekil 4.28 : Görev uzayında yapılan hareket sırasında eklem konumları.....	98
Şekil 4.29 : Hareket sırasında sensörlerden ölçülen eklem torkları.....	98
Şekil 4.30 : Uç noktanın çevresi ile teması sırasında eklemlerde oluşan tepki torkları.	99
Şekil 4.31 : Kuvvet-Tork sensöründen ölçülen ve tahmin edilen tepki kuvveti.....	99
Şekil A1.1 : İleri geometrik model benzetimi(Robot ilk konumunda).	106
Şekil A1.2 : İleri geometrik model benzetimi(Robot ikinci konumunda).	106
Şekil B.1 : Ters geometrik model benzetimi.	109
Şekil C1.1 : İleri kinematik model benzetimi.....	110
Şekil D.1 : Ters dinamik model benzetimi..	112
Şekil D.2 : H bloğu içerisinde oluşan torklar.....	112
Şekil E.1 : Görev uzayında konum kontrol benzetimi.....	113
Şekil F.1 : Sensör kullanılmadan çevre kuvvetinin bulunuşuna ait benzetim	114

STAUBLI RX160 MANİPULATÖRÜN SENSÖRSÜZ KUVVET KONTROLÜ

ÖZET

Son yıllarda gelişen teknoloji ile endüstriyel robotların saniyede kullanımı oldukça artmaktadır. Saniyede kullanılan robotlar sayesinde firmalar zamandan, iş gücünden, maliyetten tasarruf ederken, hem de ürettikleri ürünleri daha kaliteli ve hassas üretebilmektedirler. Sanayide kullanılan endüstriyel seri manipülatörlerin çoğu gerek düşük maliyeti gerekse de kolay uygulanabilirliği nedeni ile geleneksel PID(orantısal, integral, türev) kontrol stratejisi ile sabit kazanç katsayıları değiştirilerek kontrol edilmektedir. Fakat bu yöntem robotların doğrusal olmayışı ve eklemlerin bir birine bağlantılı olması nedeniyle hızın arttığı çalışma aralıklarında iyi sonuçlar vermemektedir. Diğer bir taraftan ise sadece konum kontrolü strateji ile endüstriyel robotları kontrol etmek nokta kaynağı, spreyci boyama yada hafif yüklerin bir yerden bir yere taşınması sırasında daha çok robot uç noktasının uzayda serbest hareketinin önemli olduğu uygulamalarda başarı sağlamaktadır. Lakin çapak alma, taşıma ve montaj gibi işlemler sırasında robot uç noktasının çevresi ile ciddi etkileşimlere girdiği durumlarda pozisyon kontrolü yerine empedans, hibrid gibi hem konum hem de kuvvet kontrol algoritmalarının beraber uygulandığı kontrolörler önem kazanmaktadır.

Bu tezde, İTÜ Mekatronik Eğitim ve Araştırma Merkezinde bulunan altı serbestlik dereceli Staubli RX-160 manipülatörleri kullanılarak robotun geometrik, kinematik ve dinamik modeli elde edilip, konum ve sensörsüz kuvvet kontrolleri gerçekleştirilmektedir. Yapılan tez başlıca 5 bölümden oluşmaktadır. İlk bölümde tez çalışması sırasında kullanılan Staubli RX-160 manipülatörünün hem donanım hem de yazılım kısmı anlatılmaktadır.

İkinci bölümde Staubli firmasından alınan ölçü ve parametreler sayesinde Matlab programı kullanılarak, robotun önce ileri geometrik modeli sonra ise ters geometrik modeli elde edilmektedir. Elde edilen ileri geometrik model sayesinde robot eklemlerinin hızları ile uç noktanın hızı arasındaki ilişkiyi veren Jakobiyen matris elde edilmektedir. Daha sonra Staubli firmasından alınan robot uzuvlarına ait kütle ve eylemsizlik matrisleri kullanılarak robotun dinamik modelini açıklayan Euler-Lagrange denklemleri elde edilmektedir. Bu bölümde en son olarak analitik denklemlerden elde edilen robot dinamiğine sürtünme ve yay modelleri eklenerek robotun nihai dinamik modeli kurulmaktadır.

Üçüncü bölümde tez sırasında robota uygulanan kontrol algoritmaları elde edilmektedir. Elde edilen ilk kontrol algoritması, görev uzayında uygulanan ters dinamik konum kontrolüdür. Ayrıca kontrol algoritmalarında referans giriş olarak kullanılan yörüngeler de bu bölümde tanımlanmaktadır. Robot ile çevresi arasında temas olduğunda, tepki kuvveti oluşmaktadır. Oluşan tepki kuvvet ve moment değerleri genellikle kuvvet-moment sensörleri ile elde edilmektedir. Fakat tez sırasında tepki kuvvet ve momentleri farklı olarak bozucu tork denklemi kullanılarak elde edilmektedir. Son olarak, kuvvet kontrol algoritmaları olan hibrid konum/kuvvet ve empedans kontrol algoritmaları açıklanıp, benzetimleri yapılmaktadır.

Dördüncü bölüm, ikinci ve üçüncü bölümlerin sonunda elde edilen model ve kontrol algoritmalarının deneysel olarak doğrulandığı bölümdür. Konum ve sensörsüz kuvvet algoritmalarının deneysel sonuçları bu bölümde ayrı ayrı elde edilmektedir.

Son bölüm sonuç bölümü olup, tez sonunda elde edilen gözlemler, yaşanan zorluklar anlatılıp, çeşitli sonuçlar çıkarılmaktadır.

SENSORLESS FORCE CONTROL OF STAUBLI RX160 MANIPULATOR

SUMMARY

In recent years, using of industrial robots in industry has increased thanks to developing technology. Due to robots used in industry, companies can save time, manpower and cost also they can manufacture their products more quality and precise. However most of serial manipulators used in industry are controlled via traditional PID(proportional ,integral ,derivative) control strategies with changing constant gains because of its low cost and easiness. But this approach doesn't give good results when the work of robot is fast because of its nonlinear and coupling effects. On the other hand controlling of industrial robots using only position control strategies can have success in application of point welding, spray painting or pick and place, where free motion of end effector of robot is important. But during such an application like deburring, grinding and assembly, where the end effector of robot and its environment are in interaction should be controlled with both position and force controls algorithms like impedance and hybrid control instead of using only position control.

In this thesis, using Staubli RX-160 manipulators in ITU Mechatronics Education and Research Center, geometric, kinematics and dynamics model of robot have been obtained and then both position and sensorless force control have been applied. This thesis mainly consists of five chapters. In the first chapter, both hardware and software of used Staubli RX-160 manipulator are explained.

In the second chapter, with taken dimensions and parameters from Staubli RX-160, firstly the forward and inverse geometric models of the robot are found in Matlab program and then helping forward geometric model, Jacobian matrix which explains relation between velocities of robot joints and end effector is obtained. Secondly, thanks to mass and inertia matrices taken from Staubli Company, the robot is modelled dynamically in Matlab using Euler-Lagrange equations and Newton Euler equations. Finally adding friction model and spring model to dynamics model, final model of robot has been obtained. Then, to test success of the kinematics and dynamics model of robot, Matlab Simulink has been used. At the end of second chapter. Two dynamics models are compared each other.

In the third chapter, control algorithms applied to the robot are obtained. These algorithms are separated into two parts. First of them is motion control in task space. The second one is compliant motion control. To realize motion control in the task space, the trajectory in the task space is generated. Using geometric, kinematics, dynamics models and trajectory in the task space, the computed torque control is applied to the robot. The method of the sensorless force control is explained in the beginning of the compliant motion segment. Using this method, reaction forces and moments on the end effector of robot is found. So we don't have to use any force-moment sensor in compliant motion of the robot. In the thesis, two different control is used to execute compliant motion. The first control is hybrid motion-force control. Using this control, we manage to control both position of robot and force between the end effector and the environment. The second control method is impedance control. In this control, the end effector of robot is assumed to be a spring with mass and damper. The last segment of the third chapter. Both position control and force

control is simulated in Matlab Simulink. Showing different Simulink plots, results of control algorithms applied to robot can be seen.

In the fourth chapter, lots of experimental tests are applied to Staubli Rx-160 manipulators. During experiments, the low level program, named LLI, is used to realize position and force control algorithms. LLI is a kind of C libraries. LLI program provides users four feedback signals and for commands at every four milliseconds. Respectively, These feedback signals are joint positions, joint position errors, joint velocities and joint torques. The command signals are joint positions, joint velocities, joint forward torques and joint torques. Also the the feedback signal of force-moment on the end effector can be used at every four milliseconds in the compliant motion control. The first applied control method to the robot is computed torque control in joint space. In the computed torque control, the acceleration of the robot is used as a control signal. To determine this control signal, PID method is used. Changing proportional, derivative and integral gains, the performance of robot is achieved. Because of nonlinearities and coupling effects of joints, the forward dynamic model is used to compensate these effects. These model parts are gravity, Coriolis and centrifugal, spring and friction torques. After achieving position control in joint space, the computed torque control in task space is achieved. In this control, the position and velocity of the end effector is used directly. The acceleration of the end effector is used as a control signal. After, this acceleration is converted into the acceleration of the joints using Jacobian matrices. During the experiments, the robot is controlled in the direction of x, y and z. So the robot is followed to desired position of the end effector coordinates in space. Before the experiments of force control, experimental setup is constructed. A device which includes a spring and ball is attached to end effector of the robot so the robot can't be damaged by its environment. The first control algorithm in the force control is hybrid control. In this control, robot can be controlled by using both position and force control. To select which robot axis will be controlled using position control algorithm or force control algorithm, a selection matrix is used. If the corresponding diagonal element of the matrix is 1, the axis is controlled in position control algorithm, otherwise it is controlled in force control algorithm. In hybrid control tests, the motion of robot is controlled by using computed motion controller until it reaches the surface, and then both force between end effector and its environment and motion of robot is controlled using hybrid controller. Also by using hybrid controller, the success of the sensorless force control algorithm is measured. First, the force between the end effector and environment is controlled using ATI force-moment sensor. Second, instead of using sensor, the force is controlled by using sensorless force control algorithm. To compare two different control algorithm, the data provided by LLI is used and plotted in Matlab. The last compliance control algorithm is the impedance control. In this control, the end effector of robot is assumed to be a spring-mass-damper device. When the robot contacts its environment its impedance is changed.

The last and fifth chapter. The results of thesis and suggestion about the future works are mentioned in this chapter. Because the works with using Staubli RX-10 robots is related each other. The useful and critical advices is given the next working about

these robots. Especially friction model of robot can be improved by identification methods. The use of this thesis, several scenarios like deburring, assembling can be applied the robot. Also at the end of the thesis, lots of Simulink block diagrams used in the thesis and Matlab codes are shared.

1. GİRİŞ

Robotlar elektronik, mekanik, yazılım ve kontrol bilimlerinin hepsini bünyesi içerisinde barındıran mekatronik ürünlerdir. Robotlar günümüzde uzay, savunma, otomotiv, imalat, nükleer, tıp, kimya gibi sektörlerde kendilerini hızla göstermeye başlamışlardır. Robotlar tasarımlarına veya kullanıldıkları yerlere göre çeşitli sınıflara ayrılmaktadırlar. Bu tezde ele alınacak olan robot tipleri endüstriyel seri robotlardır. Endüstriyel seri robotlar, işlev ve şekil bakımından insan koluna benzeyen prizmatik veya döner mafsallardan oluşan mekanizmalardır. Endüstriyel seri manipülatörler sensörleri vasıtası ile çevresindeki değişiklikleri algılayabilen, kontrolörleri ile karar verip, bu kararları eyleycilerine elektriksel sinyal olarak gönderip, bir tepki oluşturabilen elektromekanik cihazlardır. Çok hızlı ve hassas çalışabilmeleri, zamandan ve insan gücünden tasarruf yapabilmelerinin yanı sıra insanlar için tehlikeli olabilecek görevleri rahatlıkla yerine getirebildikleri için robotların, zamanla sanayideki sayıları ve etkinlikleri artmaktadır. Bir robotun başarı ölçütlerini belirleyen en önemli alt sistemi kontrolcüsüdür. Robotlar kontrol performanslarına göre daha hassas, güvenilir, tekrarlanabilir ve hızlı çalışabilmektedirler. Günümüz teknolojisinde robotlara uygulanan çeşitli kontrol yöntemleri bulunmaktadır. Bunlar geleneksel PID konum kontrolden, kuvvet kontrolüne değişmektedir. Burada robota uygulanacak kontrol algoritması robotun çalışma ortamına, maliyetine, istenen çalışma hızı ve ivme değerlerine, kullanılan eyleycilerine, sensörlerine ve yapacağı işe göre belirlenmektedir. Örneğin kolaylık ve basitlik istenen yerlerde klasik olarak PID yöntemi kullanılarak robot kontrolü gerçekleştirilmektedir. Ama bu yöntem operasyon hızının yüksek olduğu veya kullanıcının robottan hassasiyet beklediği durumlarda dezavantajlara sahiptir. Çünkü endüstriyel robotların dinamik yapısı doğrusal değildir. Ayrıca seri robotların eklemleri birbirleriyle etkileşim halindedir. Robotun doğrusal olarak davranmasını sağlamak amacı ile hesaplamalı tork kontrolü(ters dinamik kontrol) kullanılmaktadır. Bu kontrol yönteminde robotun ters dinamik modeli kullanıldığı için robotun dinamik modeli hassas bir şekilde elde edilmelidir. Montaj, çapak alma gibi

işlemlerde ise robotun sadece konumunun kontrol edilmesi yeterli olmayabilir. Robotun çevreyle olan etkileşimi, robotun uç noktasına takılan kuvvet-moment sensörü ile belirlenerek robotun hem konum hem de kuvvet kontrolü gerçekleştirilmektedir. Bu tür kontrol yöntemlerinde ise kullanılan kuvvet-moment sensörleri problem yaratabilmektedirler. Bu problemin nedenleri, kullanılan sensörlerin ağırlığı, maliyeti veya sensörlerin sıcaklık gibi parametrelere bağlı olarak değişimi olabilmektedir. Bunun için önerilen yöntem kuvvet-moment sensörünü robottan çıkararak kuvvet kontrol algoritmalarını gerçekleştirmektir. Bu tezde yukarıda anlatılan kontrol algoritmalarını gerçekleştirebilmek için ilk önce, robotun geometrik ve kinematik modelleri elde edilmektedir. Daha sonra bulunan geometrik ve kinematik modeller yardımı ile robotun dinamik modeli Euler-Lagrange denklemleri ile bulunmaktadır. Dinamik model elde edildikten sonra görev uzayında ters dinamik tork kontrol algoritması robota uygulanmaktadır. En son olarak, sensörsüz kuvvet kontrol algoritmaları uygulanmaktadır. Kontrol algoritmaları hem Matlab Simulink ortamında, hem de deneysel olarak Staubli RX-160 robotu üzerinde gerçekleştirilip, test edilmektedir.

1.1. Tezin Amacı

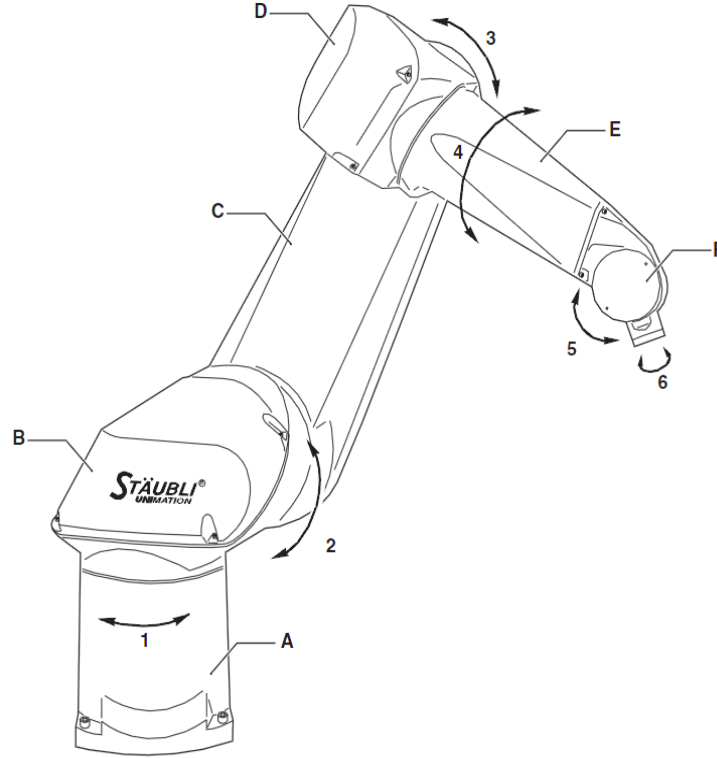
İTÜ Mekatronik Eğitim ve Araştırma Merkezinde lisans, yüksek lisans ve doktora araştırmalarında kullanılmak üzere toplam 3 tane Staubli-RX160 manipülatörü mevcuttur. Bugüne kadar yapılan çalışmaların aksine robot dinamik modeli farklı olarak Euler-Lagrange denklemleri kullanılarak robotun gerçeğe yakın modeli elde edilmektedir. Analitik olarak elde edilen dinamik modele ek olarak, önceki yıllarda robot üzerinde yapılan deneysel çalışmalarla elde edilen sürtünme ve yay modelleri de kullanılmaktadır(Egemen Zengin,2013). Daha sonra ise hem ters dinamik kontrolü, hem de sensörsüz kuvvet kontrolü algoritmaları doğrudan robot üzerinde deneysel olarak uygulanmaktadır. Bu çalışma ile gelecekte İTÜ öğrencilerinin Staubli RX-160 robot üzerinde sahip olacağı bilgiler arttırılmış olup, robot üzerinde yapılabilecek yeni çalışmalara temel oluşturabilecektir. Öte yandan, üniversite bünyesinde çalışılmakta olan insansı robotlar için, tezde kullanılacak kontrol algoritmaları referans oluşturabilecektir.

1.2 Staubli RX-160 Manipulatörü

Tez sırasında gerek elde edilen geometrik, kinematik ve dinamik modeller gerek oluşturulan kontrol algoritmaları için Staubli RX-160 manipulatörü kullanılmaktadır. Bu sebeple tezin ilk bölümünde ilk önce, kullanılan robota ait teknik özellik ve ölçüler sonra ise, robot donanımı ve yazılımı hakkında bilgiler verilmektedir.

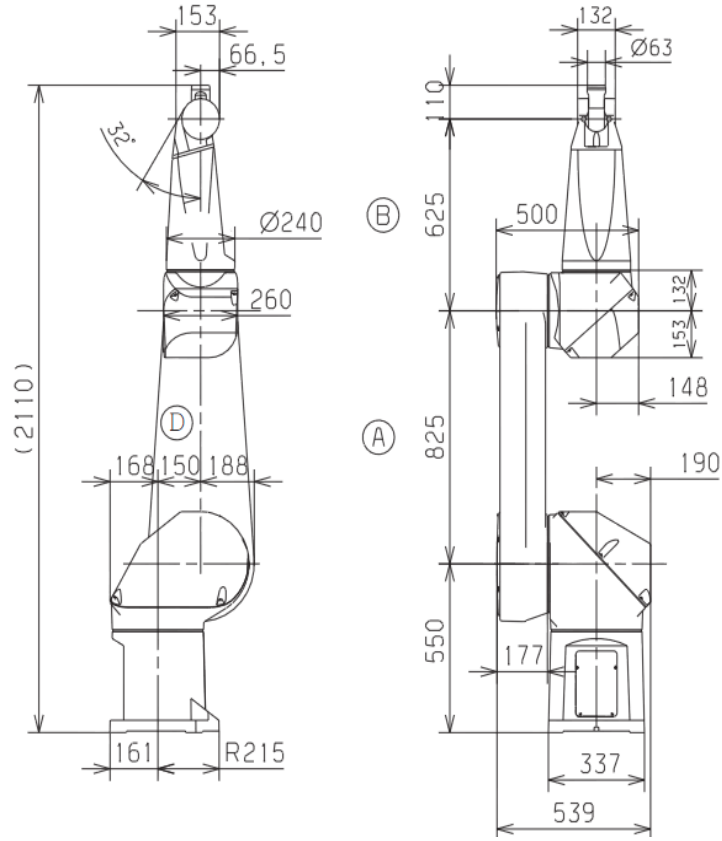
1.2.1 Teknik özellikler ve ölçüler

Staubli RX-160 manipulatörü birbiri ile ilişkide olan 6 döner eklemden oluşmaktadır. Bu eklemler sayesinde robot aynı insan kolu gibi 3 tane öteleme 3 tane de dönme hareketi yapabilmektedir. Robot şekil 1.1’de gösterildiği gibi altı bölüme ayrılırsa, A ile gösterilen kısım taban, B ile gösterilen kısım omuz, C ile gösterilen kısım kol, D ile gösterilen kısım dirsek, E ile gösterilen kısım ön kol ve son olarak F ile gösterilen kısım ise bilek mekanizmasını temsil etmektedir.

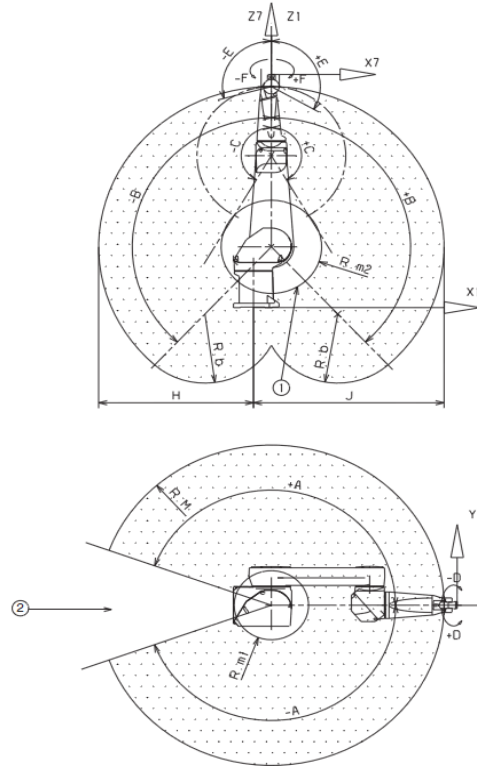


Şekil 1.1: Staubli RX-160 robotunun mekanik bölümleri [1].

Sırası ile, şekil 1.2’de RX-160 robotunun mekanik ölçüleri, şekil 1.3’de ise robotun çalışma uzayındaki yapabileceği maksimum hacim verilmektedir.



Şekil 1.2: RX-160 robotunun mekanik ölçüleri [1].



Şekil 1.3: RX-160 Manipülâtörünün çalışma uzayı [1].

Manipülâtörün tasarım özellikleri olarak, nominal hız koşulunda taşıyabileceği maksimum yük 20 Kg olup, manipülâtör ağırlığı toplam 248 Kg'dır. Robotun çalışma aralığı, nominal ve maksimum hızları, çözünürlüğü gibi teknik özellikleri ise çizelge 1.1'de gösterilmektedir.

Çizelge 1.1: Staubli RX-160 robotunun teknik özellikleri [1].

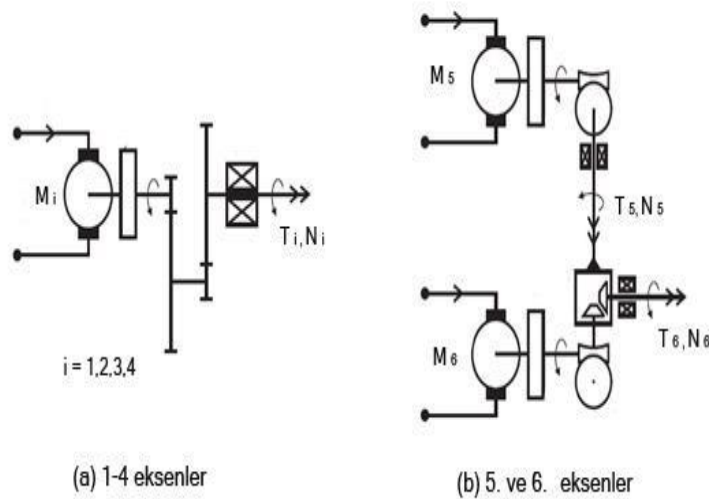
Eksen	1	2	3	4	5	6
Dönme açısı (°)	320	275	300	540	225	540
Çalışma aralığı dağılımı (°)	±160	±137.5	±150	±270	+120 -105	±270
Nominal hız (°/s)	165	150	190	295	260	440
Maksimum hız (°/s)	200	200	255	315	390	870
Açısal çözünürlük (°. 10^{-3})	0.042	0.042	0.054	0.062	0.12	0.17

1.2.2 Donanım

RX-160 manipülâtörünün donanım kısmı çeşitli alt bölümlerden oluşmaktadır. Bunlar; manipülâtör mekaniği, eyleyiciler, sensörler, kontrolör, el kumandası ve parça tutucudur. Ayrıca bu mekanizmaların yanında robota harici olarak eklenebilen kuvvet-moment sensörü ile bilgisayar da mevcuttur. Robot donanımının alt bölümleri şekil 1.4'de gösterilmektedir.



Şekil 1.4: RX-160 Manipülâtörünün donanımı [2].



Şekil 1.5: RX-160 manipülatörüne ait iki farklı eklem mekanizması [2].

Staubli RX-160, 6 serbestlik dereceli bir robottur. Robota bu serbestlik derecesini kazandıran her bir eklemden bulunan dönme eksenlerdir. Bu eksenler, Kartezyen koordinatlarına göre x, y veya z eksenlerindeki bağımsız dönel hareketleri nitelemektedirler. Robotta oluşturulan bu altı eksen tahrik eden elektromekanik mekanizmalar ise üç fazlı servo motorlardır.

Kapalı çevrim robot kontrol algoritmalarında geri besleme sinyali olarak eklemlerin konum ve hız büyüklükleri kullanılmaktadır. Bu fiziksel büyüklükleri elektrik sinyaline çevirip robot kontrolörüne ileten cihazlar sensörlerdir. RX-160 robotunda eklemlerin konum ve hız bilgisini ölçmek için konuşlandırılmış enkoderler bulunmaktadır.

RX-160 manipülatörünün kontrolcüsü CS8C'dir. Robot kontrolcüsü diğer elektriksel ekipmanların da bulunduğu elektrik kabinine monte edilmiştir. Kontrolör, gerçek zamanlı işletim sisteminde çalışabilen bir endüstriyel bilgisayar, güvenlik donanımları, motorları süren güç amplifikatörleri, AC-DC dönüştürücü, üzerindeki

mantıksal devreleri beslemek için ARSP güç kaynağı, Ethernet, seri ve haberleşme portlarından oluşmaktadır. Şekil 1.6’da CS8C kontrolörü gösterilmektedir.



Şekil 1.6: CS8C kontrolörü [2].

Pendant(el kumandası) manipülatör kontrolcüsü ile operatör arasındaki iletişimi sağlayan mekanizmadır. İstendiği takdirde pendant üzerinden robota güç verilip kesilebilmektedir. Robot eklem uzayında ve çalışma uzayında istenilen hızda pendant kullanılarak hareket ettirilebilmektedir. El kumandası manipülatörün ulaşması istendiği noktaları tanıtarak robotu programlayabilmektedir. Ayrıca üzerinde acil kontrol butonu bulundurmaktadır. Şekil 1.7’de Staubli RX-160 manipülatörünün el kumandası gösterilmektedir.



Şekil 1.7: El kumandası(pendant) [2].

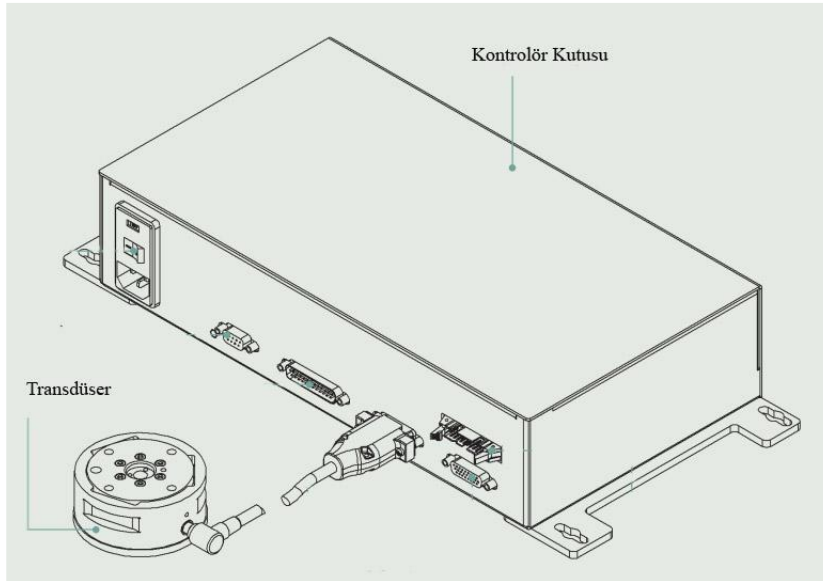
Robotun uç noktasına eklenen parça tutucu(gripper) sayesinde iş parçaları bir noktadan bir noktaya güvenle taşınabilmektedirler. Parça tutucular elektriksel veya pnömatik olarak tahrik edilebilmektedirler.

Robota harici olarak eklenebilen elemanlardan biri ATI kuvvet-moment sensörüdür. Robotun çevresiyle fiziksel olarak teması halinde konum kontrolü ile birlikte kuvvet

kontrolü de uygulanmalıdır. Kuvvet kontrol algoritmasında geri besleme sinyali olarak, robot uç noktasının çevresi ile teması sırasında oluřturduđu 3 tane kuvvet(F_x, F_y, F_z) ve 3 tane moment(M_x, M_y, M_z) fiziksel büyüklüđu kullanılmaktadır. Robot kontrolörüne bu deđerlerin iletilmesi için robot uç noktasına ATI marka kuvvet-moment sensörü monte edilmektedir. řekil 1.8’de gösterilen kuvvet-moment sensörü toplam iki parçadan oluřmaktadır. Bu kısımlar; transdüser ve kontrolördür. řekil 1.9’da gösterilen ATI marka sensöre ait kontrolör, transdüserden aldığı sinyali işleyerek, robot kontrolörüne göndermektedir.



řekil 1.8: ATI marka kuvvet-tork sensörü.



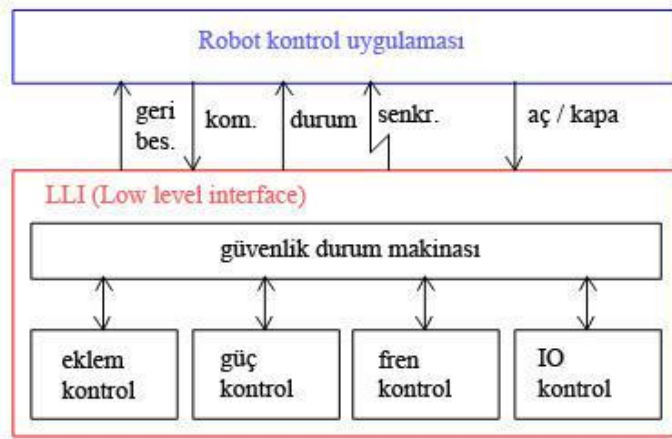
řekil 1.9: Transdüser ve kontrolör kutusu [3].

1.2.3 Yazılım

Staubli firmasının kullanıcılara sunduğu iki farklı yazılım programlama dili mevcuttur. Bunlardan ilki, VAL3 olarak adlandırılan endüstriyel uygulamalarda robotu programlamak için kullanılan üst seviye programlama dilidir. Diğeri ise LLI(Low Level Interface) olarak adlandırılan C ve C++ kütüphanelerinden oluşan alt seviye programlama dilidir.

Manipülâtör programlama dillerinin en eski üyelerinden olan VAL3, üst seviye bir programlama dilidir. Kullanıcıların endüstriyel uygulamalarda robotu kolaylıkla programlamasına olanak sağlamaktadır.

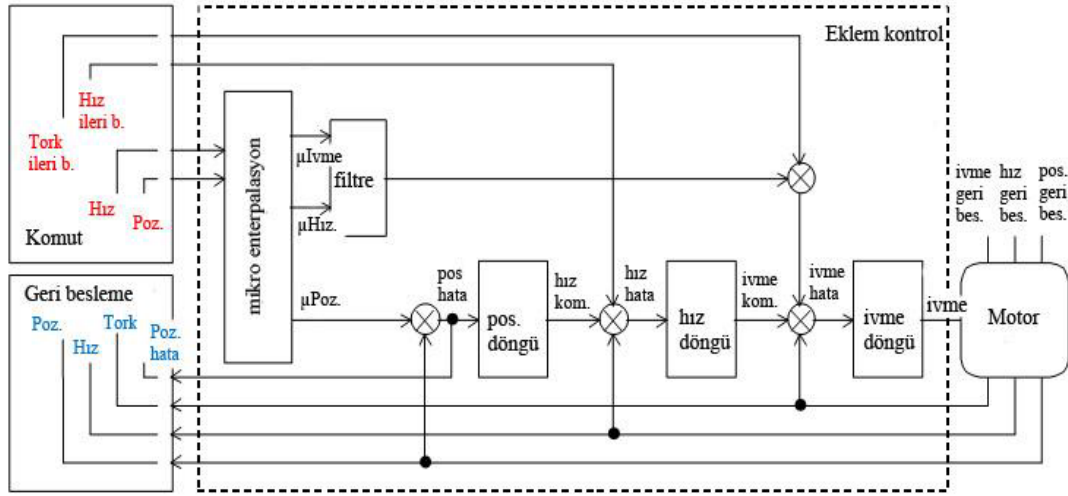
LLI programı RX-160 endüstriyel manipülâtörünü kontrol etmek için kullanılan bir C,C++ kütüphanesidir. VAL3 programlama dilinin tersi olarak alt seviye bir programlama dilidir. Şekil 1.10'da LLI programının çalışma yapısı gösterilmektedir. LLI programı ile robotun 4 geri besleme sinyali kullanılabilmektedir. Bunlar sırası ile konum, hız, konum hatası ve tork geri besleme sinyaleridir. Robota komut olarak da konum, hız, hız ileri besleme ve tork ileri besleme sinyalleri gönderilebilmektedir. Bu işlemlerin yanı sıra LLI ile geri besleme ve komut büyüklüklerinin çalışma sırasında aldığı değerler yazdırılabilmektedir. Motorlara güç verilip kesilmesi işlemi de LLI alt seviye programı ile gerçekleştirilebilmektedir.



Şekil 1.10: LLI çalışma yapısı [4].

LLI programının çalışma şeması 3 farklı kontrol döngüsüne ayrılabilir. Bunlar; konum kontrol döngüsü, hız kontrol döngüsü ve tork kontrol döngüsüdür. Bu kontrol döngülerine her 4 mili saniyede bir geri besleme sinyalleri gönderilip, komutlar alınmaktadır. Böylece robot gerçek zamanlı olarak kontrol

edilebilmektedir. Şekil 1.11’de yukarıda bahsedilen LLI programına ait kontrol döngüleri gösterilmektedir.



Şekil 1.11: LLI programına ait kontrol döngüleri [4].

2. MODELLEME VE SİSTEM TANIMA

Tez sırasında robota uygulanacak olan kontrol algoritmalarına geçilmeden önce, sırası ile Staubli RX-160 robotuna ait geometrik, kinematik ve dinamik modeller elde edilmektedir. Geometrik model ile robot hareketine neden olan moment ve kuvvetler düşünülmeksizin, eklemler ve robot uç noktası arasındaki öteleme ve dönme bağıntıları çıkarılmaktadır. İkinci olarak, eklem ve uç nokta arasındaki hız ilişkilerini veren kinematik model oluşturularak, Jakobiyen matris bulunmaktadır. Sonraki kısımda elde edilen geometrik ve kinematik modeller kullanılarak, harekete neden olan kuvvetlerin ve momentlerin bulunduğu dinamik model elde edilmektedir. Denklemler yardımı ile analitik olarak bulunan dinamik model, robot hareketini açıklamada yetersiz kaldığı için son bölümde, sistem tanımlanarak oluşturulan(Egemen Zengin,2013) sürtünme ve yay modelleri dinamik modele eklenmektedir.

2.1 Geometrik Model

Hem kinematik model, hem de dinamik model elde edilirken robotun geometrik modeline ihtiyaç vardır. Öte yandan ileri bölümlerde görev uzayda uygulanan kontrol algoritmalarında da geometrik modele gereksinim duyulmaktadır. Bu nedenle, RX-160 manipülatörüne ait ileri ve geri geometrik modeller Matlab programı kullanılarak bulunmaktadır.

2.1.1 İleri geometrik model

İleri geometrik modelin amacı, eklemlerin açısal yer değiştirme hareketlerini kullanarak robot uç noktasının uzaydaki konumunu belirlemektir. Bir endüstriyel seri robot n tane eklemden ve $n+1$ tane uzuvdan oluşmaktadır. Uzuvlar taban noktasından başlayarak 0'dan n 'e, eklemler ise 1'den n 'e kadar numaralandırılmaktadırlar. i . eklem hareket ettirilirse i . uzuv hareket etmektedir. Böylece 0.uzuv hareketsiz kabul edilmektedir. i . eklemin değişkeni q_i olarak tanımlanmaktadır. Bir seri robotun eklemi ya dönel, ya da prizmatiktir. Staubli RX-160 robotu toplam 6 tane dönel

eklemden oluşmaktadır. Böylece Staubli RX-160 manipülatörünün eklem değişkenleri $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ ve θ_6 açı değerleri olmaktadır.

2.1.1.1 Eklemlere eksen takımlarının atanması

İleri geometrik modeli oluşturabilmek için her robot uzvuna eksen takımları atanmaktadır. Örneğin i . uzva o_i, x_i, y_i, z_i adlı eksen takımı atanmaktadır. Böylece i . eklem hareket ettirildiğinde i . uzuvda, ona atanmış eksen takımı da aynı harekete maruz kalmış olur. o_0, x_0, y_0, z_0 eksen takımı robotun taban noktasına atanmaktadır. Robot üzerinde son eksen takımı olan n . eksen takımı da robot uç noktasına atanarak eksen atama işlemi tamamlanmaktadır.

Eksen takımları atanırken belirli kurallara göre atanmaktadır. Bu kural, Denavit-Hartenberg kuralı olarak adlandırılmaktadır. Eksen atama işlemine z eksenlerinin belirlenmesi ile başlanmaktadır. z_i eksen robotun $i+1$. ekleminin dönme eksenine olacak şekilde atanmaktadır. Böylece z_0 eksen, 1. eklemin dönme eksenine, z_1 eksen ise, 2. eklemin dönme eksenine yerleştirilir.

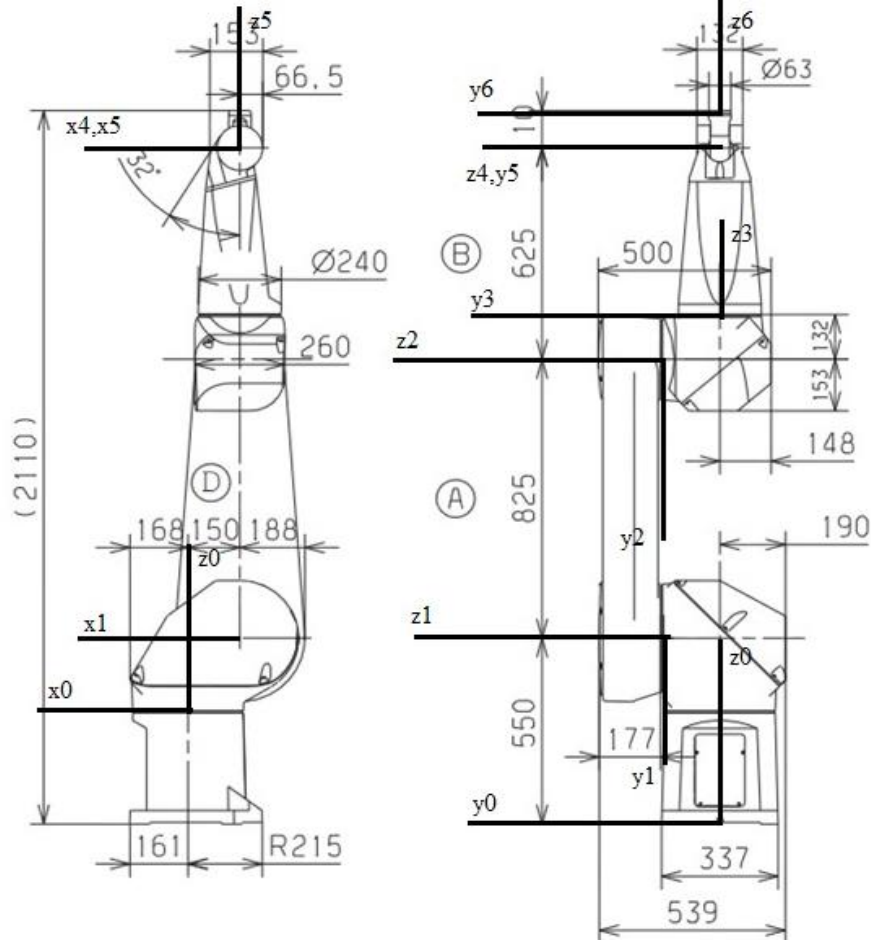
Denavit-Hartenberg kuralına göre, z eksenleri belirlendikten sonra, z eksenlerinin geometrik konumuna göre üç özel durum bulunmaktadır.

(i) z_{i-1} ve z_i eksenlerinin eş düzlemsel olmadığı durumlar: Bu durumlarda z_{i-1} ve z_i eksenleri arasında bu iki eksene dik olan özel bir dikme bulunmaktadır. Bu dikmeye özel olarak, ortağın dikmesi ismi verilmektedir. Ayrıca bu dikme z_{i-1} ve z_i eksenleri arasındaki en kısa mesafedir. Bu dikmenin z_i ile kesiştiği noktaya o_i merkezi ve x_i eksenine atanmaktadır. Son olarak, sağ el kuralını tamamlayacak şekilde y_i eksen seçilerek, o_i, x_i, y_i, z_i eksen takımı atama işlemi tamamlanmaktadır.

(ii) z_{i-1} ve z_i eksenlerinin birbirine paralel olduğu durumlar: Bu durumda iki eksen arasında sonsuz sayıda ortak dikme vardır. o_i eksen merkezi, z_i eksen üzerinde olacak şekilde keyfi olarak herhangi bir noktaya atanmaktadır. Son olarak o_i eksen merkezi üzerinde olacak şekilde x_i ve y_i eksenleri sağ el kuralına göre atanmaktadır.

(iii) z_{i-1} ve z_i eksenlerinin birbirleri ile kesiştiği durumlar: Genellikle, bu durumun olduğu yerlerde o_i merkez noktası, bu iki eksenin kesiştiği noktaya yerleştirilmektedir. Son olarak x_i ve y_i eksenleri de sağ el kuralına uyacak şekilde atanmaktadır.

Yukarıda yer alan kurallar çerçevesinde 0'dan n-1'e kadar olan eksen takımları atandıktan sonra, son olarak n uzuvlu robotun n. eksen takımı atanmaktadır. Bu eksen takımı(o_n, x_n, y_n, z_n) robot uç noktasına yerleştirilerek, atama işlemi tamamlanır. Şekil 2.1'de Denavit-Hartenberg kuralına göre eksen atamaların atandığı Staubli-RX160 manipülatörü gösterilmektedir.



Şekil 2.1: Staubli-RX160 manipülatörünün eksen takımları.

2.1.1.2 Eksen takımları arasındaki dönme ve ötelenmeler

o_i, x_i, y_i, z_i eksen takımının $o_{i-1}, x_{i-1}, y_{i-1}, z_{i-1}$ eksen takımına göre öteleme ve dönme hareketi A_i homojen dönüşüm matrisi ile tanımlanmaktadır. A_i matrisi, q_i eklem değişkenlerine bağlı olduğu için, robot hareketi ile değişkenlik göstermektedir.

$$A_i = A_i(q_i) \quad (2.1)$$

Staubli RX-160 manipülatörü üzerinde toplam 7 eksen takımı bulunduğu için toplam 6 tane homojen dönüşüm matrisi bulunmaktadır. Dolayısıyla robot uç noktasının

robot taban noktasına göre öteleme ve dönme hareketlerini veren dönüşüm matrisi denklem 2.2’de gösterilmektedir.

$$T_6^0 = A_1(q_1)A_2(q_2)A_3(q_3)A_4(q_4)A_5(q_5)A_6(q_6) \quad (2.2)$$

T_6^0 matrisi, o_6^0 ve R_6^0 olmak üzere 2’ye ayırabilmektedir. Burada o_6^0 , robot uç noktasının robot taban noktasına göre yapmış olduğu öteleme hareketini veren 3×1 vektördür. R_6^0 matrisi ise, uç noktanın robot taban noktasına göre yaptığı dönme hareketini veren 3×3 matrisidir.

$$T_6^0 = \begin{bmatrix} R_6^0 & o_6^0 \\ 0 & 1 \end{bmatrix} \quad (2.3)$$

Yukarıda yazılan denklemlerden faydalanılarak, robot üzerine yerleştirilmiş olan eksen takımları arasındaki öteleme vektörleri ve dönme matrisleri bulunmaktadır.

0. eksen takımı ile 1. eksen takımı arasındaki dönme matrisi ve öteleme vektörü($A_1(q_1)$)

$$R_1^0 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) \\ 0 & -1 & 0 \end{bmatrix} \quad (2.4)$$

$$Trans_1^0 = \begin{bmatrix} 0.15 \times \cos(\theta_1) - 0.126 \times \sin(\theta_1) \\ 0.126 \times \cos(\theta_1) + 0.15 \times \sin(\theta_1) \\ 0.176 \end{bmatrix} \quad (2.5)$$

1. eksen takımı ile 2. eksen takımı arasındaki dönme matrisi ve öteleme vektörü ($A_2(q_2)$)

$$R_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$Trans_2^1 = \begin{bmatrix} 0.825 \sin(\theta_2) \\ -0.825 \cos(\theta_2) \\ 0 \end{bmatrix} \quad (2.7)$$

2. eksen takımı ile 3. eksen takımı arasındaki dönme matrisi ve öteleme vektörü ($A_3(q_3)$)

$$R_3^2 = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) \\ \sin(\theta_3) & 0 & -\cos(\theta_3) \\ 0 & 1 & 0 \end{bmatrix} \quad (2.8)$$

$$\text{Trans}_3^2 = \begin{bmatrix} 0.105 \sin(\theta_3) \\ -0.105 \cos(\theta_3) \\ -0.126 \end{bmatrix} \quad (2.9)$$

3. eksen takımı ile 4. eksen takımı arasındaki dönme matrisi ve öteleme vektörü ($A_4(q_4)$)

$$R_4^3 = \begin{bmatrix} \cos(\theta_4) & 0 & -\sin(\theta_4) \\ \sin(\theta_4) & 0 & \cos(\theta_4) \\ 0 & -1 & 0 \end{bmatrix} \quad (2.10)$$

$$\text{Trans}_4^3 = \begin{bmatrix} 0 \\ 0 \\ 0.520 \end{bmatrix} \quad (2.11)$$

4. eksen takımı ile 5. eksen takımı arasındaki dönme matrisi ve öteleme vektörü ($A_5(q_5)$)

$$R_5^4 = \begin{bmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) \\ \sin(\theta_5) & 0 & -\cos(\theta_5) \\ 0 & 1 & 0 \end{bmatrix} \quad (2.12)$$

$$\text{Trans}_5^4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.13)$$

5. eksen takımı ile 6. eksen takımı arasındaki dönme matrisi ve öteleme vektörü ($A_6(q_6)$)

$$R_6^5 = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

$$\text{Trans}_6^5 = \begin{bmatrix} 0 \\ 0 \\ 0.110 \end{bmatrix} \quad (2.15)$$

Değerleri bulunan homojen matrislerin yardımı ile denklem 2.2’de yer alan 6 tane homojen matris yerine koyularak, robot uç noktasının taban noktasına göre olan toplam dönüşüm matrisi T_6^0 bulunmaktadır. Bu hesaplamalar Matlab ve C programları kullanılarak elde edilmektedir. İleri geometrik modelin doğruluğunu ölçebilmek için Matlab ortamında oluşturulmuş olan ileri geometrik modeli veren kodlar kullanılarak Matlab Simulink ile benzetim yapıp, oluşturulan ileri geometrik modelin doğruluğu ispat edilmektedir.

Robot ileri geometrik modeline ait benzetimler ve kodlar ekler bölümünde paylaşılmaktadır.

2.1.1.3 Euler açıları

Yörünge planlanırken bazı durumlarda robot uç noktasının dönme hareketini veren Euler açıları kullanılmaktadır. Sırası ile z, y, z eksenlerindeki α , β ve γ dönme açıları ile tanımlanan z-y-z Euler açıları, tez sırasında kullanılmaktadırlar. Denklem 2.16'da Euler açıları ile elde edilen 3×3 dönme matrisi verilmektedir.

$$R_B^A(\alpha, \beta, \gamma) = \begin{bmatrix} \cos(\alpha) \cos(\beta) \cos(\gamma) - \sin(\alpha) \sin(\gamma) & -\cos(\alpha) \cos(\beta) \sin(\gamma) - \sin(\alpha) \cos(\gamma) & \cos(\alpha) \sin(\beta) \\ \sin(\alpha) \cos(\beta) \cos(\gamma) + \cos(\alpha) \sin(\gamma) & -\sin(\alpha) \cos(\beta) \sin(\gamma) + \cos(\alpha) \cos(\gamma) & \sin(\alpha) \sin(\beta) \\ -\sin(\beta) \cos(\gamma) & \sin(\beta) \sin(\gamma) & \cos(\beta) \end{bmatrix}$$

$$R_B^A(\alpha, \beta, \gamma) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.16)$$

Eğer $\sin(\beta)$ sıfıra eşit değilse, α , β , γ açıları bir önceki denklemde oluşturulan dönme matrisinden faydalanılarak bulunmektedir.

$$\beta = \text{atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}) \quad (2.17)$$

$$\alpha = \text{atan2}\left(\frac{r_{23}}{\sin(\beta)}, \frac{r_{13}}{\sin(\beta)}\right) \quad (2.18)$$

$$\gamma = \text{atan2}\left(\frac{r_{32}}{\sin(\beta)}, -\frac{r_{31}}{\sin(\beta)}\right) \quad (2.19)$$

2.1.2 Ters geometrik model

İleri geometrik modelden farklı olarak, ters geometrik modelde, manipülatör uç noktasının taban noktasına göre tanımlı olan öteleme vektörü ve rotasyon matrisi kullanılarak eklem değişkenleri bulunmaktadır. T_n^0 matrisi, robot uç noktanın referans öteleme ve dönme hareketlerini temsil etmektedir. Ters geometrik modelin amacı, bu matrisi kullanarak $q_1, q_2, \dots, q_n$ eklem değişkenlerini bulmaktır.

$$T_n^0(q_1, \dots, q_n) = A_1(q_1) \dots \dots \dots A_n(q_n) \quad (2.20)$$

Ters geometrik modelin ileri geometrik modelden diğer bir farkı ise, her zaman tek çözümünün olmamasıdır. Çözüm ya tek, ya birden fazla ya da yoktur. Çözümün birden fazla olduğu durumlarda minimum eklem değişkenleri (Staubli Rx-160 manipülatörü için θ 'dir) seçilmelidir.

Staubli Rx-160 manipulatorünün ters geometrik modeli bulunurken geometrik yaklaşım ve kinematik ayrıştırma metotları kullanılmaktadır. Hesaplamalarda o_4 ile o_5 eksen takımı merkezlerinin aynı noktada seçilmesi büyük kolaylık sağlamaktadır. Bu merkez noktaları robotun bilek mekanizması olarak adlandırılan kesişim noktasına(o_C) yerleştirilmektedir.

Robot uç noktasının istenen dönme hareketine göre 3×3 rotasyon matrisi, Euler açıları kullanılarak bulunmaktadır.

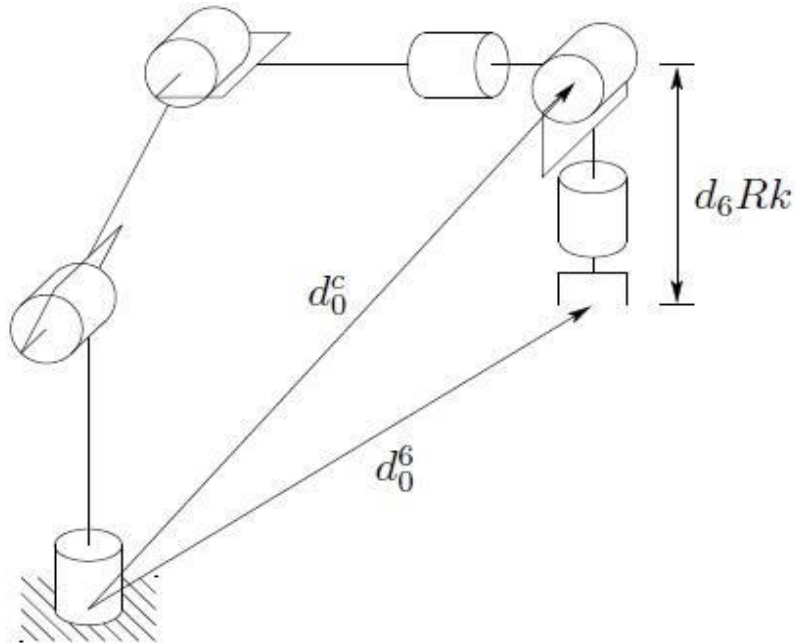
$$R = R_{Z,\alpha} R_{Y,\beta} R_{Z,\gamma} \quad (2.21)$$

Hesaplamalarda kinematik ayrıştırma metodunun tercih edilmesinin 2 sebebi vardır. Birinci sebep, kesişme noktası konumunun $o_C(x_C, y_C, z_C)$ geometrik yerinin, dönme matrisi ve uç noktanın koordinatları(x, y, z) kullanılarak hesaplanabilmesidir.

$$x_C = x - d_6 * R(1,3) \quad (2.22)$$

$$y_C = y - d_6 * R(2,3) \quad (2.23)$$

$$z_C = z - d_6 * R(3,3) \quad (2.24)$$



Şekil 2.2: Kinematik ayrıştırma metodu [5].

Kinematik ayrıştırma metodunun kullanılmasının ikinci sebebi ise, robot uç noktasının istenen dönme hareketini yapmasını sağlayan θ_4 , θ_5 ve θ_6 açılarının bu

yöntem ile bulunabilmesidir. Euler açıları ile oluşturulan dönme matrisi denklem 2.25’de gösterilen şekilde iki parçaya ayrılabilir.

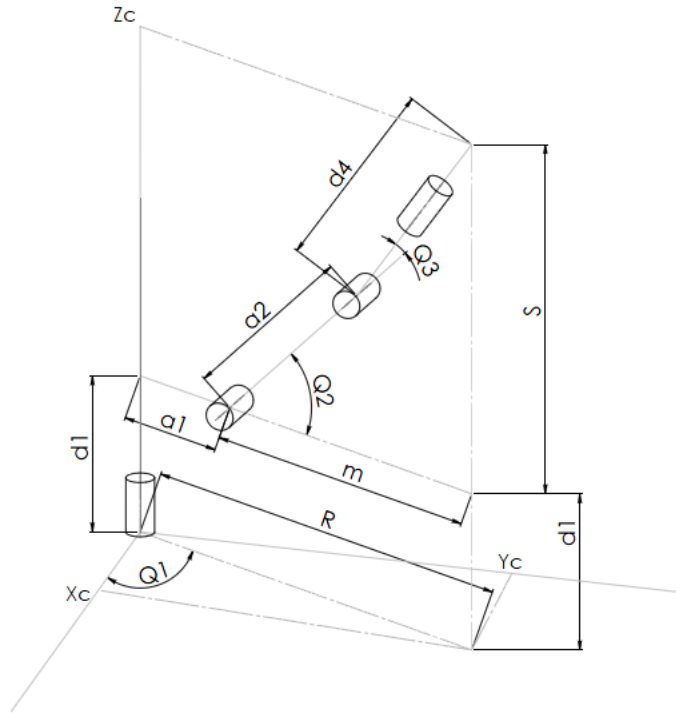
$$R = R_3^0 R_6^3 \quad (2.25)$$

Denklem düzenlenirse;

$$R_6^3 = (R_3^0)^T \cdot R \quad (2.26)$$

$(R_3^0)^T$ matrisini bulabilmek için θ_1 , θ_2 ve θ_3 açıları bulunmalıdır. Bu üç açıyı belirlemek için ise geometrik yaklaşım metodu kullanılmaktadır. Geometrik yaklaşım metodu adım adım şu şekilde yapılmaktadır;

- İlk önce robot uç noktasının yapması istendiği öteleme ve dönme hareketleri belirlenmektedir.
- Daha sonra kinematik ayrıştırma metodu kullanılarak kesişim noktasının geometrik yeri O_C bulunmaktadır.
- Kesişim noktasının konumu(x_C, y_C, z_C) bulunduktan sonra ise, uzaydaki bu noktayı(O_C) sağlatacak olan θ_1, θ_2 ve θ_3 eklem açıları bulunmaktadır.
- Son olarak ise rotasyon matrisi ikiye bölünüp, θ_4, θ_5 ve θ_6 açılara bağlı olan R_6^3 rotasyon matrisi bulunmaktadır.



Şekil 2.3: Geometrik yaklaşım metodu.

Şekil 2.3’de görüldüğü gibi referans o_C noktasını sağlayacak olan iki tane θ_1 açısı bulunmaktadır. Böylelikle θ_1 açısı iki farklı denklemden elde edilebilmektedir. Bulunan iki θ_1 açısından biri robotu sağ tarafa yönlendirirken diğeri sol tarafa yönlendirmektedir.

$$\theta_1 = \text{atan2}(y_C, x_C) \quad (2.27)$$

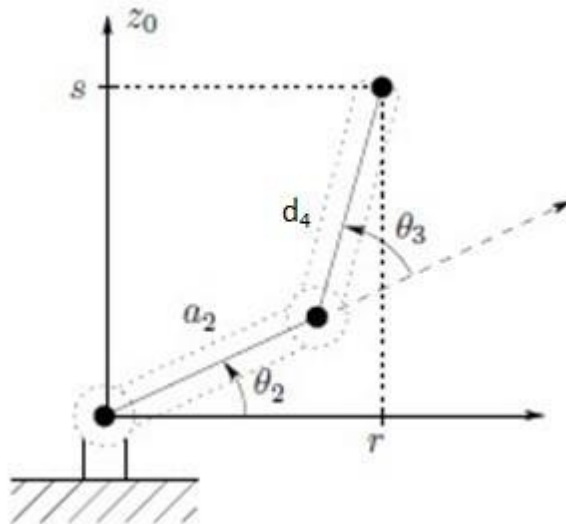
$$\theta_1 = \pm\pi + \text{atan2}(y_C, x_C) \quad (2.28)$$

Bulunan iki θ_1 açısından biri ters geometrik modeli oluşturmak için seçilmektedir. Bu seçim, Staubli RX-160 manipülatörünün eklemlerinin sağlayabileceği maksimum ve minimum açı sınırlarına göre seçilmelidir. Eklemlerin sağlayabileceği maksimum ve minimum açı sınırları çizelge 2.1’de verilmektedir.

Çizelge 2.1: Maksimum ve minimum açı değerleri [1].

θ_1 max	θ_2 max	θ_3 max	θ_4 max	θ_5 max	θ_6 max
$\pm 160^\circ$	$\pm 137.5^\circ$	$\pm 150^\circ$	$\pm 270^\circ$	$+120^\circ, -105^\circ$	$\pm 270^\circ$

İstenilen kesişim noktasını verecek olan θ_1 açısı belirlendikten sonra, sıra θ_2 ve θ_3 açılarını bulmaya gelmektedir. Aynı θ_1 açısında olduğu gibi bu iki açı içinde olası iki sonuç bulunmaktadır. Seçilen θ_2 ve θ_3 ikili açı grubundan birisi robotu aşağı dirsek adı verilen konuma götürürken, diğer açı grubu yukarı dirsek adı verilen konuma götürmektedir.



Şekil 2.4: θ_2 ve θ_3 açılarını belirlemek için kullanılan geometrik yaklaşım metodu.

$$s = z_c - d1 \quad (2.29)$$

$$r = \sqrt{x_c^2 + y_c^2} - a1 \quad (2.30)$$

$$\cos(\theta3) = \frac{r^2 + s^2 - a2^2 - d4^2}{2a2d4} = D \quad (2.31)$$

$$\theta3(1,2) = \text{atan2}(D, \pm\sqrt{1 - D^2}) \quad (2.32)$$

$$\theta2(1,2) = \frac{\pi}{2} - [\text{atan2}(s, r) - \text{atan2}(d4 \sin(\theta3(1,2))), a2 + d4 \cos(\theta3(1,2))] \quad (2.33)$$

Yukarıdaki denklemler sırası ile $\theta1$, $\theta2$ ve $\theta3$ açılarını bulmak için çözülmektedir. Fakat çözümler tek olmadığı için eklem sınırlarını aşmayan ve minimum açı değerlerini veren $\theta1$, $\theta2$ ve $\theta3$ üçlüsü bulunmaktadır. Bu işlem aşağıdaki prosedür izlenerek bulunmaktadır.

- İlk denklem sonucunda bulunan iki adet $\theta1$ açısından daha küçük olanı seçilmektedir. Seçilen bu açı değeri eklem sınırları içerisinde ise ters geometrik modelde kullanılmak üzere $\theta1$ açısı olarak seçilmektedir.
- Denklemden elde edilen $\cos(\theta3)$ değerinin 1 değerini aşıp aşmadığına bakılmaktadır. Eğer 1 değeri aşılmış ise seçilen $\theta1$ değerinin tekil çözüm olduğu anlaşılmaktadır. Bu yüzden bir önceki adımda seçilen $\theta1$ değeri yerine diğer $\theta1$ açı değeri seçilmektedir.
- Daha sonra bulunan iki tane $\theta2$ açısı değerinden küçük olan $\theta2$ açısı seçilerek, bu $\theta2$ değerini veren $\theta3$ açı değeri seçilmektedir.

$\theta1$, $\theta2$ ve $\theta3$ açı değerleri bulunduktan sonra, kinematik ayrıştırma metoduna geçilerek ilgili $\theta4$, $\theta5$ ve $\theta6$ açı değerleri bulunmaktadır.

Kinematik ayrıştırma metodunun denklemi aşağıda bir kez daha verilmektedir.

$$R_6^3 = (R_3^0)^T \cdot R \quad (2.34)$$

Yukarıdaki denklemde R_6^3 ve $(R_3^0)^T$ 3×3 matrisleri bir önceki adım olan ileri geometrik model kullanılarak oluşturulmaktadır. Denlem 2.35’de verilen R matrisi, Staubli RX-160 manipülatörünün uç noktasının taban eksen takımına göre dönme hareketini veren matristir.

$$R = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} \quad (2.35)$$

İleri geometrik modelden yararlanılarak çözülen denklem 2.34 sayesinde θ_4 , θ_5 ve θ_6 değerleri bulunmaktadır. Fakat θ_1 , θ_2 ve θ_3 açılarında olduğu gibi θ_4 , θ_5 ve θ_6 açı değerleri de tek değerli değildirler. Dolayısı ile minimum açıları seçebilmek için aşağıdaki prosedür uygulanmaktadır.

- Bir önceki prosedürde seçimi yapılmış olan θ_1 , θ_2 ve θ_3 açı değerlerini kullanarak iki olası θ_5 açısı değeri bulunmaktadır.
- Bulunan θ_5 değerlerinin 5.eklemin açı limitlerini geçip geçmediğine bakılmaktadır. Eğer bu değer geçilirse tekrar θ_1 , θ_2 ve θ_3 seçimine dönülerek diğer olası θ_1 , θ_2 ve θ_3 açı değerleri seçilmektedir.
- 5. Motor, 4.motordan daha büyük olduğu için minimum θ_4 değerini verecek θ_5 açı değeri seçilmektedir.
- θ_4 ve θ_5 açı değerleri bulunduktan sonra minimum değerli θ_6 açısı seçilmektedir.
- Son olarak ise, tekrardan bulunan altı açı değerinin robotun çalışma sınırları içinde olup olmadığına bakılmaktadır. Eğer içlerinden bir tanesi bile çalışma sınırları dışında ise Staubli RX-160 manipülatörünün uç noktası istenen konum ve rotasyon hareketini yapamamaktadır.

Yukarıda yazılan denklemler ve prosedürlere göre ters geometrik model hem Matlab ortamında hem de C programlama dilinde yazılmaktadır. Aynı ileri geometrik modelde yapıldığı gibi ters geometrik modelin benzetimi de Matlab Simulink ortamında yapıp, doğruluğu test edilmektedir. Ters geometrik modele giriş olarak, ileri geometrik modelin çıkışları verilerek, her iki modelinde sağlaması yapılmaktadır. Ters geometrik modele ait benzetim ekler kısmında paylaşılmaktadır.

2.2 İleri Kinematik Model

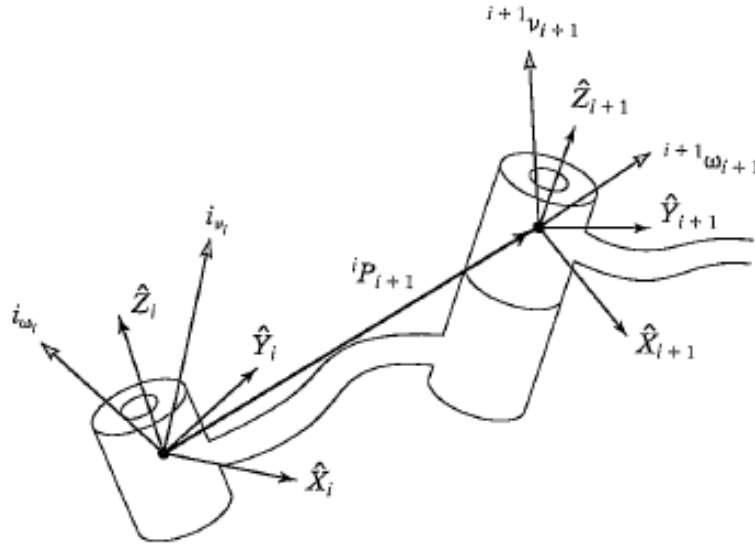
İleri kinematik model kullanılarak robot uç noktasının hızı ile eklem hızları arasındaki ilişki elde edilmektedir. Matematiksel olarak kinematik model, çalışma uzayı ile eklem uzayı arasındaki hızların fonksiyonunu veren modeldir. Bir cismin hızı tanımlanırken, açısal ve çizgisel olmak üzere ikiye ayrılmaktadır. Açısal hız ve çizgisel hızın tanımları denklem 2.36 ve 2.37’de verilmektedir.

$$\omega = \dot{\theta}k \quad (2.36)$$

$$\mathbf{v} = \boldsymbol{\omega} \times \mathbf{r} \quad (2.37)$$

Denklem 2.36’da yer alan $\mathbf{k}$ elemanı, dönme eksenini yönündeki birim vektörü temsil etmektedir. Denklem 2.37’de yer alan $\mathbf{r}$ elemanı ise, dönen katı bir cismin merkezinden çizgisel hızın bulunduğu noktaya kadar olan uzunluğu olarak tanımlanmaktadır.

Yukarıda verilen açısal ve çizgisel hız büyüklükleri, robot uzuvlarının hareketi incelerken de kullanılmaktadır. Şekil 2.5’de birbirine komşu olan iki uzvun hızları gösterilmektedir.



Şekil 2.5: Birbirine komşu olan iki uzvun hız vektörleri [6].

Eğer aynı eksen takımlarına göre tanımlanırlarsa, açısal hızlar toplanabilmektedir. Böylece $i+1$. uzvun açısal hızı, i . linkin açısal hızı artı $i+1$. eklemin sebep olduğu açısal hızın toplamına eşit olmaktadır. $i+1$. uzvun i . eksen takımına göre hızı denklem 2.38’de verilmektedir.

$$\boldsymbol{\omega}_{i+1}^i = \boldsymbol{\omega}_i^i + \mathbf{R}_{i+1}^i \dot{\theta}_{i+1} \mathbf{z}_{i+1}^i \quad (2.38)$$

$$\dot{\theta}_{i+1} \mathbf{z}_{i+1}^i = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{bmatrix} \quad (2.39)$$

Denklem 2.38’in sağ tarafı $\mathbf{R}_i^{i+1}$ dönme matrisi ile çarpılırsa, $i+1$. uzvun açısal hızı, kendi üzerine atanan eksen takımına göre bulunur.

$$\boldsymbol{\omega}_{i+1}^{i+1} = \mathbf{R}_i^{i+1} \boldsymbol{\omega}_i^i + \dot{\theta}_{i+1} \mathbf{z}_{i+1}^{i+1} \quad (2.40)$$

i+1. eksen takımına ait merkez noktasının çizgisel hızı, i. eksen takımına ait merkez noktasının çizgisel hızı artı i. uzvun sahip olduğu açısıl hızdan kaynaklanan hızdır.

$$v_{i+1}^i = v_i^i + \omega_i^i \times P_{i+1}^i \quad (2.41)$$

Aynı şekilde denklem 2.41 R_i^{i+1} dönme matrisi ile çarpılırsa, i+1. uzvun çizgisel hızını kendi üzerine atanmış olan eksen takımına göre bulunur.

$$v_{i+1}^{i+1} = R_i^{i+1} (v_i^i + \omega_i^i \times P_{i+1}^i) \quad (2.42)$$

Bu şekilde taban noktasından başlanarak, robot uç noktasına ait açısıl ve çizgisel hızlar bulunabilmektedir.

2.2.1 Jakobiye matris

Robotikte, eklem hızları ile robot uç noktası arasındaki hız ilişkisini bulmak için Jakobiye adı verilen özel bir matris kullanılmaktadır. Jakobiye matris, hem robot analizinde hem de robot kontrolünde geniş çaplı olarak kullanılmaktadır. Başlıca kullanıldığı alanlar; yörünge üretimi, robot dinamik modeli, tekil yapıların tespiti, robot kontrolü ve uç nokta ile eklem arasındaki kuvvet-moment bağıntısını veren denklemdir.

Staubli RX-160 manipülatörü toplam 6 tane dönel eklemden oluştuğu için oluşturulacak Jakobiye matris 6×6 matrisdir. Robotun eklem değişkenleri sırası ile $q_1, q_2, q_3, q_4, q_5, q_6$ olarak belirlenirse, taban noktasına ait eksen takımı ile robot uç noktası eksen takımı arasındaki dönüşüm matrisi şu şekilde yazılmaktadır.

$$T_6^0 = \begin{bmatrix} R_6^0 & o_6^0 \\ 0 & 0 \end{bmatrix} \quad (2.43)$$

Robot uç noktasının açısıl hızı ω_6^0 olarak ve çizgisel hızı da $v_6^0 = \dot{o}_6^0$ şeklinde tanımlanırsa, robot uç noktasına ait bu hız vektörleri ile eklem hızları arasındaki ilişki denklem 2.44 ve 2.45’de J_v ve J_ω 3×6 Jakobiye matrisler kullanarak tanımlanabilmektedir.

$$v_6^0 = J_v \dot{q} \quad (2.44)$$

$$\omega_6^0 = J_\omega \dot{q} \quad (2.45)$$

J_v ve J_ω 3×6 matrisleri birleştirilip 6×6 Jakobiye matris(J) şeklinde yazılırsa, denklem 2.44 ve 2.45 denklem 2.46 ve 2.47’de gösterilen şekle dönüşmektedirler.

$$\xi = J\dot{q} \quad (2.46)$$

$$\xi = \begin{bmatrix} v_6^0 \\ \omega_6^0 \end{bmatrix} \quad J = \begin{bmatrix} J_V \\ J_\omega \end{bmatrix} \quad (2.47)$$

Daha önceden de belirtildiği gibi aynı referans eksen takımında tanımlı olan açısal hızlar bir biri ile toplanabilmektedirler.+ Bu kuralı kullanarak oluşturulan denklem 2.48 ile Staubli RX-160 manipülatörüne ait uç noktanın açısal hızı bulunabilmektedir.

$$\omega_6^0 = \dot{q}_1 k + \dot{q}_2 R_1^0 k + \dots + \dot{q}_6 R_6^0 k \quad (2.48)$$

Denklem 2.48’de yer alan $R_{i-1}^0 k$ ifadesi yerine z_{i-1}^0 ifadesi kullanılırsa, denklem 2.48 denklem 2.49’a dönüşmektedir.

$$\omega_6^0 = \sum_{i=1}^6 \dot{q}_i z_{i-1}^0 \quad (2.49)$$

$$z_0^0 = k = (0,0,1)^T \quad (2.50)$$

Denklem 2.48 ve 2.49 kullanılarak 3×6 J_ω matrisi bulunmaktadır.

$$J_\omega = [z_0^0 z_1^0 \dots z_5^0] \quad (2.51)$$

Son olarak 3×6 J_V matrisi ise, zincir kuralı ve türev işlemi kullanılarak elde edilmektedir.

$$\dot{o}_6^0 = \sum_{i=1}^6 \frac{\partial o_6^0}{\partial q_i} \dot{q}_i \quad (2.52)$$

$$J_{Vi} = \frac{\partial o_6^0}{\partial q_i} \quad (2.53)$$

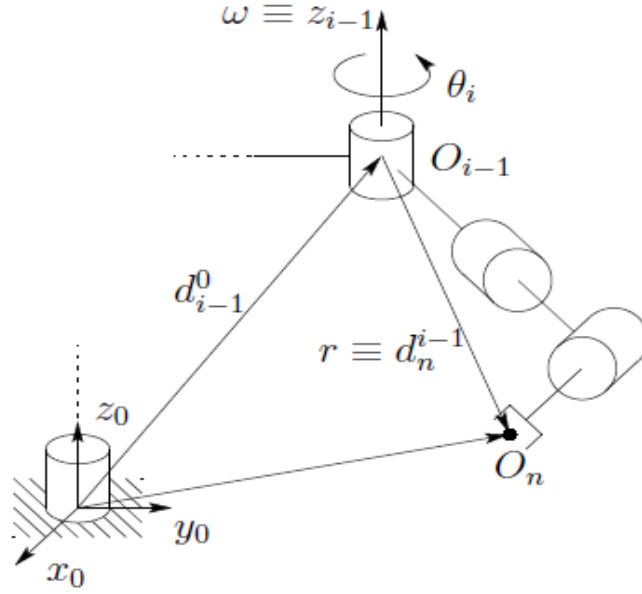
Staubli RX-160 manipülatörünün bütün eklemleri dönel eklem olduğu için $q_i = \theta_i$ yazılabilmektedir. Gerekli işlemler yapıldıktan sonra denklem 2.52, denklem 2.54’e dönüşmektedir.

$$\frac{\partial o_6^0}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} [R_i^0 o_6^i + R_{i-1}^0 o_{i-1}^{i-1}] = \dot{\theta}_i z_{i-1}^0 \times (o_6^0 - o_{i-1}^0) \quad (2.54)$$

Denklem 2.54 kullanılarak J_V matrisinin i. sütununun elemanı olan J_{Vi} bulunmaktadır.

$$J_{Vi} = z_{i-1} \times (o_6 - o_{i-1}) \quad (2.55)$$

Denklem 2.55’de yer alan elemanların üst indisleri 0 olduğu için yazılmamaktadır.



Şekil 2.6: i. uzvun, robot uç noktasının hareketine olan etkisi [5].

Tekrardan J_V ve J_ω matrisleri düzenlenerek yazılırsa, aşağıdaki denklem takımları elde edilmektedir.

$$J_V = [J_{V1} J_{V2} \dots J_{V6}] \quad (2.56)$$

$$J_\omega = [J_{\omega1} J_{\omega2} \dots J_{\omega6}] \quad (2.57)$$

$$J_{Vi} = z_{i-1} \times (o_6 - o_{i-1}) \quad J_{\omega i} = z_{i-1} \quad (2.58)$$

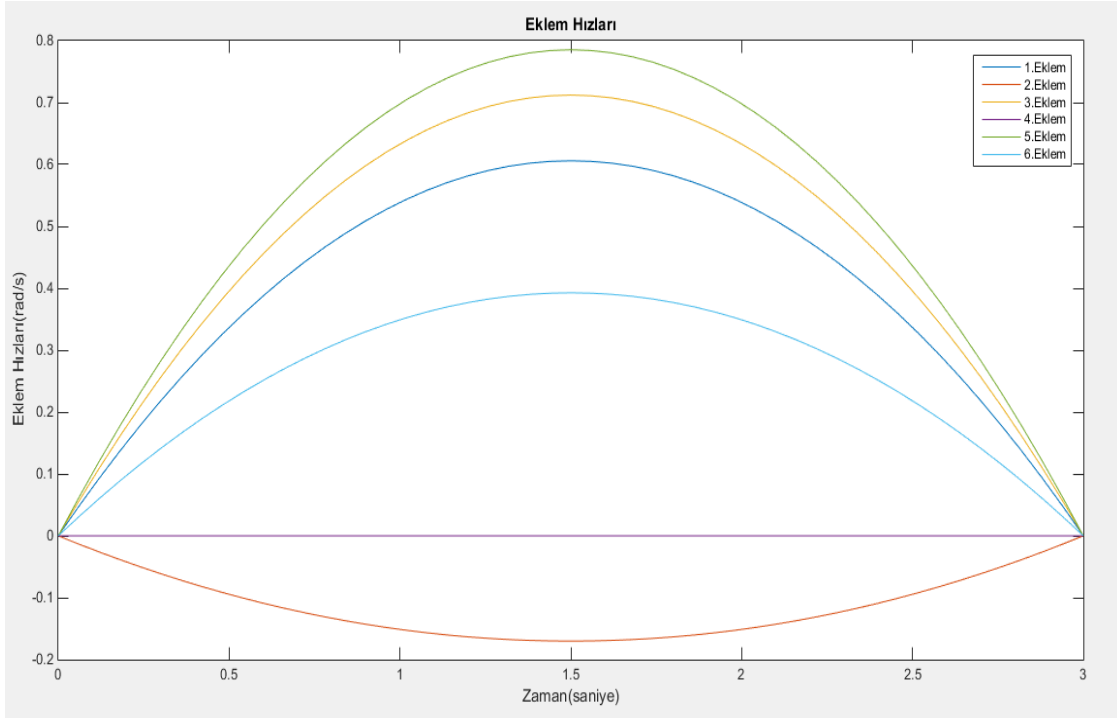
J_V ve J_ω matrisleri aynı matris içerisinde birleştirilerek, 6×6 Jakobiyen matris elde edilmektedir.

$$J_V = [J_1 J_2 \dots J_6] \quad (2.59)$$

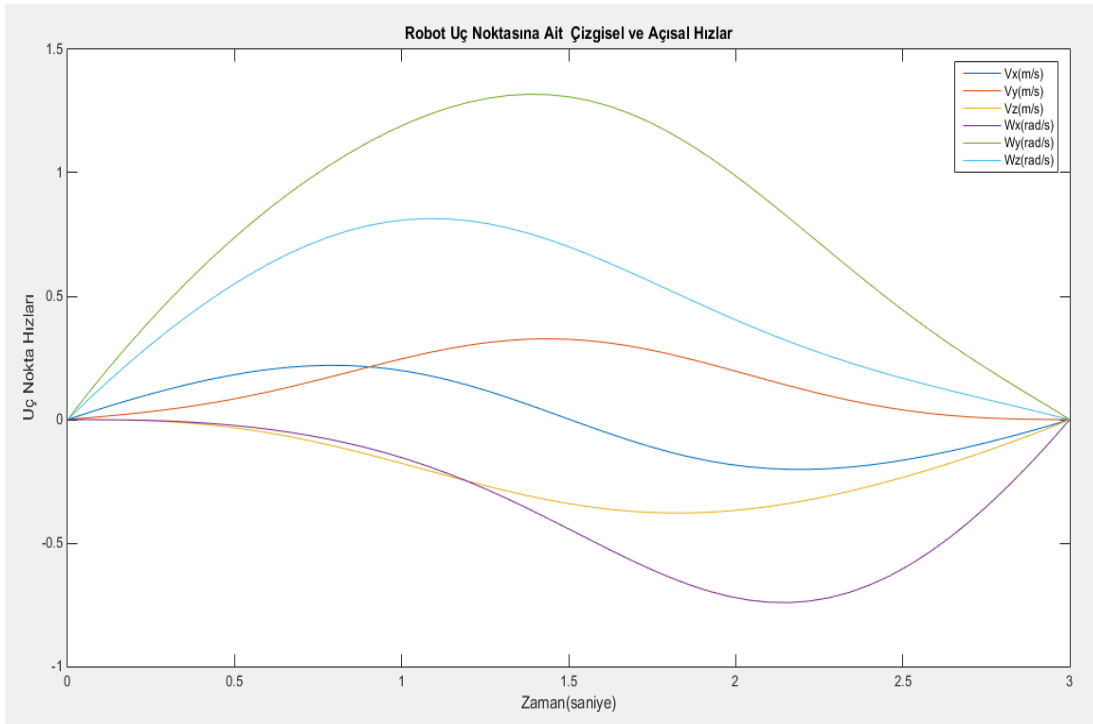
$$J_i = \begin{bmatrix} z_{i-1} \times (o_6 - o_{i-1}) \\ z_{i-1} \end{bmatrix} \quad (2.60)$$

Yukarıda verilen denklemler kullanılarak Staubli RX-160 manipülatörüne ait 6×6 Jakobiyen matris hem Matlab hem de C programlama dili kullanılarak bulunmaktadır. Ekler kısmında Matlab programı yardımı ile oluşturulan Jakobiyen matris paylaşılmaktadır. Ayrıca oluşturulan ileri kinematik modelin benzetimi Matlab Simulink programı ile yapılmaktadır. Benzetim yapılırken sonraki bölümde anlatılmakta olan üçüncü dereceden yörünge kullanılarak, Staubli RX-160 robotuna ait uç noktanın hızları bulunmaktadır. İleri kinematik modelin benzetimi sırasında

elde edilen sonuçlar şekil 2.7 ve 2.8’de görülmektedir. Ayrıca robot ileri kinematik modelinin Simulink ortamında yapılan benzetimine ait diyagramlar ekler kısmında paylaşılmaktadır.



Şekil 2.7: Eklem hızları.



Şekil 2.8: Robot uç noktasına ait çizgisel ve açısal hızlar.

2.3 Dinamik Model

Önceki bölümlerde açıklanan geometrik ve kinematik modellerde, robota etki eden kuvvet ve moment değerleri hesaba katılmadan, robot hareketi incelenmektedir. Bu bölümde açıklanmakta olan model ise, hareket ile hareketi oluşturan kuvvet ve momentleri beraber inceleyen matematiksel bir modeldir. Dinamik model, gerek robot tasarımında, gerek benzetiminde gerekse de kontrol algoritmalarında kullanılmakta olduğu için büyük bir öneme sahiptir. Robotikte, dinamik modeli elde etmek için kullanılan başlıca iki yöntem bulunmaktadır. Bunlardan birincisi, Euler-Lagrange denklemleri diğeri ise Newton-Euler algoritmasıdır. Staubli RX-160 manipülatörün dinamik modelini elde edilirken gerek hassaslığı, gerekse de model denklemlerini kapalı formda verebildiği için Euler-Lagrange denklemleri kullanılmaktadır. Fakat yapılan deneyler sırasında Euler-Lagrange denklemleri kullanılarak elde edilen dinamik modelin mevcut örnekleme zamanı olan 4 mili saniyede uygulanamaması nedeni ile deney sırasında Newton-Euler algoritması kullanılmaktadır.

2.3.1 Euler-Lagrange denklemleri

Euler-Lagrange denklemleri, belirli kısıtları olan mekanik sistemlerin zaman içerisindeki hareketini açıklayan denklemler topluluğudur. Euler-Lagrange denklemleri, m kütlesine sahip bir cimin y eksenı yönünde yerçekimi etkisi ile yapmış olduđu hareket ile açıklanabilmektedir. Cisme etki eden tek kısıt artı y yönündeki f kuvvetidir. Cismin hareket denklemi türetilirken Newton'un ikinci yasası kullanılmaktadır.

$$m\ddot{y} = f - mg \quad (2.61)$$

Denklem 2.61'in sol kısmı türev operatörü kullanılarak yeniden yazılabilmektedir.

$$m\ddot{y} = \frac{d}{dt}(m\dot{y}) = \frac{d}{dt} \frac{\partial}{\partial \dot{y}} \left(\frac{1}{2} m \dot{y}^2 \right) = \frac{d}{dt} \frac{\partial K}{\partial \dot{y}} \quad (2.62)$$

Denklem 2.62'deki K sembolü, cismin kinetik enerjisini temsil etmektedir. Benzer şekilde denklem 2.61'de yer alan mg ifadesi de türev operatörü kullanılarak yeniden yazılabilmektedir.

$$mg = \frac{\partial}{\partial y} (mgy) = \frac{\partial P}{\partial y} \quad (2.63)$$

Denklem 2.63’de yer alan P sembolü cismin potansiyel enerjisini temsil etmektedir. Bir alttaki denklemde kinetik enerji ile potansiyel enerji arasındaki fark L sembolü ile tanımlanmaktadır.

$$L = K - P = \frac{1}{2}m\dot{y}^2 - mgy \quad (2.64)$$

L ifadesinin, y değişkeni ve bu değişkenin birinci türevine göre kısmi türevleri denklem 2.65’de yazılmaktadır.

$$\frac{\partial L}{\partial \dot{y}} = \frac{\partial K}{\partial \dot{y}} \text{ ve } \frac{\partial L}{\partial y} = -\frac{\partial P}{\partial y} \quad (2.65)$$

Yukarıda yazılan denklemler kullanılıp denklem 2.61 yeniden yazılırsa cisme ait Euler-lagrange denklemi elde edilmektedir.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f \quad (2.66)$$

Denklem 2.66’da elde edilen Euler-Lagrange denklemi seri robotlarda kullanılabilecek şekilde yazılabilmektedir.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \quad i = 1, \dots, n \quad (2.67)$$

Yukarıda yazılan denklemde i sayısı eklem numarasını, τ ise tork değerini ifade etmektedir. Staubli RX-160 robotu 6 serbestlik dereceli olduğu için i sembolü 1’den başlayıp 6’ya kadar devam etmektedir.

Kinematik model kullanılarak, robot uzvu üzerinde yer alan herhangi bir noktanın hızı Jakobiyeen matris ve eklem hızları ile ifade edilebilmektedir.

$$v_i = J_{vi}(q)\dot{q} \quad \omega_i = J_{\omega i}(q)\dot{q} \quad (2.68)$$

i. uzvun kütlesi m_i ve kütle merkezine göre tanımlanan eylemsizlik matrisi I_i olarak tanımlanırsa, Staubli RX-160 manipülatörüne ait kinetik enerji ifadesi, denklem 2.69 ve 2.70’de yazılabilmektedir.

$$K = \frac{1}{2}\dot{q}^T \sum_{i=1}^6 [m_i J_{vi}(q)^T J_{vi}(q) + J_{\omega i}(q)^T R_{i(q)} I_i R_{i(q)}^T J_{\omega i}(q)] \dot{q} \quad (2.69)$$

$$K = \frac{1}{2}\dot{q}^T D(q) \dot{q} \quad (2.70)$$

Denklem 2.70’deki $D(q)$ elemanı eylemsizlik matrisi olarak tanımlanmaktadır. Eylemsizlik matrisi pozitif ve simetrik bir matrisdir. Staubli Rx-160 robotuna ait $D(q)$ matrisi 6×6 boyutundadır.

Staubli RX-160 manipulatörüne ait potansiyel enerjinin oluşmasını sağlayan fiziksel büyüklük yerçekimidir. Robota ait i. uzvun potansiyel enerjisi hesaplanırken, uzvun toplam kütesinin kütle merkezinde toplandığı farz edilmektedir.

$$P_i = g^T r_{ci} m_i \quad (2.71)$$

Denklem 2.71'deki g elemanı, sıfırıncı eksen takımı yönündeki yerçekimi ivmesini ifade etmektedir. r_{ci} elemanı ise, i. uzvun kütle merkezinin sıfırıncı eksen takımına göre mesafesini veren vektördür. 6 serbestlik dereceli Staubli RX-160 manipulatörünün toplam potansiyel enerjisi denklem 2.72'de verilmektedir.

$$P = \sum_{i=1}^6 g^T r_{ci} m_i \quad (2.72)$$

Staubli RX-160 manipulatörünün kinetik ve potansiyel enerji büyüklükleri bulunduğundan sonra, yukarıda elde edilen Euler-Lagrange denklemleri kullanılarak robota ait dinamik model oluşturulmaktadır.

$$K = \frac{1}{2} \sum_{i,j} d_{ij}(q) \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (2.73)$$

$$L = K - P = \frac{1}{2} \sum_{i,j} d_{ij}(q) \dot{q}_i \dot{q}_j - P(q) \quad (2.74)$$

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{kj} \dot{q}_j \quad (2.75)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_i d_{kj} \ddot{q}_j + \sum_j \frac{d}{dt} d_{kj} \dot{q}_j \quad (2.76)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{kj} \ddot{q}_j + \sum_{i,j} \frac{\partial d_{kj}}{\partial \dot{q}_i} \dot{q}_i \dot{q}_j \quad (2.77)$$

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial d_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial P}{\partial q_k} \quad (2.78)$$

Denklem 2.67'de verilen Euler-Lagrange denklemi, Staubli RX-160 manipulatörü için uygulanabilmektedir.

$$\sum_j d_{kj} \ddot{q}_j + \sum_{i,j} \left\{ \frac{\partial d_{kj}}{\partial \dot{q}_i} - \frac{1}{2} \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \dot{q}_j - \frac{\partial P}{\partial q_k} = \tau_k \quad (2.79)$$

2.79'da oluşturulan Euler-Lagrange denklemi matris formunda yazılabilmektedir.

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau \quad (2.80)$$

Denklem 2.80, Staubli RX-160 robotunun ters dinamik modelini göstermektedir. Denklemde kullanılan q , $\dot{q}$ ve $\ddot{q}$ ifadeleri sırası ile 6×1 eklem konumu, eklem hızı

ve eklem ivmesi vektörleridir. $D(q)$ ifadesi 6×6 eylemsizlik matrisidir. $D(q)$ matrisi sadece eklem konumlarına bağlı olduğu için robot hareket ettiğinde sürekli olarak elemanlarının büyüklüğü değişmektedir. Denklem 2.80'de yer alan ikinci ifade ise hem eklem konumlarına hem de eklem hızlarına bağlı olarak değişen 6×1 merkezkaç ve Coriolis vektörlerini ifade etmektedir. Denklemde tork değerini oluşturan üçüncü ve son terim ise yerçekimi etkisi ile oluşmakta olan torkları veren 6×1 vektördür. Aynı $D(q)$ matrisi gibi, $g(q)$ vektörü de yalnızca eklem konumlarına bağlı olarak değişmektedir. Bu üç terimin toplanması sonucunda, istenen hareketi yapması için robot eklemlerine gönderilmesi gereken tork değerlerini veren 6×1 τ vektörü elde edilmektedir. Robotun ileri dinamik modeli bulunmak istendiğinde, denklem 2.80'de yer alan $\ddot{q}$ ifadesi önce yalnız bırakılıp, daha sonra iki kez integrali alınmaktadır. Böylece robota uygulanan herhangi bir tork sonucu oluşan 6×1 eklem ivme, hız ve konum vektörleri bulunmaktadır.

$$\ddot{q} = D(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - g(q)) \quad (2.81)$$

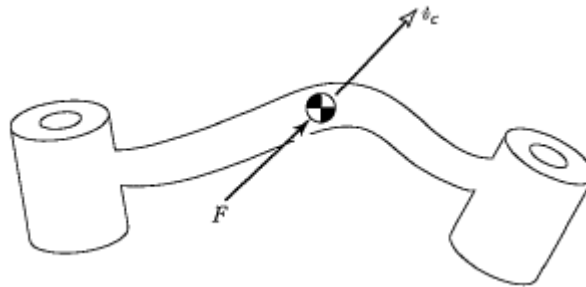
$$\dot{q} = \int \ddot{q} \quad (2.82)$$

$$q = \int \dot{q} \quad (2.83)$$

2.3.2 Newton-Euler denklemleri

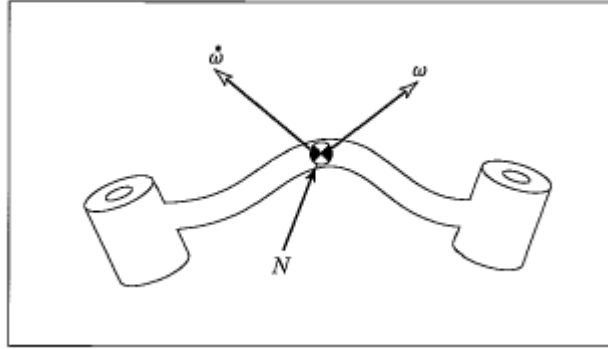
Hareketi oluşturan kuvvet ve momentler referans ivmeye bağlı olarak tanımlanabilmektedirler. Newton-Euler yöntemi elde edilen dinamik model aslında iki temel denklem kullanılarak elde edilmektedir. Bunlardan ilki, kuvveti çizgisel ivme ve kütle ile ilişkilendiren Newton denklemi, diğeri ise momenti açısal ivme ve moment ile ilişkilendiren Euler denklemidir.

$$F = m\ddot{c} \quad (2.84)$$



Şekil 2.9: Newton denklemi ile elde edilen hareket.

$$N = I\dot{w} + w \times Iw \quad (2.85)$$



Şekil 2.10: Euler denklemi ile elde edilen hareket.

Eğer yörünge planlaması sonucu elde edilen konum, hız ve ivme değerleri bilinip, robota ait parametre bilgileri olan kütle ve atalet matrisi bilgileri de mevcut ise ileri ve geri Newton-Euler denklemlerine ait iterasyonlar uygulanarak robota ait dinamik model elde edilebilmektedir. İleri denklemlere ait iterasyon uygulanarak hız ve ivmeler elde edilmektedir. İleri denklemler 0. eksen takımından başlanarak 5. eksen takımına kadar uygulanmaktadır. İleri denklemler sırası ile aşağıda gösterilmektedir.

$${}^{i+1}_w = {}^{i+1}_i R {}^i_w + \dot{\theta}_{i+1} {}^{i+1}_{i+1} z \quad (2.86)$$

$${}^{i+1}_w = {}^{i+1}_i R {}^i_w + {}^{i+1}_i R {}^i_w \times \dot{\theta}_{i+1} {}^{i+1}_{i+1} z + \ddot{\theta}_{i+1} {}^{i+1}_{i+1} z \quad (2.87)$$

$${}^{i+1}_v = {}^{i+1}_i R ({}^i_w \times {}^{i+1}_i p + {}^i_w \times ({}^i_w \times {}^{i+1}_i p)) + {}^{i+1}_v \quad (2.88)$$

$${}^{i+1}_v = {}^{i+1}_w \times {}^{i+1}_i p + {}^{i+1}_w \times ({}^{i+1}_w \times {}^{i+1}_i p) + {}^{i+1}_v \quad (2.89)$$

$${}^{i+1}_F = m_{i+1} {}^{i+1}_v \quad (2.90)$$

$${}^{i+1}_N = {}^{i+1}_i I {}^{i+1}_w + {}^{i+1}_w \times {}^{i+1}_i I {}^{i+1}_w \quad (2.91)$$

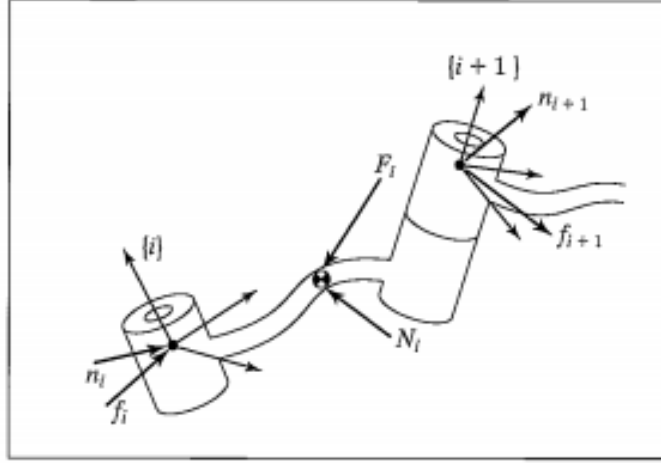
Sırası ile 0. eksen takımından 5. eksen takımına kadar ileri Newton-Euler denklemleri uygulandıktan sonra geri Newton-Euler denklemleri uygulanarak eklemlere ait kuvvet ve momentler elde edilmektedir. Aşağıda gösterilen geri Newton-Euler denklemleri sırası ile 6. eksen takımından 1. eksen takımına doğru uygulanmaktadır.

$${}^i_f = {}^i_{i+1} R {}^{i+1}_f + {}^i_F \quad (2.92)$$

$${}^i_n = {}^i_N + {}^i_{i+1} R {}^{i+1}_n + {}^i_p \times {}^i_f + {}^i_p \times {}^i_{i+1} R {}^{i+1}_f \quad (2.93)$$

$$\tau_i = {}^i_n^T {}^i_z \quad (2.94)$$

Yer çekimi ivmesine bağlı olarak robot eklemlerinde oluşan torku bulabilmek için ileri Newton-Euler denklemlerine başlamadan 0. Eksen takımına ait hız yer çekimi ivmesine eşit olarak yazılmaktadır.



Şekil 2.11: Newton-Euler denklemleri ile dinamik modeli elde edilen uzuv.

2.3.3 Sistem tanıma

Euler-Lagrange denklemleri kullanılarak, analitik olarak robota ait dinamik model elde edilmektedir. Fakat her makine gibi endüstriyel manipulatörler de hareketleri sırasında sürtünme kuvvetlerine maruz kalmaktadırlar. Bu yüzden her ne kadar analitik olarak dinamik model hassas ve doğru bir şekilde bulunsun da sürtünmenin varlığı nedeniyle oluşturulan dinamik model yetersiz kalmaktadır. Bu yüzden Staubli RX-160 manipulatörüne ait sürtünme modeli, denklem 2.80'in sol tarafına sürtünme torkularını verecek şekilde eklenmektedir. Staubli RX-160 manipulatörünün 2. Uzvunda, 2. Ekleme gelen yüklerin bir kısmını taşıması için kullanılan bir yay mevcuttur. Dolayısıyla oluşturulan sürtünme modelinin yanında yay modelinin de dinamik modele eklenmesi gerekmektedir. Staubli RX-160 robotuna ait gerekli olan sürtünme ve yay modelleri için daha önce Mekatronik Eğitim ve Araştırma Merkezinde yapılan tez çalışmasından yararlanılmaktadır(Egemen Zengin,2013). Yapılan çalışma ile robota ait sürtünme ve yay modelleri elde edilen sonuçlarla doğrulanmıştır.

2.3.3.1 Sürtünme modeli

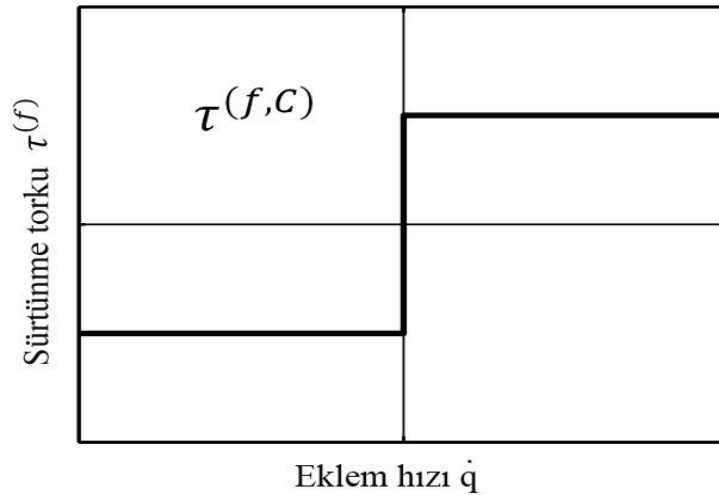
Robota etki eden sürtünme kuvvetleri kayma rejiminde oluşan sürtünme kuvvetleri ve kayma öncesi rejiminde oluşan sürtünme kuvvetleri olarak ikiye ayrılmaktadır. Kayma rejimi sırasında, sürtünme kuvvetleri yüzeyler arasındaki hızlar sıfırdan farklı

iken oluşmaktadır. Kayma öncesi rejimde ise sürtünme kuvvetleri yüzeyler arasındaki hızlar sıfır ve sıfıra yakinken oluşmaktadır.

Robotlar için geliştirilen nihai sürtünme modeli, birkaç alt sürtünme modelinin toplamı şeklinde olmaktadır. Robotun maruz kaldığı sürtünme etkisi modellenirken ilk kullanılan model Coulomb sürtünme modelidir. Coulomb sürtünme modeli denklem 2.95 ile verilmektedir.

$$\tau^{(f)} = \text{sign}(\dot{q})\tau^{(f,c)} \quad (2.95)$$

Denklem 2.95’de yer alan $\tau^{(f)}$ sürtünme torkunu, $\dot{q}$ eklem hızını ve $\tau^{(f,c)}$ ise Coulomb sürtünme torkunu temsil etmektedir. Coulomb sürtünme modeli, kuru yüzeylerin birbiri üzerinde kaymasıyla oluşan büyük sürtünme kuvvetlerini vermektedir. Denklem 2.95’de yer alan $\text{sign}(\dot{q})$ ifadesi, Coulomb sürtünme modelinin hızın sıfır olduğu yerleri açıklayamadığını göstermektedir.

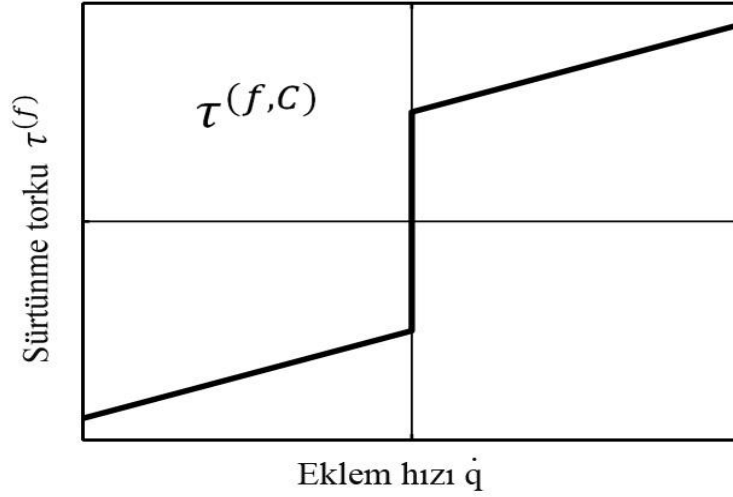


Şekil 2.12: Coulomb sürtünme modeli [2].

Hesaba katılan ikinci sürtünme modeli viskoz sürtünme modelidir. Bu model ile yağlanmış yüzeyler arasındaki viskozite etkisi açıklanmaktadır. Bir önceki denklem ile verilmekte olan sürtünme torku, denklem 2.96’e dönüşmektedir.

$$\tau^{(f)} = \text{sign}(\dot{q})\tau^{(f,c)} + c^{(v)}\dot{q} \quad (2.96)$$

Denklem 2.96’da görülmekte olan $c^{(v)}$, viskoz sürtünme parametresidir. Denklemden de görüldüğü gibi viskoz sürtünme modeli eklem hızının doğrusal fonksiyonudur. Viskoz sürtünme modelinin, Coulomb sürtünme modeline etkisi şekil 2.13’de verilmektedir.

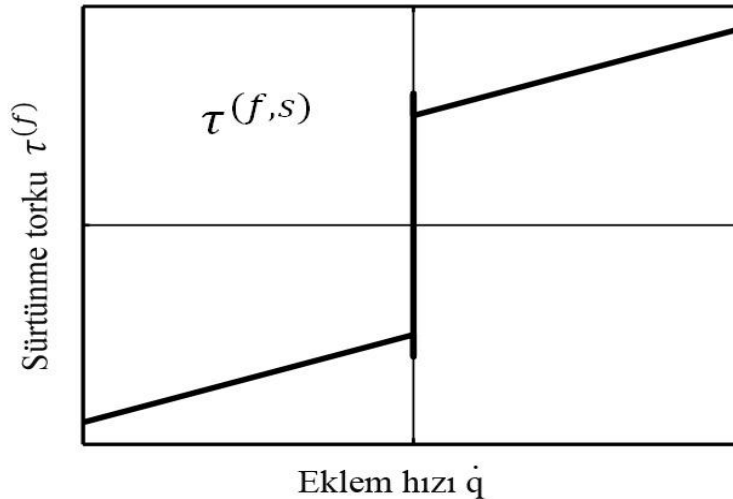


Şekil 2.13: Coulomb ve viskoz sürtünme modeli [2].

Bir diğer sürtünme modeli ise statik sürtünme modelidir. Bu model, yüksek sürtünme kuvvetlerinin yüzeyler arasında kayma hareketlerine neden olduğunu göstermektedir. Statik sürtünme modelinin, sürtünme modeline etkisini veren denklem 2.97 ile verilmektedir.

$$\tau^{(f)} = \begin{cases} |\tau^{(f)}| \leq \tau^{(f,s)}; & \dot{q} = 0 \text{ ise} \\ \text{sign}(\dot{q})\tau^{(f,c)} + c^{(v)}\dot{q}; & \dot{q} \neq 0 \text{ ise} \end{cases} \quad (2.97)$$

Statik sürtünme modelinin, sürtünme modeline yaptığı etkinin şekli ise şekil 2.14’de verilmektedir.



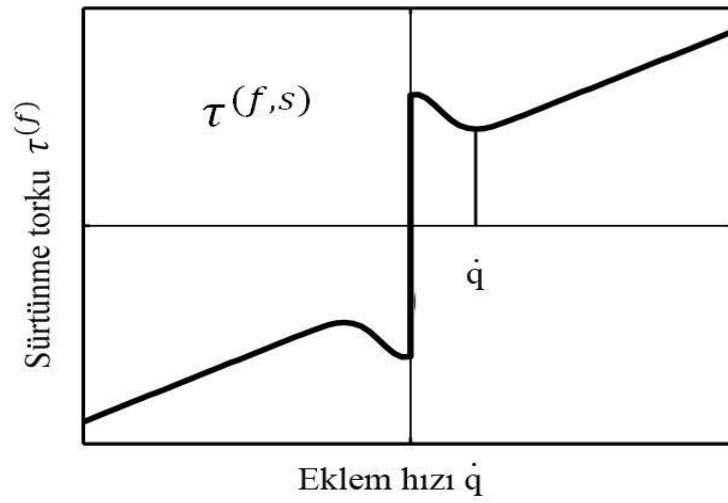
Şekil 2.14: Coulomb, viskoz ve statik sürtünme modeli [2].

Statik sürtünme modelinin eklenmesiyle, oluşan sürtünme modeli sıfırdan farklı durumlara geçiş sırasında süreksizlik göstermektedir. Stribeck(1902), yağlanmış yüzeylerde statik sürtünme modelinden Coulomb sürtünme modeline geçişte

fonksiyonun süreksiz olmadığını, eklem hızına bağlı sürekli bir fonksiyon olduğunu göstermiştir. Bo ve Pavelescu(1982), Stribeck etkisini veren modeli geliştirerek, statik sürtünmeden Coulomb sürtünme modeline geçişi üstel şekilde ifade eden denklemi oluşturmuşlardır. Bu denklem, denklem 2.98 olarak verilmektedir.

$$\tau^{(f)} = \text{sign}(\dot{q})[\tau^{(f,c)} + (\tau^{(f,s)} - \tau^{(f,c)})e^{-|\dot{q}/\dot{q}^{(s)}|^{\delta(s)}}] \quad (2.98)$$

Denklem 2.98’de yer alan $\dot{q}^{(s)}$, Stribeck hızı olarak tanımlanmakta ve Stribeck etkisinin olduğu hız aralığını göstermektedir. Şekil 2.15’de Stribeck etkisinin sürtünme modeline etkisi gösterilmektedir.



Şekil 2.15: Sürtünme modelinde Stribeck etkisi [2].

Amstrong-Helovury(1991), Stribeck modeline viskoz terimi olan $c^{(v)}\dot{q}$ ifadesini dahil etmiştir.

$$\tau^{(f)} = \text{sign}(\dot{q})[\tau^{(f,c)} + (\tau^{(f,s)} - \tau^{(f,c)})e^{-|\dot{q}/\dot{q}^{(s)}|^{\delta(s)}}] + c^{(v)}\dot{q} \quad (2.99)$$

Bu sürtünme modeli, manipülatör sürtünme modeli olarak sık sık kullanılsa da çok fazla parametreye ihtiyaç duyduğu için sürtünme modeli elde edilecek olan elemanların sürtünme özellikleri dikkate alınarak, bu modelin sadece en önemli özellikleri dikkate alınmaktadır.

Staubli RX-160 manipülatörünün sürtünme modeli, yukarıdaki modeller referans alınarak oluşturulmaktadır. Robot için sürtünme modeli oluşturulurken eklem mekanizmaları göz önünde bulundurulmaktadır. Tezin giriş bölümünde değinildiği gibi, ilk dört eklem helisel dişli, düz dişli ve rulmanlardan oluşmaktadır. Son iki eklem ise sonsuz dişli ve rulmanlardan oluşmaktadır. Ayrıca Staubli RX-160

manipülâtörünün son eklemi olan 6.eklemde konik dişli de bulunmaktadır. Eklemleri oluşturan elemanlar göz önünde bulundurularak, sürtünme modelinde rulmanların pürüzlülük etkisi ve helisel dişli çiftinin viskoz etkisi kullanılarak denklem 2.100 oluşturulmaktadır.

$$\tau_j^{(f)} = \tau_j^{(a,BL)} e^{-(\dot{q}_j/\dot{q}_j^{(s)})\delta_j^{(a)}} + c_j^{(v)} \dot{q}_j^{(1-\delta_j^{(v)})} \quad (2.100)$$

Staubli RX-160 manipülâtörünün beşinci motoruyla altıncı eklemi arasında kinematik bir etkileşim bulunmaktadır. Bu ilişki şekil 2.16’da görülmektedir.

5.servo motorun açısal hızı doğrudan $\dot{q}_5$ ile orantılı iken, 6.servo motorun açısal hızı $\dot{q}_5$ ile $\dot{q}_6$ farkına bağlıdır. Bu nedenle ilk dört eklem için kullanılan sürtünme modelinden farklı bir sürtünme modeline ihtiyaç duyulmaktadır. İhtiyacı karşılayacak olan bu modelde kullanılmak üzere yeni bir hız tanımlanmaktadır.

$$\dot{q}_7 = \dot{q}_5 + \dot{q}_6 \quad (2.101)$$

Beşinci ve altıncı eklemlerin sürtünme modelleri, iki farklı alt denklem kullanılarak oluşturulmaktadır.

$$\tau_5^{(f)} = \tau_a^{(f)} + \tau_c^{(f)} \quad (2.102)$$

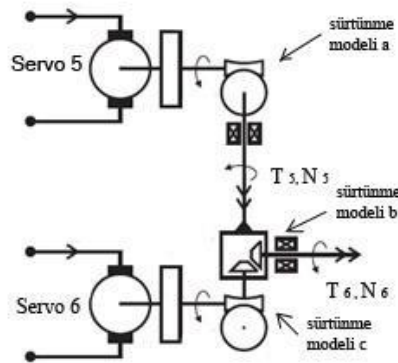
$$\tau_6^{(f)} = \tau_b^{(f)} + \tau_c^{(f)} \quad (2.103)$$

Yukarıda kullanılan $\tau_a^{(f)}$, $\tau_b^{(f)}$ ve $\tau_c^{(f)}$ sürtünme modelleri aşağıdaki denklemler kullanılarak tanımlanmaktadır.

$$\tau_a^{(f)} = \tau_5^{(a,BL)} e^{-(\dot{q}_5/\dot{q}_5^{(s)})\delta_5^{(a)}} + c_5^{(v)} \dot{q}_5^{(1-\delta_5^{(v)})} \quad (2.104)$$

$$\tau_b^{(f)} = \tau_6^{(a,BL)} e^{-(\dot{q}_6/\dot{q}_6^{(s)})\delta_6^{(a)}} + \tau_6^{(v,l)} (1 - e^{-\dot{q}_6/\dot{q}_6^{(l)}}) \quad (2.105)$$

$$\tau_c^{(f)} = \tau_7^{(a,BL)} e^{-(\dot{q}_7/\dot{q}_7^{(s)})\delta_7^{(a)}} + c_7^{(v)} \dot{q}_7^{(1-\delta_7^{(v)})} \quad (2.106)$$



Şekil 2.16: 5. ve 6.eklem arasındaki kinematik etkileşim [2].

Yukarıdaki denklemleri kullanarak sürtünme modelini oluşturabilmek için sürtünme parametrelerinin bulunması gerekmektedir. Parametrelerin tanınması işlemi daha önce İTÜ Mekatronik Eğitim ve Araştırma Merkezinde bulunan Staubli RX-160 manipülatörü için yapılmıştır(Egemen Zengin,2013). Geçmişte yapılan bu çalışma sırasında elde edilen parametre değerleri sürtünme modeli oluşturulurken doğrudan kullanılmaktadır. Çizelge 2.2’de Staubli RX-160 manipülatörü için bulunmuş sürtünme parametreleri paylaşılmaktadır.

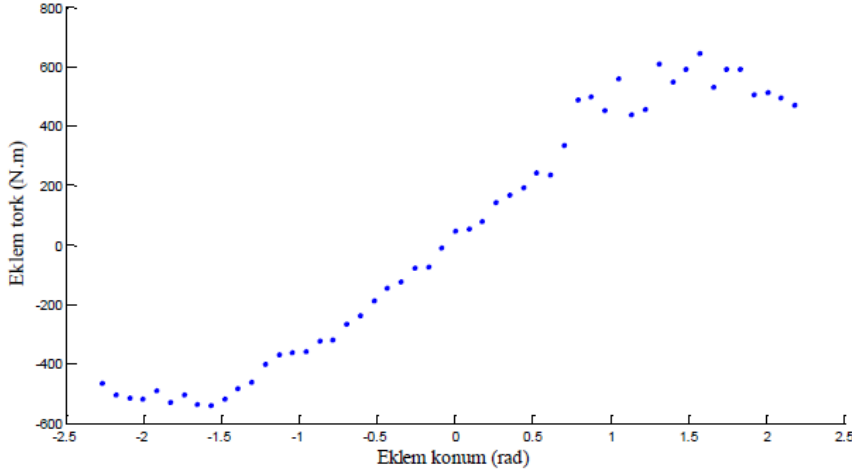
Çizelge 2.2: Staubli RX-160 manipülatörünün sürtünme parametreleri [2].

Model	1	2	3	4	a	b	c
$\tau^{(a,BL)}$	55.3234	30.8468	40.4853	168.4421	2.6767	-0.7271	1.0903
$\dot{q}^{(s)}$	0.0710	0.0400	0.0610	0.0200	0.2500	0.3500	0.0800
$\delta^{(a)}$	0.7000	0.8000	0.6000	0.5500	0.7000	0.5000	0.5000
$c^{(v)}$	191.9090	107.0665	59.4372	24.1345	20.0239	—	8.3559
$\delta^{(v)}$	0.3837	-0.1724	0.3739	0.3562	0.3494	—	0.3907
$\tau^{(v,l)}$	—	—	—	—	—	-1.3742	—
$\dot{q}^{(l)}$	—	—	—	—	—	4.6408	—

2.3.3.2 Yay modeli

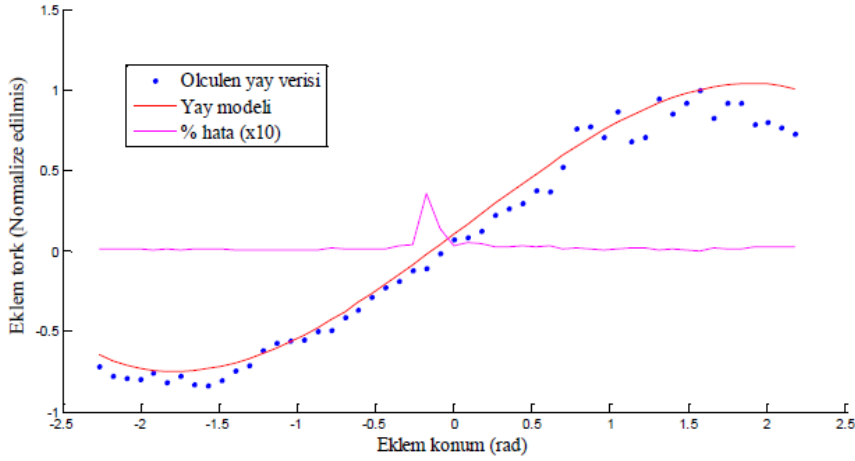
Staubli RX-160 manipülatörünün 2.uzvunda bulunan bir yay mevcuttur. Bu yay 2.eklemde sonraki mekanizmanın ağırlığını taşımak için kullanılmaktadır. Bu yayın manipülatörün 2.ekleminin açısal yer değiştirmesine göre oluşturduğu tork değeri şekil 2.17’de verilmektedir. Deneysel olarak elde edilen veriler ve en düşük kareler yöntemi kullanılarak, yay modeli oluşturulmaktadır. Oluşturulan yay modeli, denklem 2.107 ile verilmektedir.

$$\tau_s(t) = -46.56q(t)^3 + 8.07q(t)^2 + 471.09q(t) + 66 \quad (2.107)$$



Şekil 2.17: Yayın 2.eklemin hareketine bağlı oluşturduğu tork değerleri [2].

Deneyssel olarak bulunan tork değerleri ile elde edilen yay modeli şekil 2.18'de karşılaştırılmaktadır.



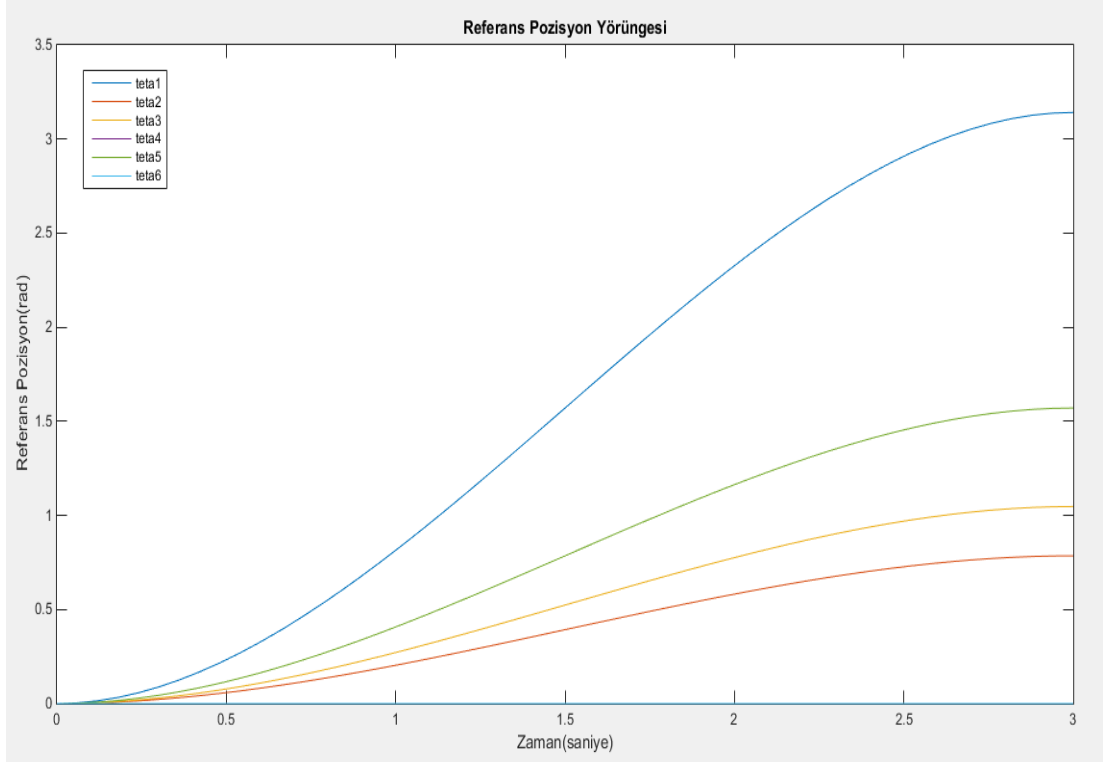
Şekil 2.18: Yay modeli ile gerçek değerlerin karşılaştırılması [2].

Denklem 2.80, analitik olarak robota ait dinamik modeli göstermektedir. Fakat bu denklemde sürtünme ve yay modeli hesaba katılmadığı için, tam olarak robot dinamiğini açıklanamamaktadır. Bu sebeple analitik olarak oluşturulan dinamik modele sistem tanıma sonucunda bulunan sürtünme ve yay modelleri eklenmektedir. Böylelikle denklem 2.80, denklem 2.108'e dönüşmektedir.

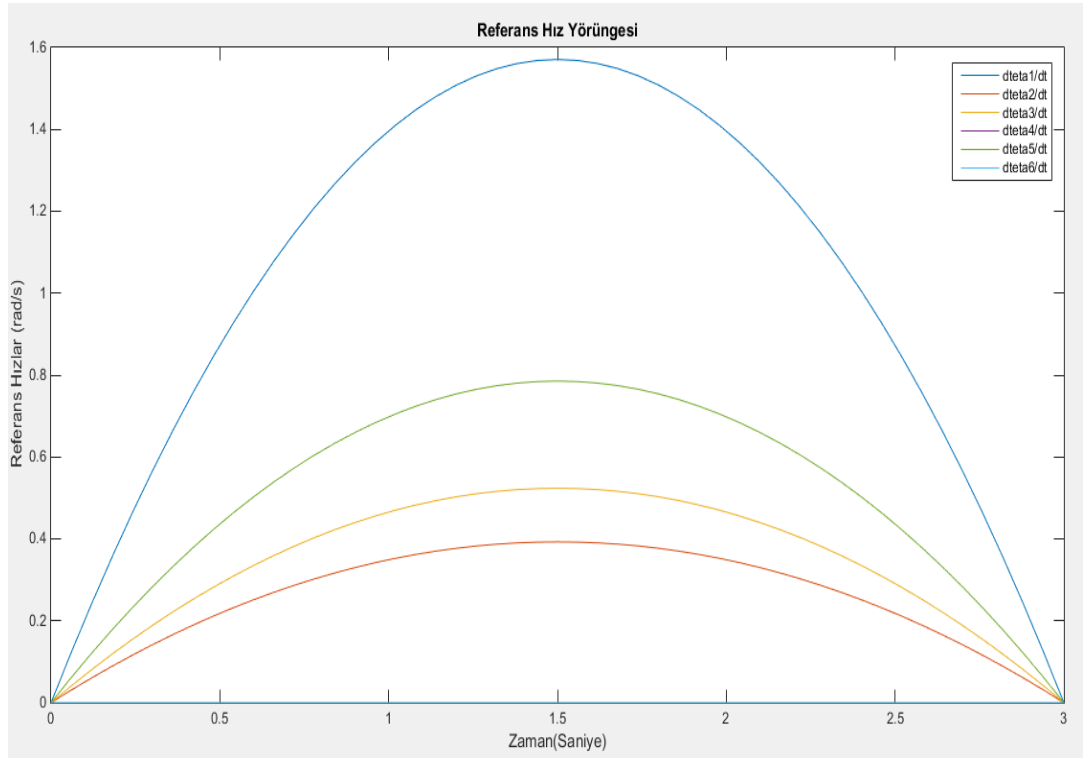
$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_s(q) + \tau_f(q, \dot{q}) = \tau \quad (2.108)$$

Denklem 2.97'de oluşturulan dinamik model kullanılarak matlab simulink ortamında robotun ters dinamik modeline ait benzetim yapılmaktadır. Konum, hız ve ivme referans yörüngeleri için bir sonraki bölümde anlatılacak olan üçüncü dereceden polinom kullanılarak, eklemlerde oluşan tork değerleri bulunmaktadır. Robot

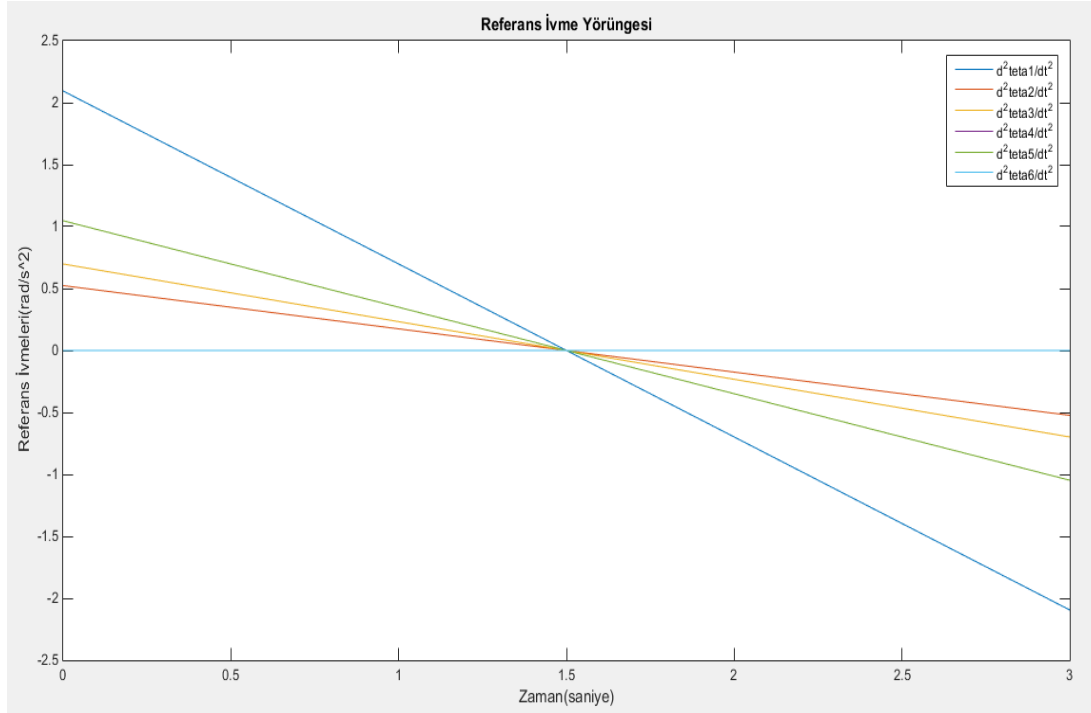
eklemelerine referans olarak gönderilen konum değerleri sırası ile π , $\pi/4$, $\pi/3$, 0, $\pi/2$ ve 0 radyandır. Dinamik modeli hesaplamak için kullanılan referans konum, hız ve ivme yörüngeleri şekil 2.19, şekil 2.20 ve şekil 2.21’de gösterilmektedir.



Şekil 2.19: Referans konum yörüngesi.

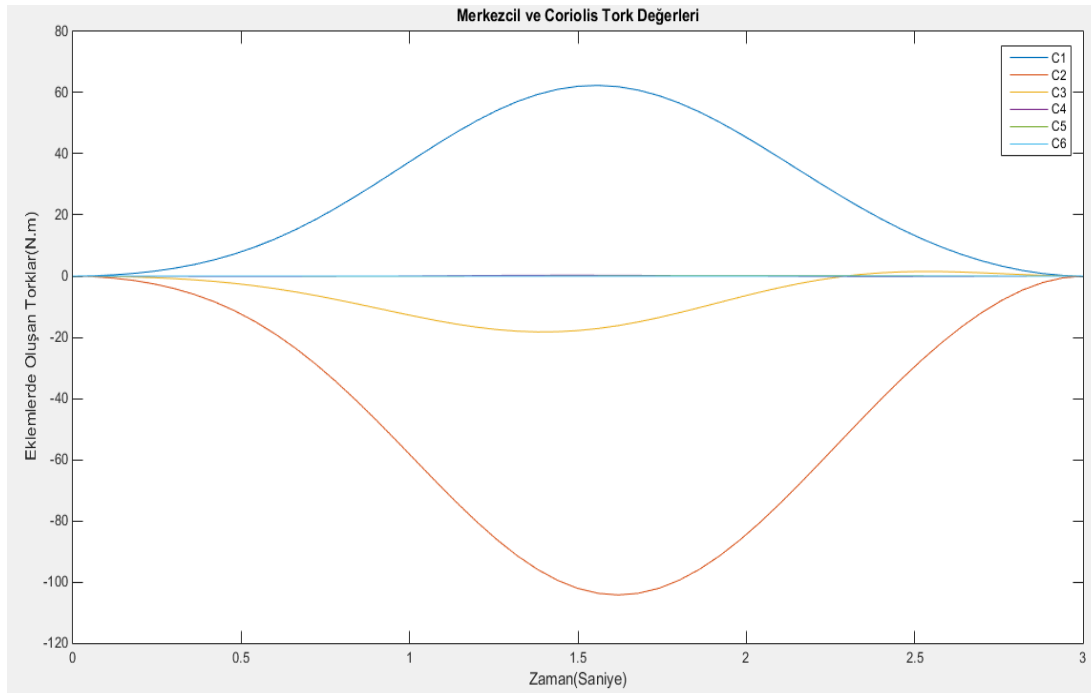


Şekil 2.20: Referans hız yörüngesi.

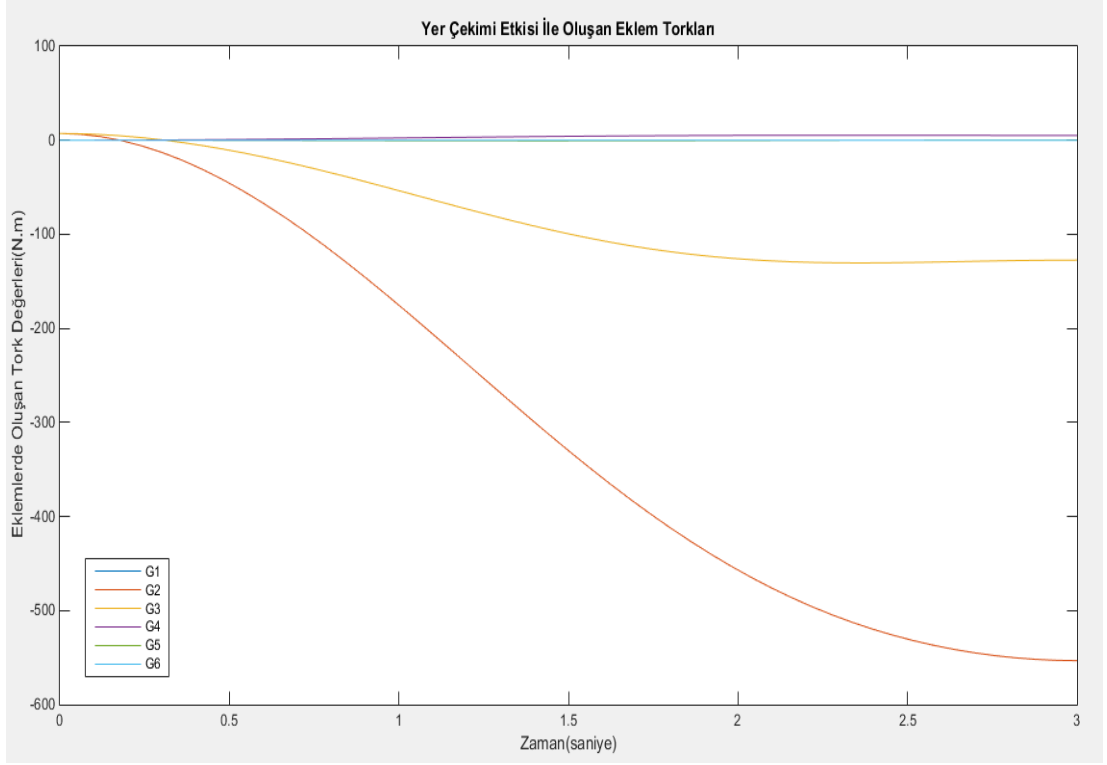


Şekil 2.21: Referans ivme yörüngesi.

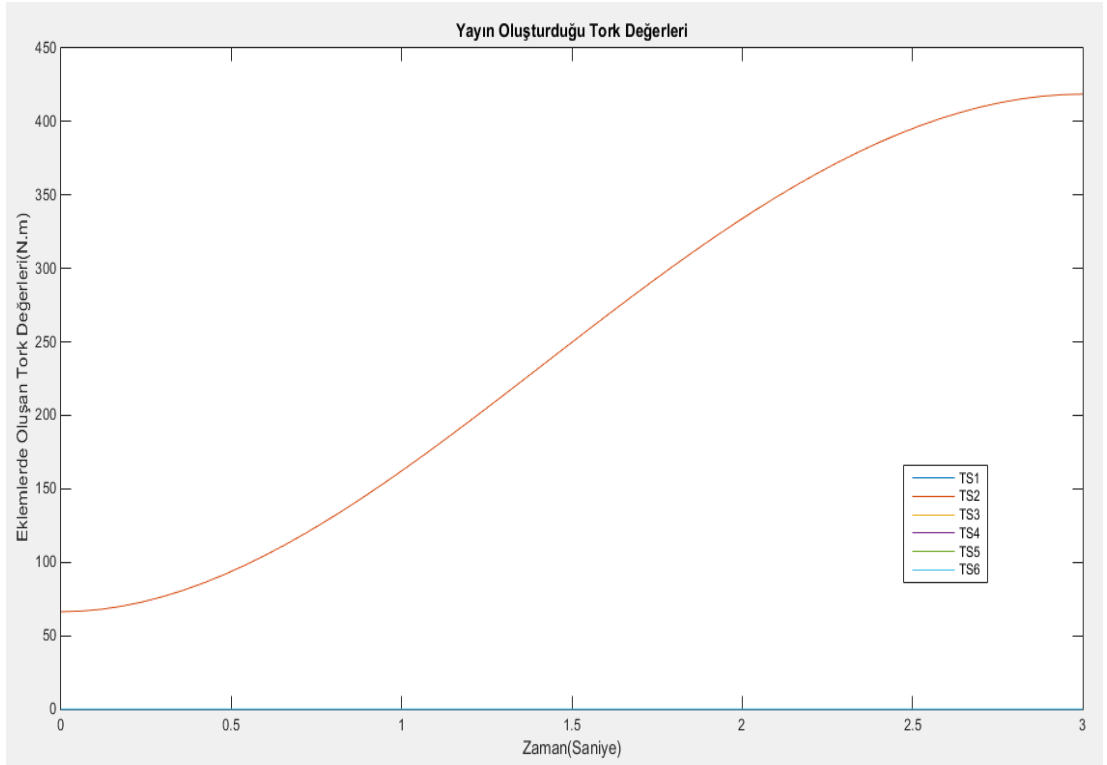
Matlab Simulink ortamında oluşturulan dinamik model benzetimi 3 saniyelik benzetim süresinde tamamlanmaktadır. Oluşturulan dinamik model benzetimine ait blok diyagramlar ekler kısmında verilmektedir. Dinamik modeli oluşturan merkezcil ve Coriolis, yer çekimi, yay ve sürtünme torkları şekil 2.19, şekil 2.20, şekil 2.21 ve şekil 2.22’de gösterilmektedir.



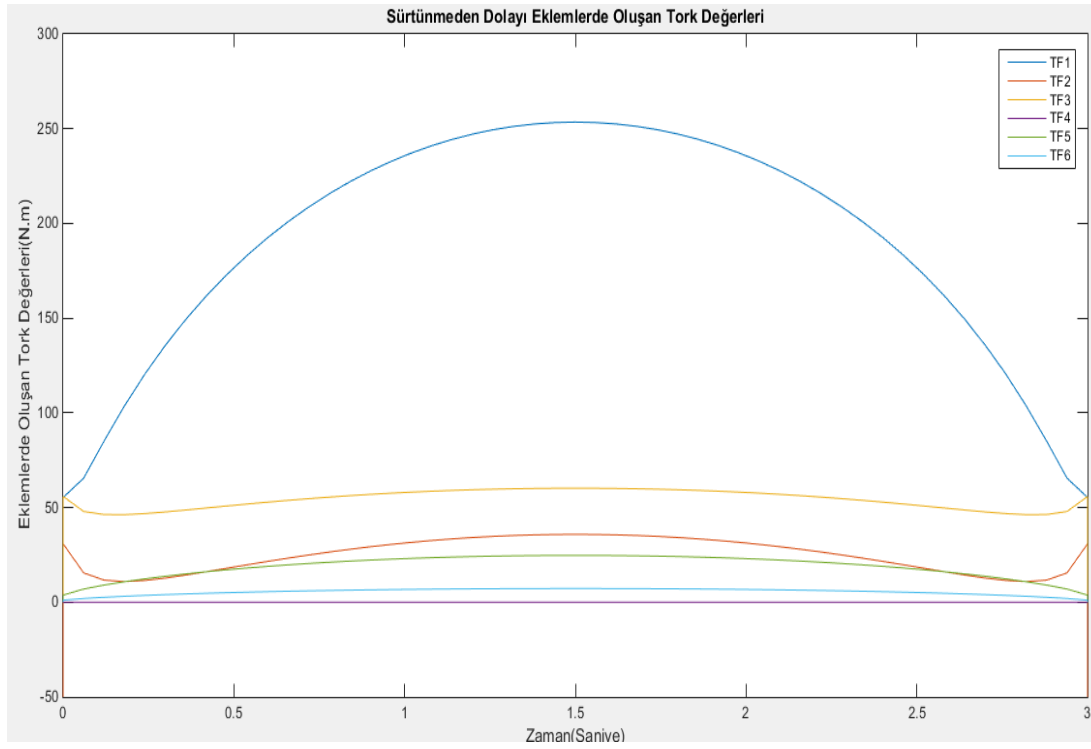
Şekil 2.22: Merkezci ve Coriolis tork değerleri.



Şekil 2.23: Yer çekimi etkisi ile oluşan tork değerleri.

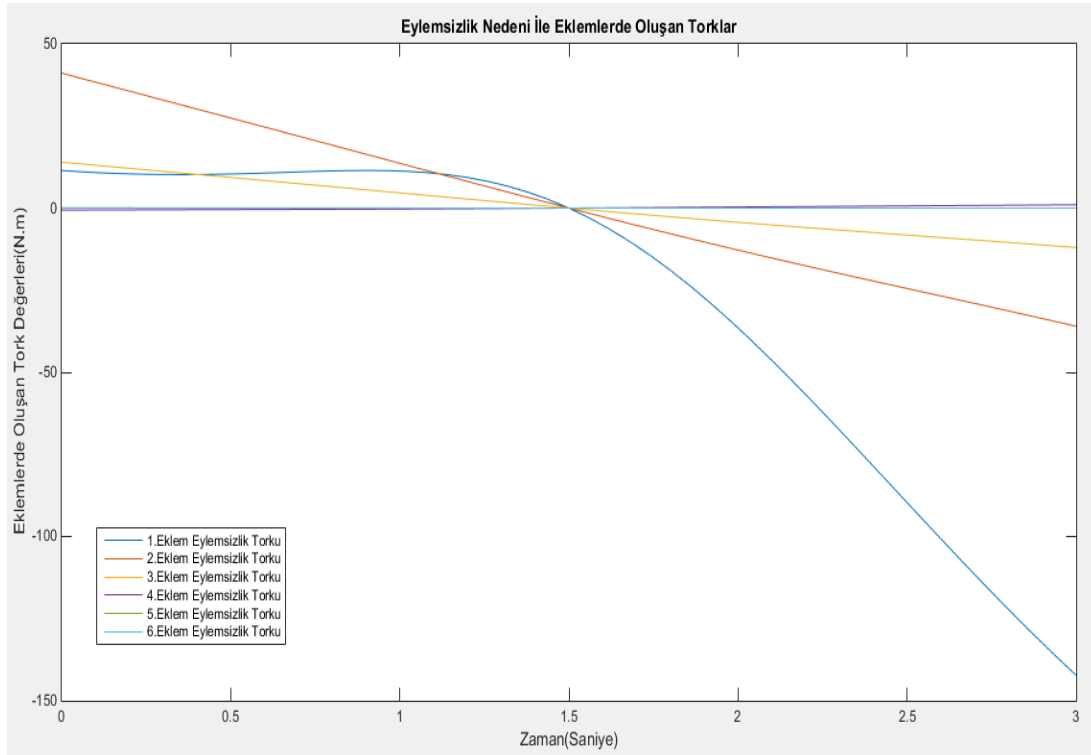


Şekil 2.24: 2.eklemdeki yayın oluşturduğu tork değeri.



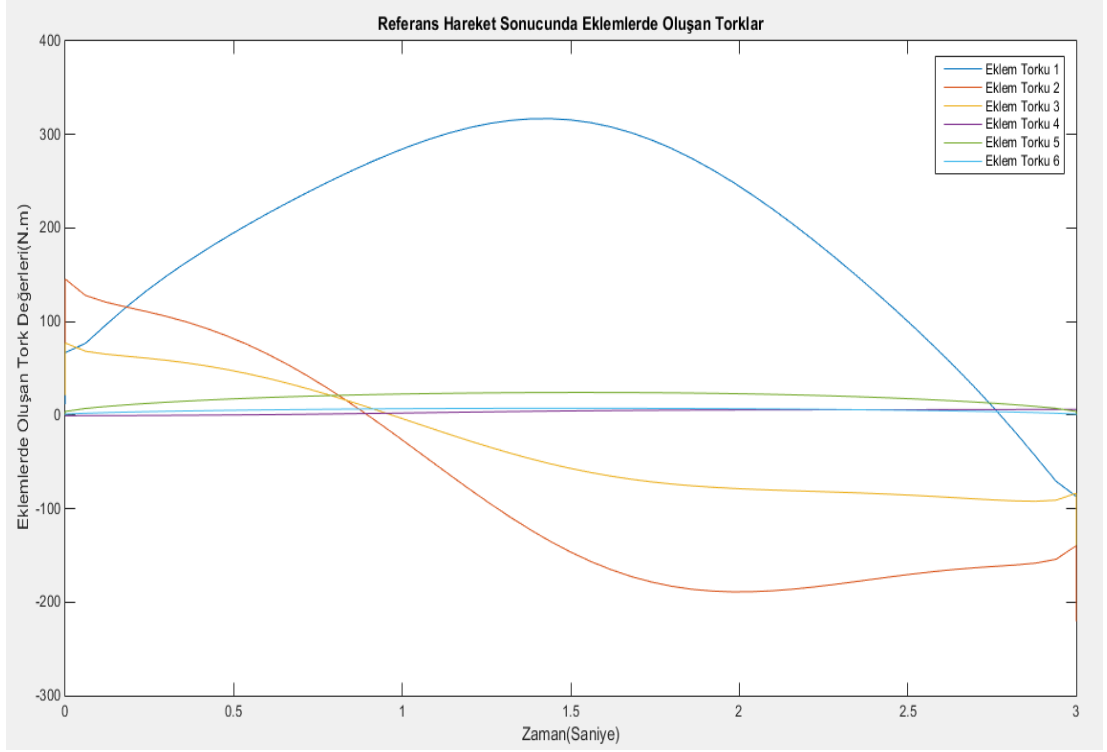
Şekil 2.25: Sürtünmeden dolayı eklemlerde oluşan tork değerleri.

Şekil 2.26’da Eylemsizlik matrisi ile ivme yörüngesinin çarpımı sonucunda oluşan tork değerleri gösterilmektedir.



Şekil 2.26: Eylemsizlik matrisi ve ivme vektörünün çarpımı ile oluşan torklar.

Şekil 2.22’den şekil 2.26’e kadar gözükten tork değerleri toplanarak, referans yörünge hareketlerini yapabilmesi için robotun eklemlerine uygulaması gereken tork değerleri bulunmaktadır.



Şekil 2.27: Referans hareketi yapabilmesi için eklemlere verilen torklar.

3.KONTROL SİSTEMİ TASARIMI VE BENZETİMLER

3.1 Konum Kontrolü

Robot uç noktasının hareketini kontrol etmek için birçok kontrol algoritması kullanılmaktadır. Endüstride, seri manipülatörlerin hareketi kontrol edilirken çoğunlukla merkezi olmayan PID kontrolü tercih edilmektedir. PID kontrolü uygulanırken her bir eklem için ayrı ayrı kontrolcü katsayıları ayarlanmaktadır. PID kontrolünün uygulanması basit, maliyeti düşüktür. Fakat robot dinamiğinin hareketinin değişmesi özellikle yüksek hızlarda kontrolcü performansını oldukça düşürmektedir. Bu dezavantajdan dolayı tez sırasında robotun dinamik modelinden yararlanan ters dinamik kontrol kullanılmaktadır. Ters dinamik kontrol ile doğrusal olmayan robot hareketi doğrusal hale getirilmektedir. Oluşturulan ters dinamik kontrolcüsü ile robotun konumuna, hızına ve ivmesine bağlı olarak, robot hareketini sağlayacak tork değerleri bulunmaktadır. Tork değerleri bulunurken robotun ters dinamik modeli kullanıldığı için, bu kontrol yöntemine ters dinamik kontrol denilmektedir. Ters dinamik kontrol hem eklem uzayı, hem de görev uzayı için kullanılmaktadır. Tez sırasında öncelikli olarak robot uç noktası kontrol edilmeye çalışıldığı için, görev uzayında ters dinamik kontrol kullanılmaktadır.

Tez sırasında, görev uzayında ters dinamik kontrolü gerçekleştirebilmek için literatürde yaygınlıkla kullanılan iki kontrol algoritması kullanılmaktadır. İlk kontrol algoritmasında görev uzayı için tanımlanan yörüngeler ters geometrik ve kinematik modeller kullanılarak eklem uzayına dönüştürülüp, kontrol edilmektedir. İkinci yöntemde ise doğrudan görev uzayı için oluşturulan konum, hız ve ivme yörüngeleri kullanılarak robot uç noktası kontrol edilmektedir.

3.1.1 Görev uzayında kontrolcü şeması

Şekil 3.1’de yer alan hareket kontrol algoritmasındaki yörünge, görev uzayı için tanımlanmaktadır. Daha sonra oluşturulan bu yörünge, bölüm 2’de elde edilen geometrik ve kinematik modeller yardımıyla eklem uzayında tanımlı hale

Denklem 3.7’de yer alan K_p ve K_d matrisleri orantısal ve türevsel kontrol katsayılarını temsil etmektedir. Denklem 3.6’nın sol tarafı ile denklem 3.7’nin sağ tarafı birbirine eşitlenir ve oluşacak eşitliğin sağ tarafı sıfır olarak bırakılırsa, sistemin kapalı çevrim cevabını belirleyen hata denklemi bulunmaktadır.

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (3.8)$$

Denklem 3.8’de yer alan e vektörü $q^d - q$ ifadesine eşit olup, hata vektörünü temsil etmektedir. Denklem 3.8’in Laplace dönüşümü alınıp, sistemin doğal frekansı ve sönüm katsayısı cinsinden yazılırsa, i. ekleme ait K_{pi} ve K_{di} kazanç değerleri sistemin arzu edilen davranışı gösterebilmesi için ayarlanabilmektedir.

$$e(s)(s^2 + K_d s + K_p) = 0 \quad (3.9)$$

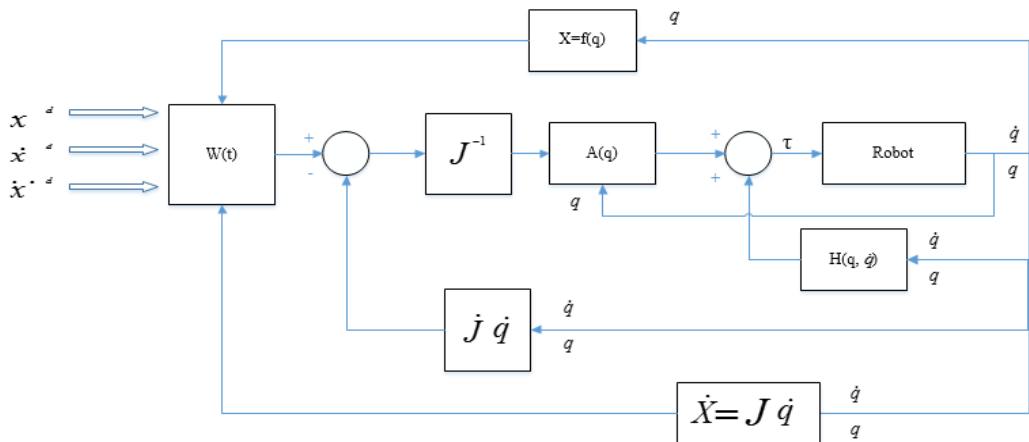
$$s^2 + 2\xi\omega_s + \omega^2 = 0 \quad (3.10)$$

Denklem 3.9 ve 3.10’un sol tarafları eşitlenirse i. eklemin kazanç katsayıları olan K_{pi} ve K_{di} değerleri denklem 3.11’de gösterildiği gibi yazılabilmektedir.

$$K_{pj} = \omega_j^2 \quad K_{dj} = 2\xi_j\omega_j \quad (3.11)$$

Tez sırasında, Staubli RX-160 manipülatörünün kritik olarak sönümlenmesi istediği için ξ_j değeri 1 olarak seçilmektedir.

Görev uzayında robot uç noktasını kontrol etmek için kullanılan diğer kontrol algoritması ise şekil 3.2’de gösterilmektedir. Kararlı ivme kontrolü adı da verilen bu kontrol algoritmasında doğrudan görev uzayında tanımlanan yörünge hatası kullanılarak kontrolcü tasarlanmaktadır.



Şekil 3.2: Doğrudan görev uzayında ters dinamik kontrol.

Şekil 3.2’ e göre, robotun görev uzayında tanımlanan yörüngeyi takip edebilmesi için eklemeler verilmesi gereken tork değerlerini veren ifade denklem 3.12 ile verilmektedir.

$$\tau = A(q)J^{-1}(W(t) - \dot{J}\dot{q}) + H(q) \quad (3.12)$$

$W(t)$ kontrol vektörü, tam dinamik modelin elde edildiği varsayılarak, görev uzayında tanımlı olan robot uç noktasının ivmesine eşit olarak yazılabilmektedir.

$$\ddot{X} = W(t) \quad (3.13)$$

Kontrol vektörü W , denklem 3.14’de PD kontrolcü kullanılarak yazılmaktadır.

$$W(t) = \ddot{X}^d + K_d(\dot{X}^d - \dot{X}) + K_p(X^d - X) \quad (3.14)$$

Robotun kapalı çevrim davranışı, denklem 3.13 ve 3.14’den yararlanılarak oluşturulan hata denklemi ile açıklanabilmektedir.

$$\ddot{e}_x + K_d\dot{e}_x + K_p e_x = 0 \quad (3.15)$$

Denklem 3.15’de yer alan e_x hata vektörü $X^d - X$ ifadesine eşittir. Yukarıda yer alan denklemin Laplace dönüşümü alınıp sistemin doğal frekansı ve sönüm katsayısı cinsinden yazılırsa uygun K_p ve K_d kontrol katsayıları elde edilebilmektedir. Böylece robot, arzu edilen dinamik yapıyı sağlayabilmektedir.

$$e_x(s)(s^2 + K_d s + K_p) = 0 \quad (3.16)$$

$$K_{pj} = \omega_j^2 \quad K_{dj} = 2\xi_j \omega_j \quad (3.17)$$

Şekil 3.2’de yer alan kontrol algoritmasının uygulanması kolay olmamaktadır. Bunun en büyük nedeni, dönme hareketinin kontrolünün zor olmasıdır. Bu yüzden, Staubli RX-160 manipülatörünün görev uzayında kontrolü yapılırken, Khalil tarafından tasarlanan algoritma kullanılmaktadır. Khalil’in kullandığı algoritma, öteleme ve dönme hareketlerinin hata vektörlerini ayrı ayrı hesaplamaktadır. Denklem 3.18’de hata vektörünün(e_x) iki farklı kısma ayrılışı gösterilmektedir.

$$e_x = [dX_p^T \quad dX_r^T]^T \quad (3.18)$$

Öteleme ve dönme hata vektörleri denklem 3.19 ve 3.20’de ayrı ayrı olarak hesaplanmaktadır.

$$dX_p = 0_{p_6^d} - p_6^0 \quad (3.19)$$

$$dX_r = \frac{1}{2} \left[s_6^0 \times 0_{s_d} + n_6^0 \times 0_{n_d} + a_6^0 \times 0_{a_d} \right] \quad (3.20)$$

Denklem 3.19 ve 3.20’de yer alan p_6^0 , s_6^0 , n_6^0 ve a_6^0 3×1 öteleme ve dönme vektörleri, uç noktanın sıfırıncı eksen takımına göre hareketini veren denklem 3.21’de yer almaktadırlar.

$$T_6^0 = \begin{bmatrix} s_6^0 & n_6^0 & a_6^0 & p_6^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

3.1.2 Yörünge planlama

Robotlar, planlanan bir yörünge boyunca hareket ettirilmeye çalışılırlar. Oluşturulacak bu yörünge ya eklem uzayı ya da görev uzayı için tanımlanmaktadır. Eklem uzayında yörünge hesaplanırken robot eklemlerinin uygulaması gereken referans eklem konumları, hızları ve ivmeleri hesaplanmaktadır. Görev uzayında ise, robot uç noktasının istenen hareketleri yapabilmesi için, uç noktaya ait konum, hız ve ivme vektörleri hesaplanmaktadır. Yapılan tezin amacı Staubli RX-160 manipülörünün görev uzayında gerçekleşen konum ve kuvvet kontrolü olduğu için, oluşturulacak yörüngeler görev uzayında tanımlanmaktadır. Bu amaç doğrultusunda Staubli RX-160 manipülörünün uç noktasına ait 3 tane çizgisel konum(x, y, z), 3 tane açısal konum(α, β, γ), 6 tane hız ve 6 tane ivme vektörleri hesaplanarak yörünge oluşturulmaktadır. Fakat görev uzayında tanımlanan bir yörüngeciin ciddi dezavantajları olabilmektedir. Hesaplanan yörünge tekil noktalara takılırsa, robot hareketi gerçekleşmeyebilmektedir. Bir diğer problem ise, oluşturulacak yörünge, eklemlerden birinin veya birkaçının çalışma sınırlarının dışına çıkmasına neden olabilmektedir. Bu nedenle kullanılacak yörüngeler oluşturulurken bu olası problemler göz önünde bulundurulmaktadır.

Staubli RX-160 robotuna ait konum ve kuvvet kontrol algoritmaları görev uzayında gerçekleştirildiği için görev uzayında noktadan noktaya yörüngeler planlanmaktadır. T_E^i ve T_E^f robot uç noktasının bulunmasının istendiği ilk ve son noktaları tanımlayan 4×4 homojen matrislerdir. Görev uzayında tanımlı yörünge kullanılarak, robotun bu iki nokta arasındaki hareketi sağlanmaktadır. İlk ve son noktaları tanımlayan bu iki matris denklem 3.22 ile verilmektedir.

$$T_E^i = \begin{bmatrix} A^i & P^i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_E^f = \begin{bmatrix} A^f & P^f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

Manipülatör uç noktasını görev uzayında bir noktadan diğer bir noktaya götürebilmek için, T_E^i ve T_E^f noktaları arasında oluşacak doğrusal bir öteleme hareketi ve uç noktanın dönme eksenine göre oluşacak olan bir α açısal hareketini yapmak gerekmektedir.

İki nokta arasında gerçekleşmekte olan öteleme hareketine ait matematiksel tanım, denklem 3.23 ile verilmektedir.

$$D = \|P^f - P^i\| = \sqrt{(P_x^f - P_x^i)^2 + (P_y^f - P_y^i)^2 + (P_z^f - P_z^i)^2} \quad (3.23)$$

Görev uzayında oluşturulan noktadan noktaya öteleme hareketini zamanın fonksiyonu olarak veren denklem, aşağıda paylaşılmaktadır.

$$P(t) = P^i + r(t)(P^f - P^i) \quad (3.24)$$

Denklem 3.24’de tanımlanan t ifadesi zamanı, $r(t)$ ifadesi ise interpolasyon fonksiyonunu temsil etmektedir.

Robot uç noktasının bir noktadan diğer noktaya yapmış olduğu dönme hareketi, $\text{rot}(u, \alpha)$ matrisi olarak verilebilmektedir. 3×3 $\text{rot}(u, \alpha)$ matrisi, u vektörü çevresindeki α rotasyon açısını temsil etmektedir.

$$A^i \text{rot}(u, \alpha) = A^f \quad (3.25)$$

Denklem 3.25’de yer alan 3×3 A matrisi, $[s \ n \ a]$ şeklinde yazılıp, $\text{rot}(u, \alpha)$ ifadesi yalnız bırakılırsa aşağıda yer alan denklemler grubu oluşmaktadır.

$$\text{rot}(u, \alpha) = [A^i]^T A^f \quad (3.26)$$

$$\text{rot}(u, \alpha) = \begin{bmatrix} s^i & n^i & a^i \\ s^f & n^f & a^f \end{bmatrix} = \begin{bmatrix} s^i \cdot s^f & s^i \cdot n^f & s^i \cdot a^f \\ n^i \cdot s^f & n^i \cdot n^f & n^i \cdot a^f \\ a^i \cdot s^f & a^i \cdot n^f & a^i \cdot a^f \end{bmatrix} \quad (3.27)$$

$$\cos \alpha = \frac{1}{2} [s^i \cdot s^f + n^i \cdot n^f + a^i \cdot a^f - 1] \quad (3.28)$$

$$\sin \alpha = \frac{1}{2} \sqrt{(a^i \cdot n^f - n^i \cdot a^f)^2 + (s^i \cdot a^f - a^i \cdot s^f)^2 + (n^i \cdot s^f - s^i \cdot n^f)^2} \quad (3.29)$$

Denklem 3.28 ve 3.29 kullanılarak α açısı ve 3×1 u birim vektörü hesaplanmaktadır.

$$\alpha = \text{atan2}(\sin \alpha, \cos \alpha) \quad (3.30)$$

$$u = \frac{1}{2\sin(\alpha)} \begin{bmatrix} a^i \cdot n^f - n^i \cdot a^f \\ s^i \cdot a^f - a^i \cdot s^f \\ n^i \cdot s^f - s^i \cdot n^f \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \quad (3.31)$$

İnterpolasyon polinomu kullanılarak $\text{rot}(u, \alpha)$ matrisi, zamana bağlı bir fonksiyon olarak yazılabilmektedir. Oluşturulacak fonksiyonda zaman değişkenine 0 verilirse robotun ilk konumuna ait dönme hareketi, zaman değişkeni yerine hesaplanan toplam hareket süresi verilirse robot uç noktası son konumuna ait dönme hareketi bulunmaktadır.

$$A(t) = A^i \text{rot}(u, r(t)\alpha) \quad (3.32)$$

Denklem 3.32 oluşturulduktan sonra önemli olan $\text{rot}(u, r(t)\alpha)$ matrisinin nasıl elde edileceğidir. Bu matrise ait elemanları oluşturabilmek için denklem 3.27'den, denklem 3.31'e kadar olan denklemler kullanılmaktadır. Elde edilen matris elemanları, denklem 3.33'den başlayarak denklem 3.41'e kadar verilmektedir.

$$\text{rot}(u, r(t)\alpha)(1,1) = u_x^2(1 - \cos(r(t)\alpha)) + \cos(r(t)\alpha) \quad (3.33)$$

$$\text{rot}(u, r(t)\alpha)(1,2) = u_x u_y(1 - \cos(r(t)\alpha)) - u_z s(r(t)\alpha) \quad (3.34)$$

$$\text{rot}(u, r(t)\alpha)(1,3) = u_x u_z(1 - \cos(r(t)\alpha)) + u_y s(r(t)\alpha) \quad (3.35)$$

$$\text{rot}(u, r(t)\alpha)(2,1) = u_x u_y(1 - \cos(r(t)\alpha)) + u_z s(r(t)\alpha) \quad (3.36)$$

$$\text{rot}(u, r(t)\alpha)(2,2) = u_y^2(1 - \cos(r(t)\alpha)) + \cos(r(t)\alpha) \quad (3.37)$$

$$\text{rot}(u, r(t)\alpha)(2,3) = u_y u_z(1 - \cos(r(t)\alpha)) - u_x s(r(t)\alpha) \quad (3.38)$$

$$\text{rot}(u, r(t)\alpha)(3,1) = u_x u_z(1 - \cos(r(t)\alpha)) - u_y s(r(t)\alpha) \quad (3.39)$$

$$\text{rot}(u, r(t)\alpha)(3,2) = u_y u_z(1 - \cos(r(t)\alpha)) + u_x s(r(t)\alpha) \quad (3.40)$$

$$\text{rot}(u, r(t)\alpha)(3,3) = u_z^2(1 - \cos(r(t)\alpha)) + \cos(r(t)\alpha) \quad (3.41)$$

Öteleme ve dönme hareketlerinin yörüngeleri elde edildikten sonra, bu iki yörünge birleştirilerek konum yörüngesine ait 6×1 vektör bulunmaktadır. Konum yörüngesinin zamana bağlı türevi alınarak hız yörüngesi, hız yörüngesinin zamana bağlı türevi alınarak ivme yörüngesi elde edilmektedir.

Görev uzayında yörünge planlamak için kullanılan denklemlerde $r(t)$ olarak kullanılan ifade, interpolasyon polinomunu ifade etmektedir. Tez sırasında

interpolasyon polinomu olarak üçüncü ve beşinci dereceden polinomlar kullanılmaktadır.

Eğer oluşturulacak hareket yörüngesinde ilk ve son hızların sıfır olması isteniyorsa, kullanılması gereken interpolasyon polinomu en az üçüncü dereceden olmalıdır. Üçüncü derece polinoma özel olarak kübik polinom ismi de verilmektedir.

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (3.42)$$

Denklem 3.42’de yer alan katsayılar ilk ve son hızın 0 olması gibi istenen sınır koşulları uygulanarak bulunmaktadır. İlgili katsayılar denklem 3.43’de verilmektedir.

$$a_0 = q^i \quad a_1 = 0 \quad a_2 = \frac{3}{tf^2}D \quad a_3 = -\frac{2}{tf^3}D \quad (3.43)$$

Denklem 3.43’de yer alan tf toplam hareket süresini, D ise ilk ve son hareket arasındaki farkı göstermektedir. İlk ve son nokta arasında oluşturulan yörünge denklem 3.44 ile üçüncü dereceden polinom ise denklem 3.45 ile gösterilmektedir.

$$q(t) = q^i + r(t)D \quad 0 \leq t \leq tf \quad (3.44)$$

$$r(t) = 3\left(\frac{t}{tf}\right)^2 - 2\left(\frac{t}{tf}\right)^3 \quad (3.45)$$

Denklem 3.44’de konum yörüngesini veren ifadenin zamana göre birinci ve ikinci türevleri alınırsa, hız ve ivme yörüngeleri elde edilmektedir.

$$\dot{q}(t) = \dot{r}(t)D \quad (3.46)$$

$$\ddot{q}(t) = \ddot{r}(t)D \quad (3.47)$$

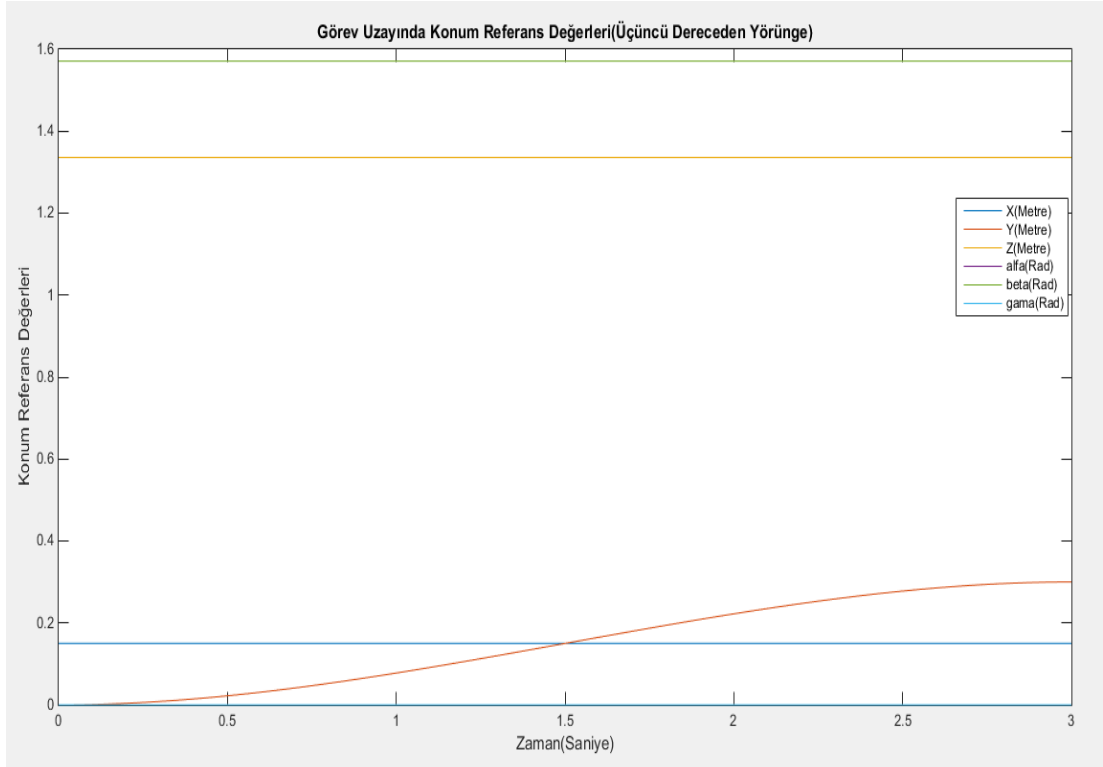
Daha yüksek hızlarda çalışan veya daha ağır yükleri kaldırmaya çalışan robotlar için hareket ivmesinin devamlılığı önem arz etmektedir. Bu yüzden ilk ve son hareket ivmelerinin sıfır olması istenmektedir. Bu koşulları sağlayacak polinom ise en az beşinci dereceden olmaktadır. Gerekli kısıtlar uygulanırsa denklem 3.48’de yer alan beşinci dereceden interpolasyon polinomu elde edilmektedir.

$$r(t) = 10\left(\frac{t}{tf}\right)^3 - 15\left(\frac{t}{tf}\right)^4 + 6\left(\frac{t}{tf}\right)^5 \quad (3.48)$$

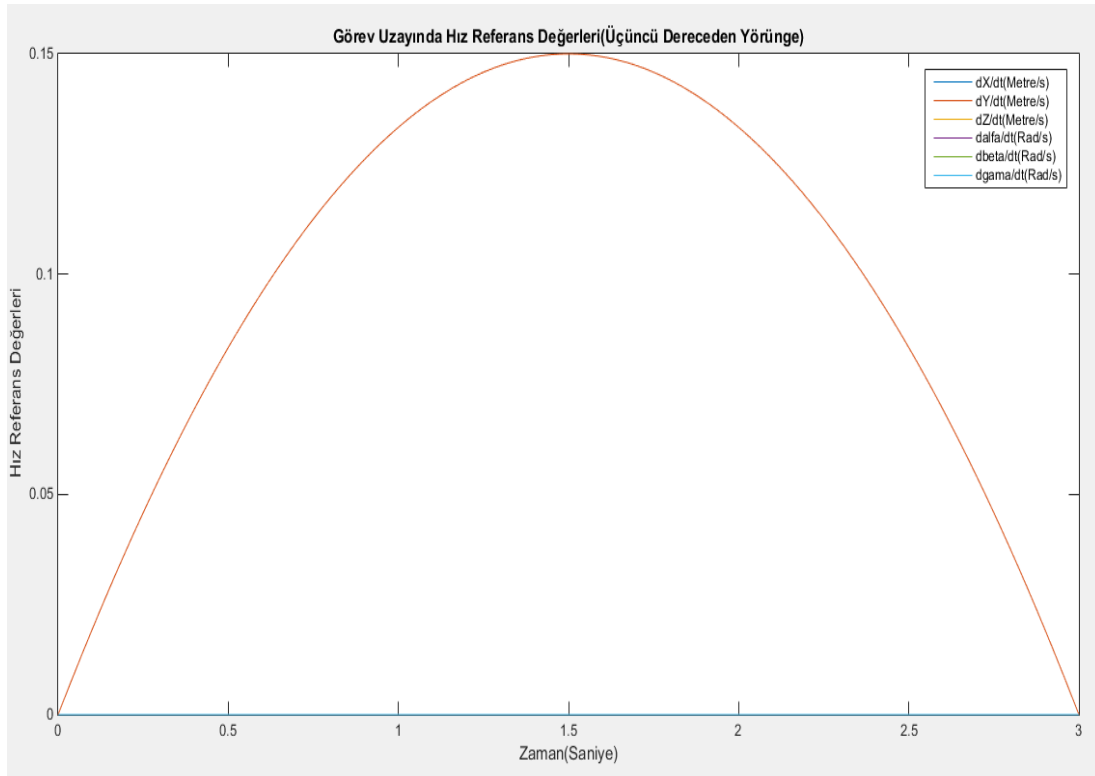
3.1.3 Benzetim sonuçları

Bölüm 3.1.1 ve 3.1.2’de konum kontrolü için geliştirilen kontrol algoritmaları ve yörüngelerin benzetimleri Matlab Simulink programı kullanılarak yapılmaktadır.

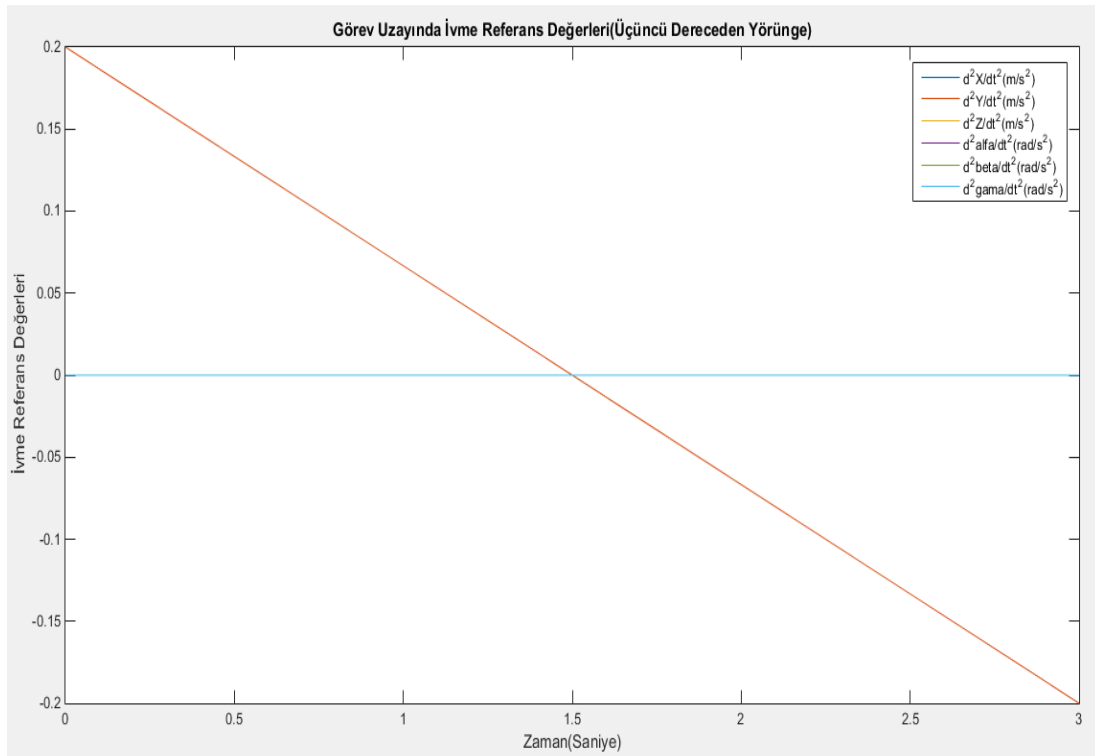
Konum kontrolü için iki farklı algoritma kullanılmaktadır. İlk algoritmada görev uzayında oluşturulan yörünge geometrik ve kinematik modeller kullanılarak eklem uzayında tanımlı hale dönüştürülmektedir. İkinci konum kontrolü algoritmasında ise doğrudan görev uzayında tanımlanan konum, hız ve ivme referans vektörleri PD kontrol algoritması içerisinde kullanılmaktadır. Görev uzayında ilk konum kontrolü algoritması olan, eklemlerin kontrolüne dayanan ters dinamik kontrolde üçüncü dereceden yörünge referans girişi olarak uygulanmaktadır. Görev uzayında oluşturulan ikinci konum kontrolü algoritması olan kararlı ivme kontrolünde ise referans giriş değerleri olarak beşinci dereceden yörünge kullanılmaktadır. Ayrıca Matlab Simulink programında oluşturulan konum kontrolü blok diyagramları, ekler kısmında paylaşılmaktadır. Sonuçları verilmekte olan ilk algoritma, görev uzayında eklemlerin kontrolüne dayanan ters dinamik kontrolüdür.



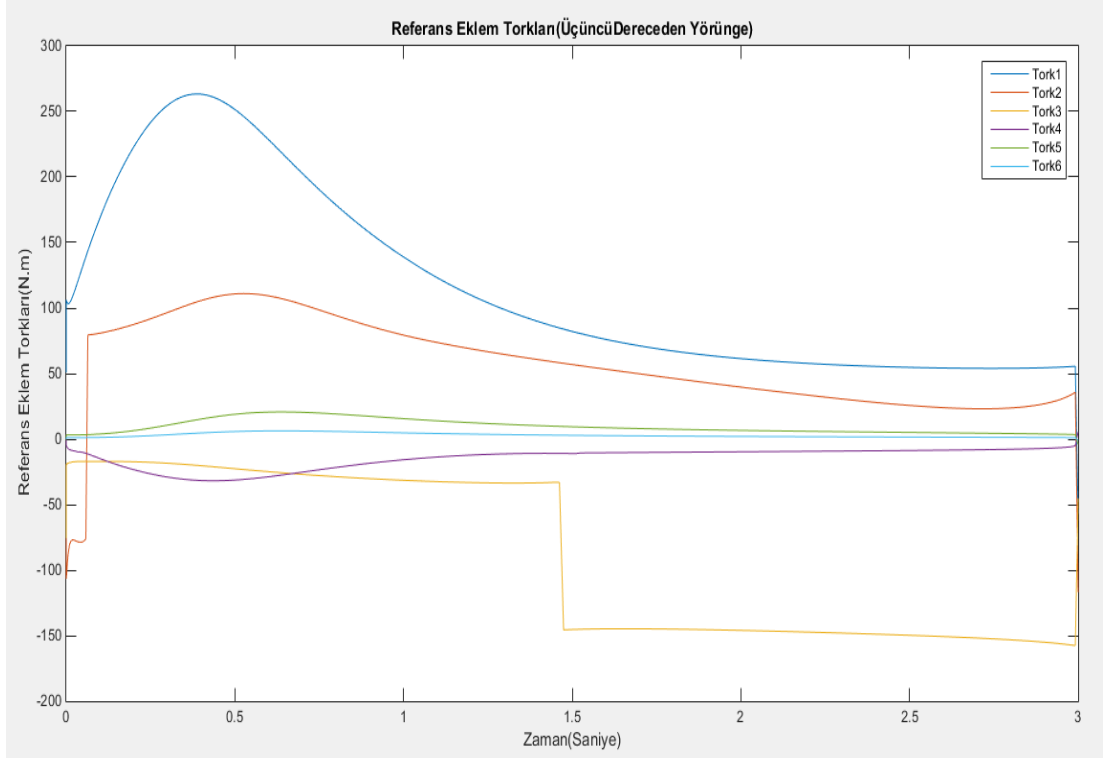
Şekil 3.3: Görev uzayında konum referans değerleri(üçüncü dereceden yörünge).



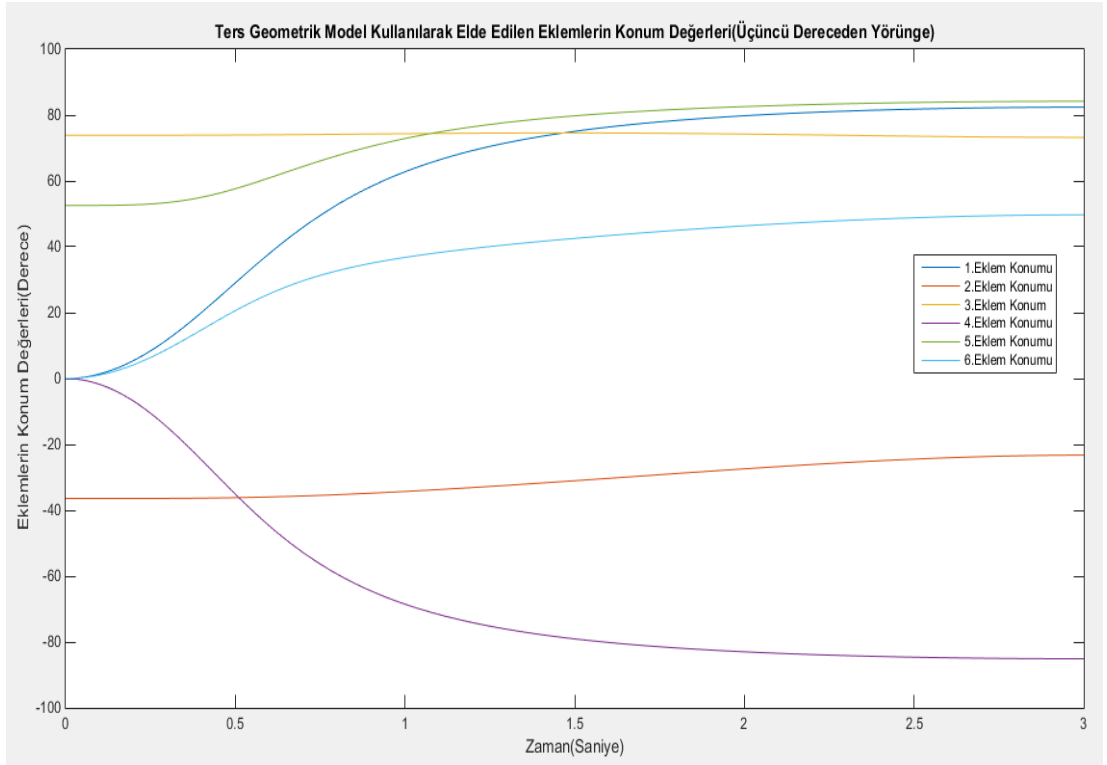
Şekil 3.4: Görev uzayında hız referans değerleri(üçüncü dereceden yörünge).



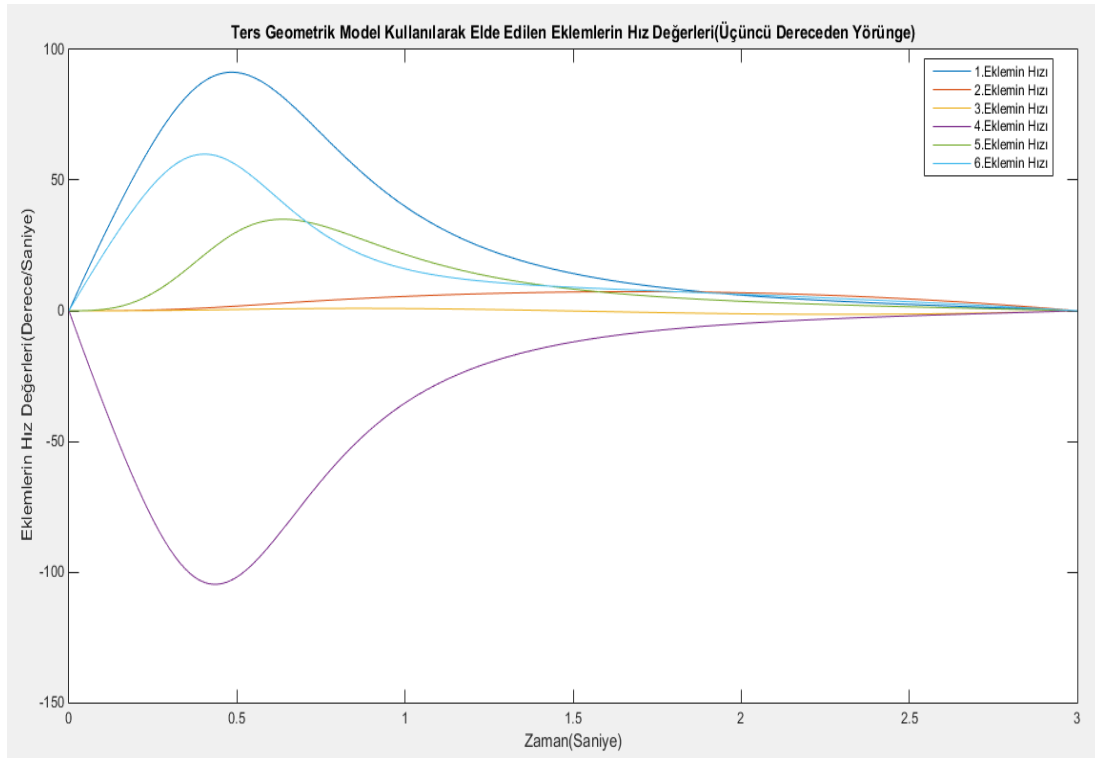
Şekil 3.5: Görev uzayında ivme referans değerleri(üçüncü dereceden yörünge).



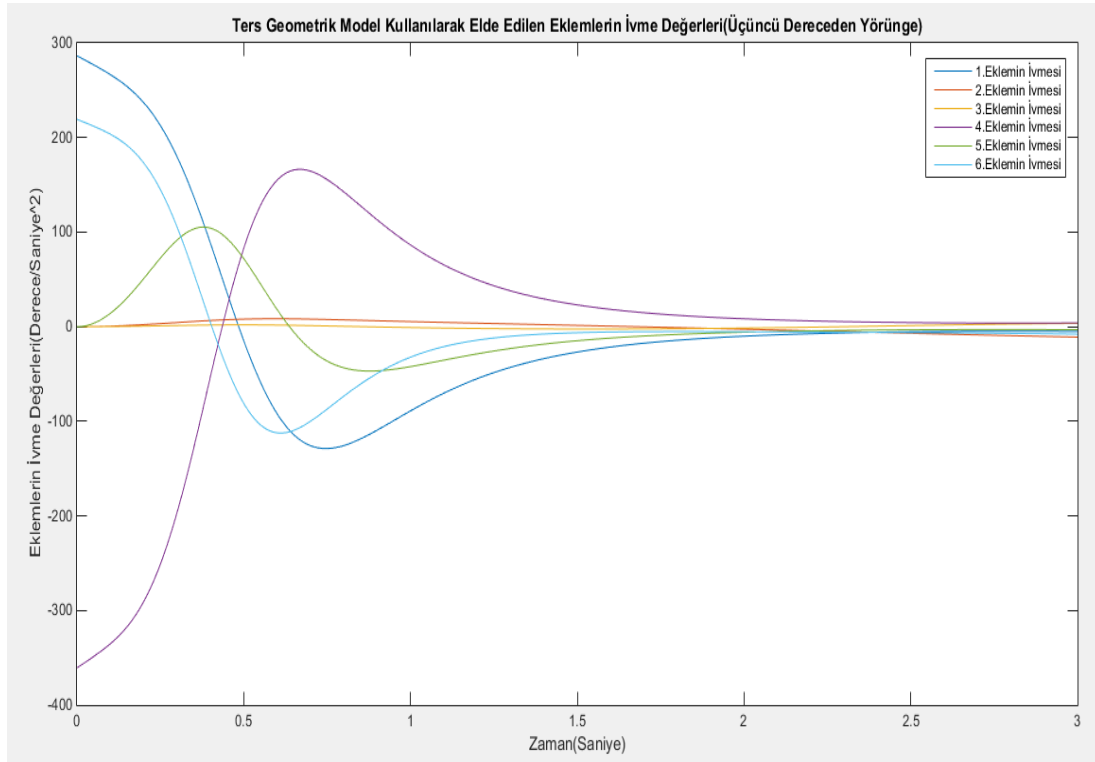
Şekil 3.6: Referans eklem torkları(üçüncü dereceden yörünge).



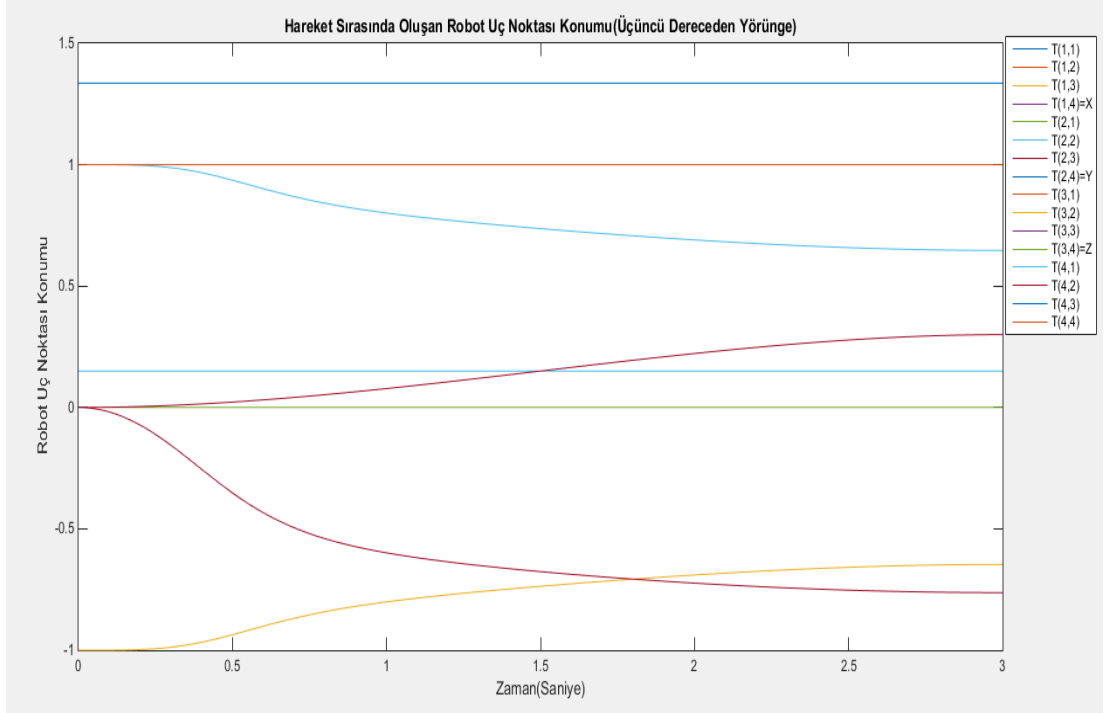
Şekil 3.7: Eklemlerin konum değerleri(üçüncü dereceden yörünge).



Şekil 3.8: Eklemlerin hız değerleri(üçüncü dereceden yörünge).

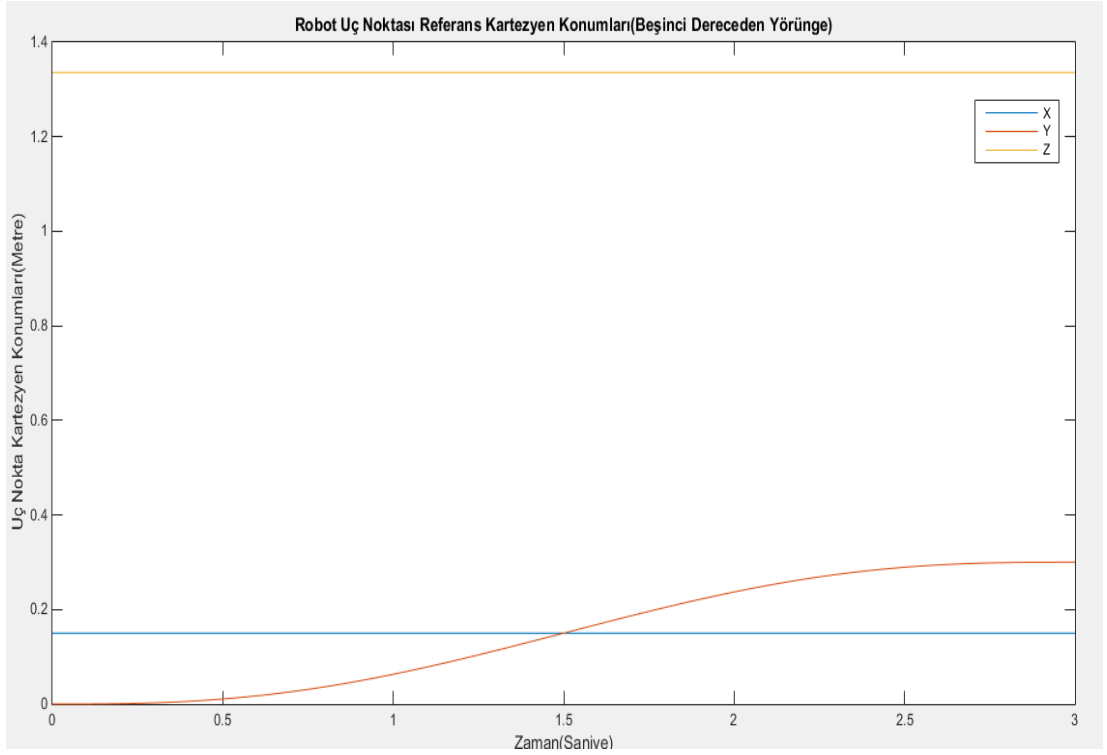


Şekil 3.9: Eklemlerin ivme değerleri(üçüncü dereceden yörünge).

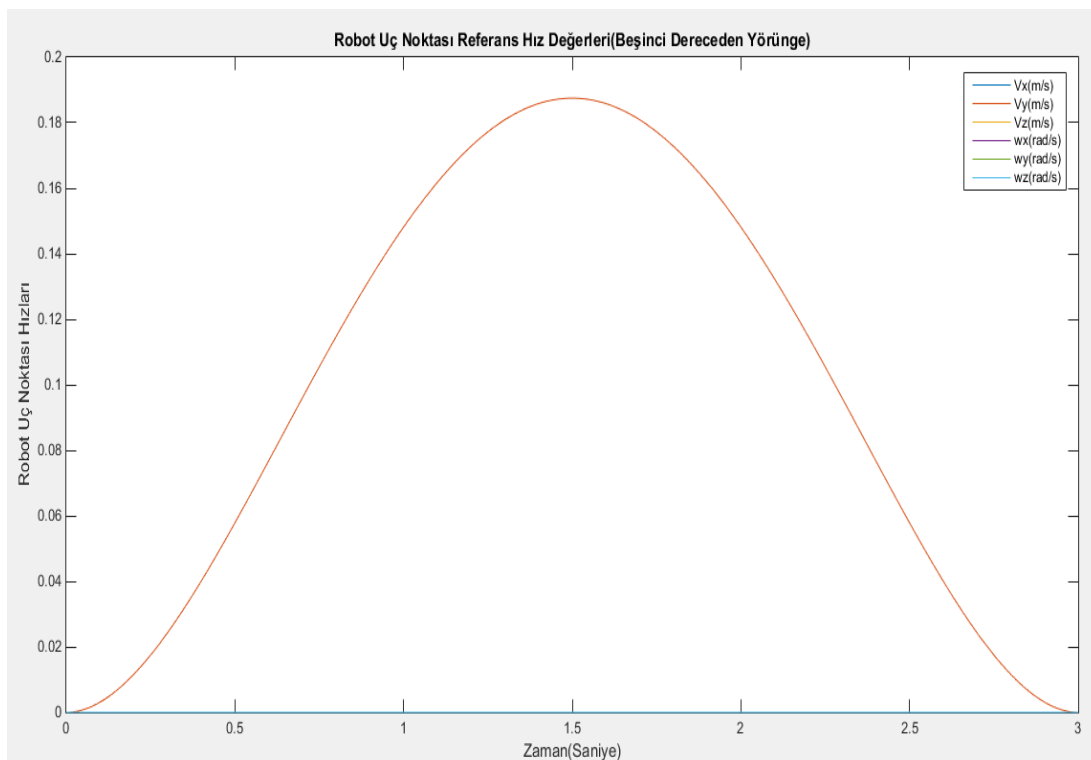
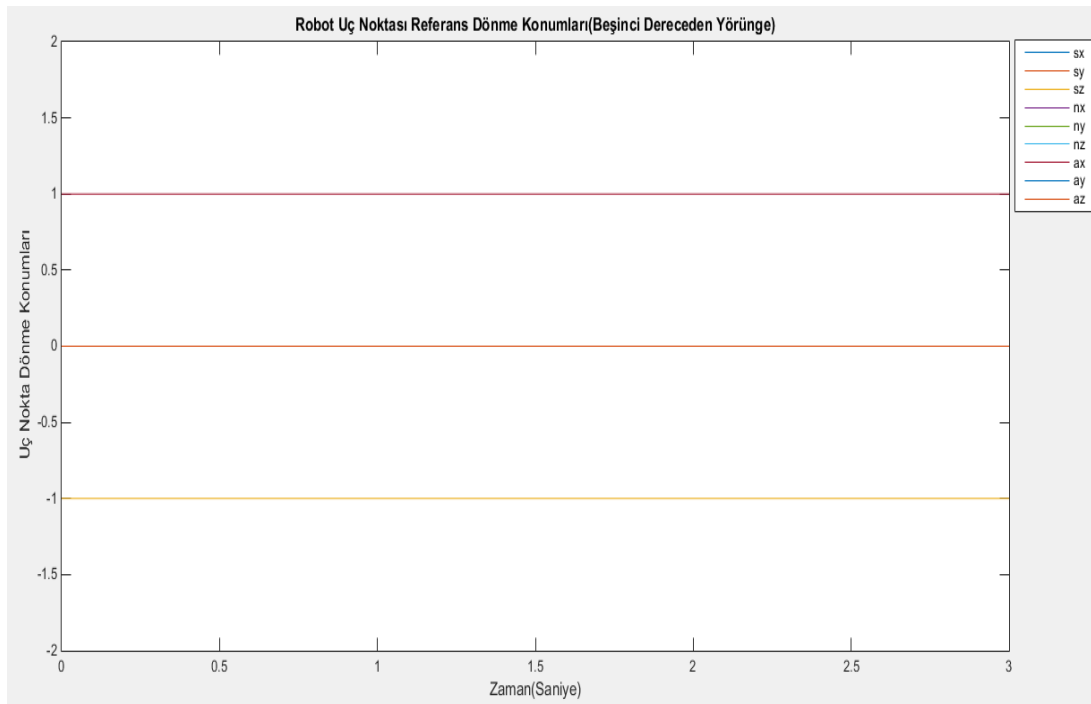


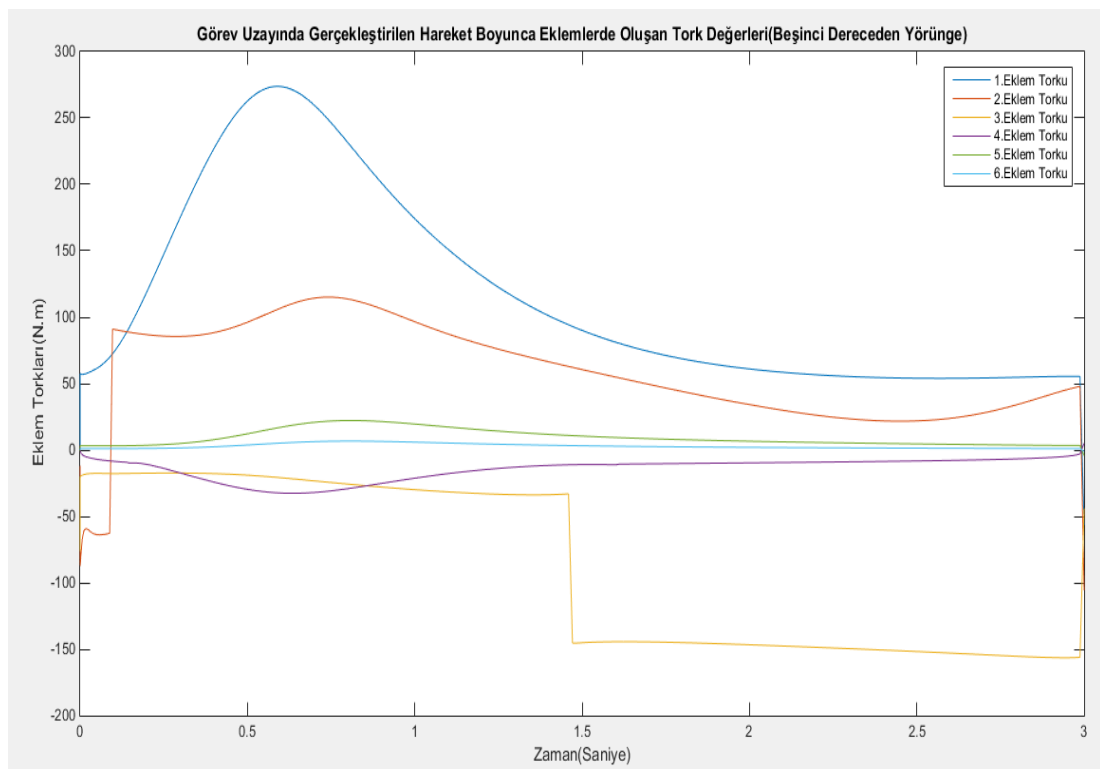
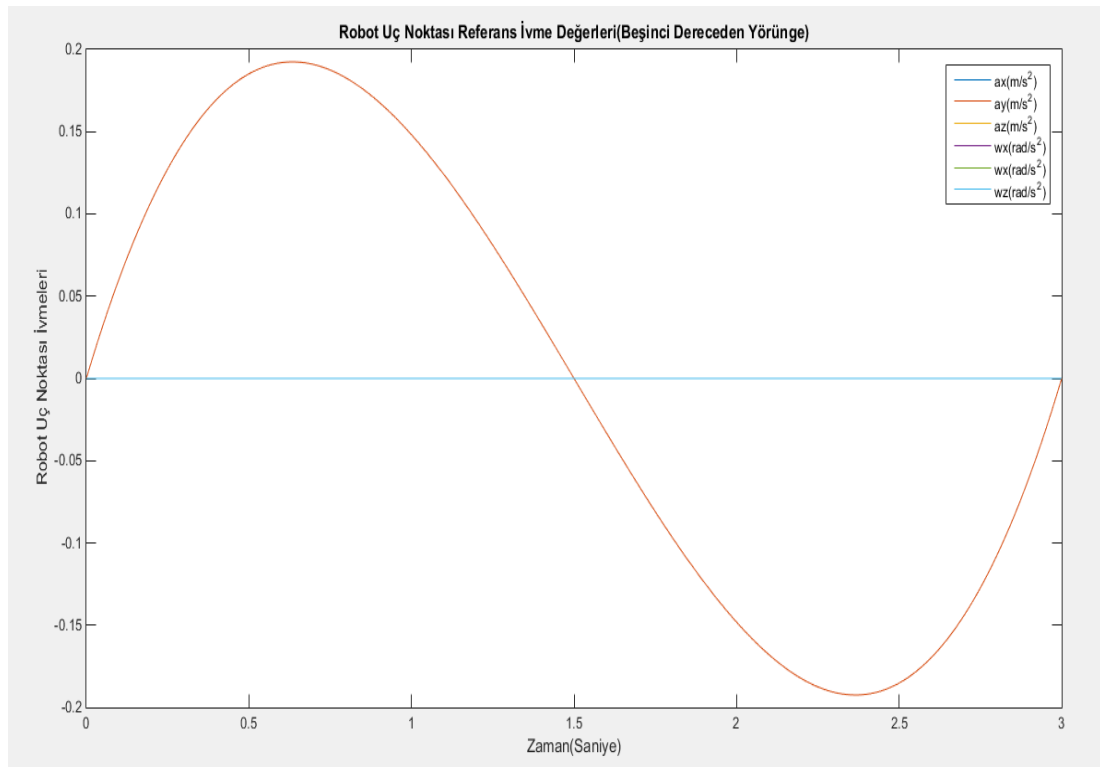
Şekil 3.10: Robot uç noktası konumu (üçüncü dereceden yörünge).

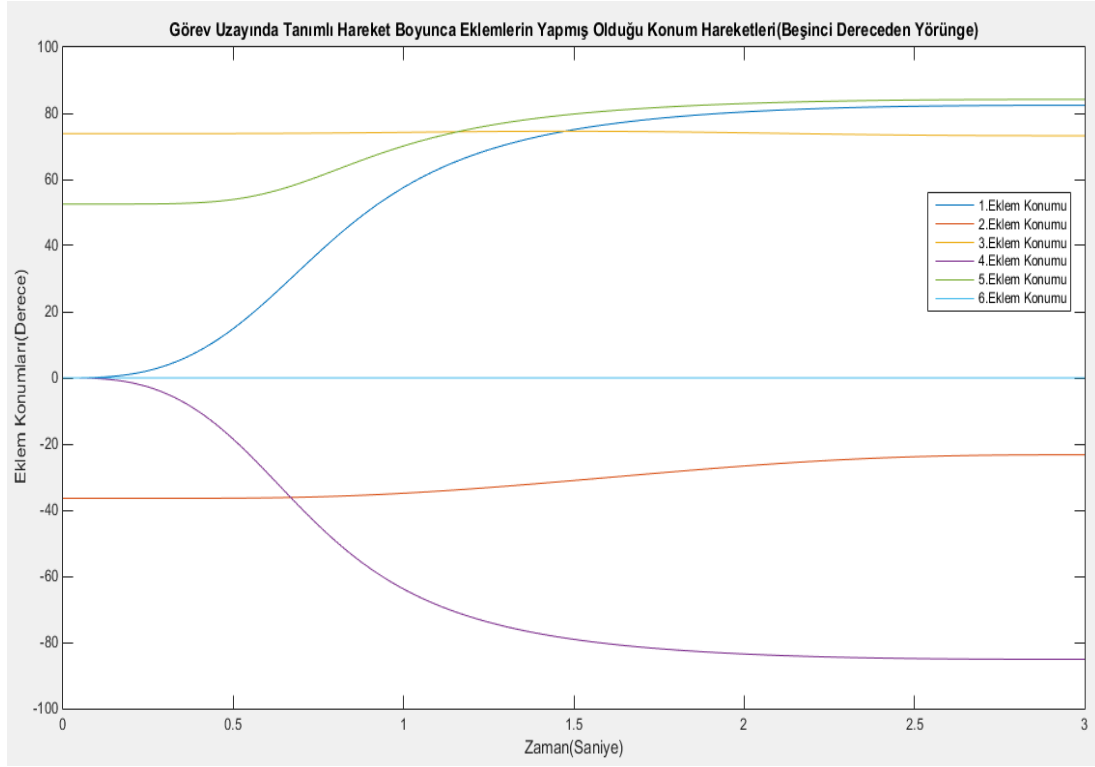
Diğer bir ters dinamik kontrol algoritması olan kararlı ivme kontrolüne ait referans girişler ve sonuçlar aşağıda yer alan şekillerde görülmektedir.



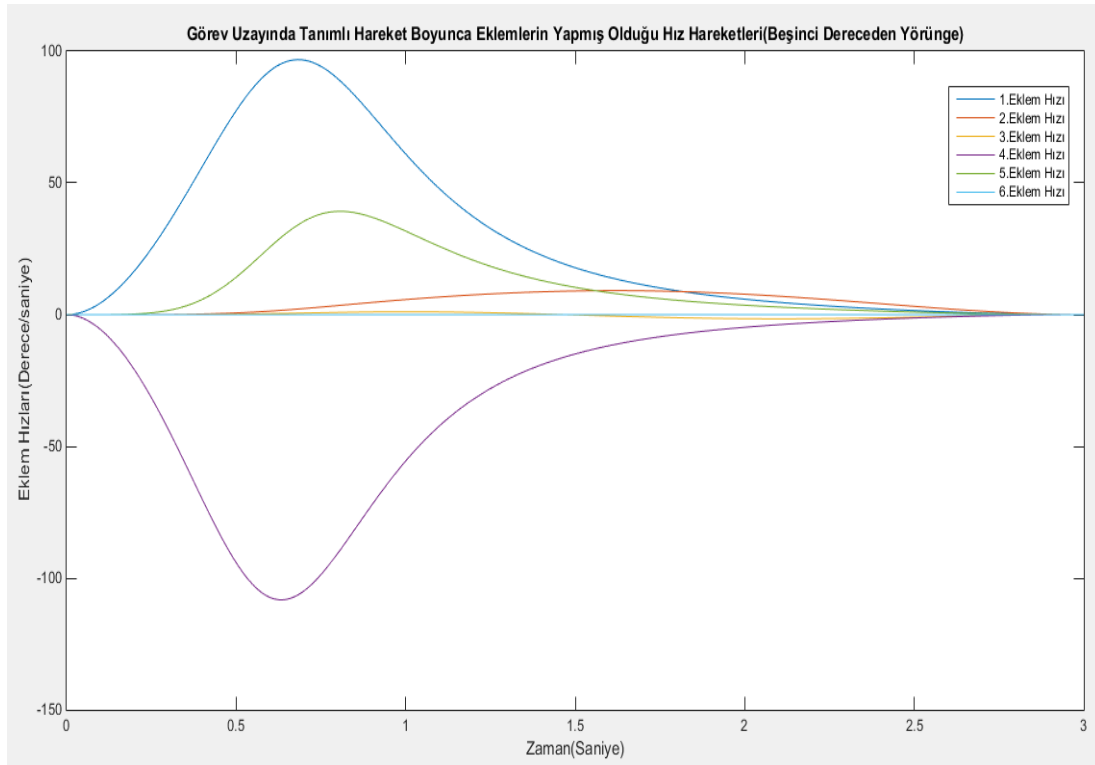
Şekil 3.11: Robot uç noktası referans kartezyen konumu (beşinci dereceden yörünge).



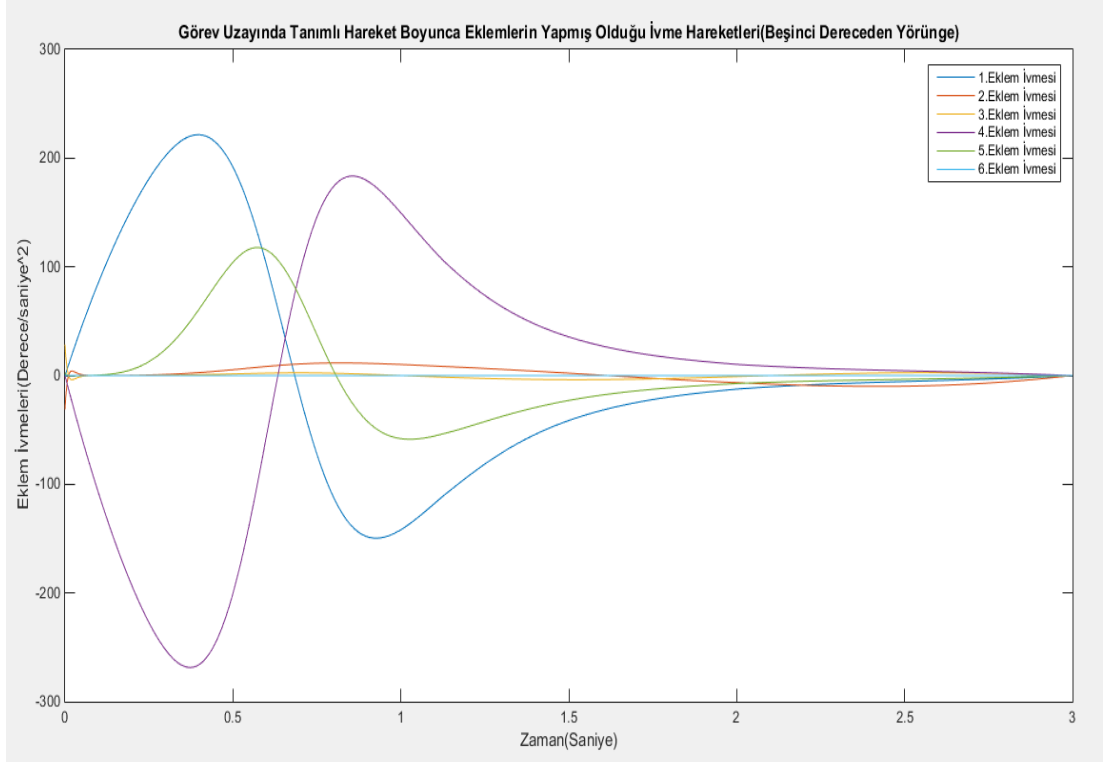




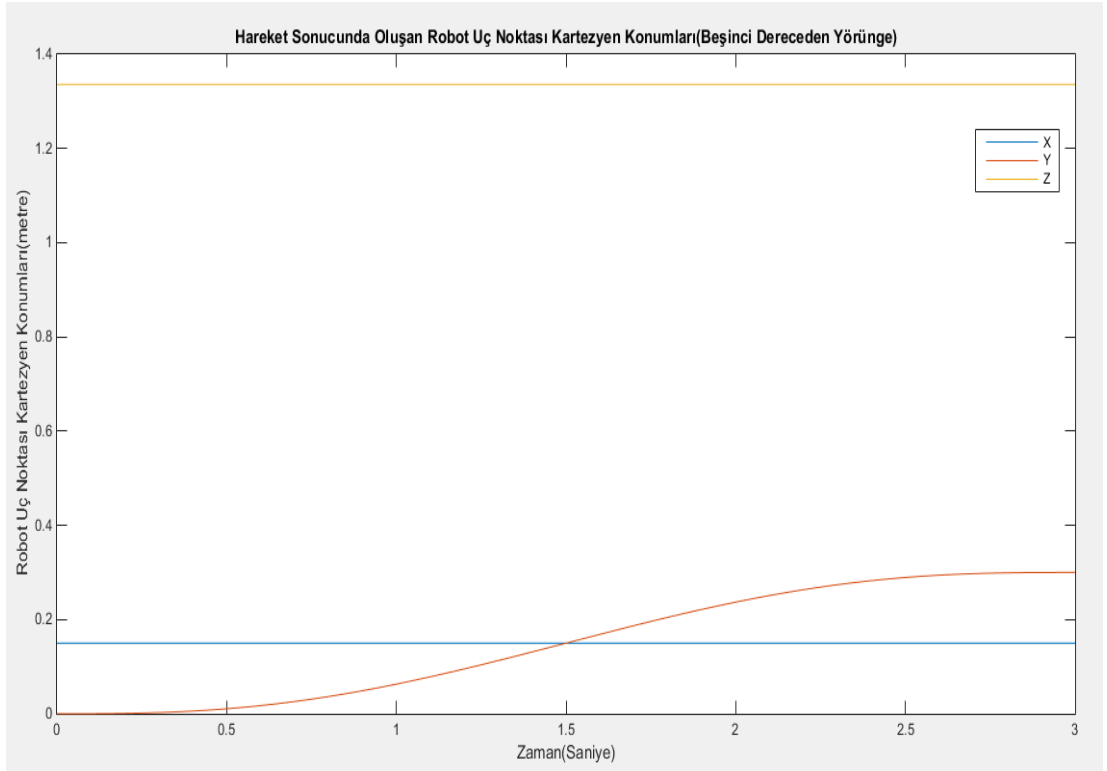
Şekil 3.16: Eklem konumları(beşinci dereceden yörünge).



Şekil 3.17: Eklem hızları(beşinci dereceden yörünge).



Şekil 3.18: Eklem ivmeleri(beşinci dereceden yörünge).



Şekil 3.19: Robot uç noktası kartezyen konumları(beşinci dereceden yörünge).

3.2 Sensörsüz Kuvvet Kontrolü

Robotlar montaj, çapak alma ve taşlama gibi görevleri icra ederken dış ortamla temasta bulunmaktadır. Dış ortamla olan bu temas nedeni ile robotlara sadece konum kontrolü uygulamak yetersiz ve etkisiz kalmaktadır. Bu nedenle, endüstriyel seri robotlara hem konum hem de kuvvet kontrolü uygulamak gerekmektedir. Kuvvet kontrolü sırasında, robot ile dış ortam arasında oluşan tepki kuvvetinin bilinmesi gerekmektedir. Genellikle tepki kuvvet vektörüne ait 3 tane kuvvet ve 3 tane moment değerini veren kuvvet-moment sensörleri kullanılmaktadır. Fakat kuvvet ve moment değerlerini ölçmek için kullanılan bu sensör, sıcaklık gibi fiziksel değişkenlerden etkilene bilmektedir. Bunun yanında kuvvet-moment sensörleri robotun mevcut ağırlığını arttırarak robota ait dinamik yapıyı ve modeli bozabilmektedirler. Kullanılan bu sensörlerin sahip olduğu teknik dezavantajların yanında maliyeti arttırıcı etkileri de bulunmaktadır. Tez sırasında robotun çevresiyle yaptığı hareket nedeni ile oluşan tepki kuvveti, elde edilen dinamik model ve geri besleme tork sinyali kullanılarak elde edilmektedir.

Coriolis ve merkezkaç etkisi nedeni ile oluşan torklar, yerçekimi nedeni ile oluşan torklar, sürtünme torkları, robotun çevresi ile teması sonucu eklemlerde oluşan tepki torkları ve yayın hareketi sonucu oluşan torkun toplamı bozucu tork değerini vermektedir.

$$\tau_{\text{bozucu}} = C(q, \dot{q})\dot{q} + g(q) + \tau_s(q) + \tau_f(q, \dot{q}) + \tau_{\text{tepki}} \quad (3.49)$$

Denklem 3.49 ile verilmekte olan bozucu tork denkleminde yer alan τ_{tepki} bileşeni, robot ile çevre temas halinde iken robot eklemlerinde oluşan tepki torkunu vermektedir.

Robot eklemlerinde oluşan bozucu tork, aynı zamanda enkoderlerden gelen geri besleme tork değeri ile eylemsizlik matrisinin eklem ivmeleriyle çarpımı sonucu oluşan tork değeri arasındaki farka eşittir.

$$\tau_{\text{bozucu}} = \tau_{\text{geri besleme}} - D(q)\ddot{q} \quad (3.50)$$

Denklem 3.50’de yer alan bozucu tork ve sensörlerden gelen geri besleme tork değerleri 6×1 vektördürler. Eğer denklem 3.49 ve denklem 3.50’nin sağ tarafları birbirleri ile eşitlenirse, robotun dış ortamla teması sonucu eklemlerde oluşan tepki torkları elde edilebilmektedir.

$$\tau_{\text{tepki}} = \tau_{\text{geri besleme}} - D(q)\ddot{q} - [C(q, \dot{q})\dot{q} + g(q) + \tau_s(q) + \tau_f(q, \dot{q})] \quad (3.51)$$

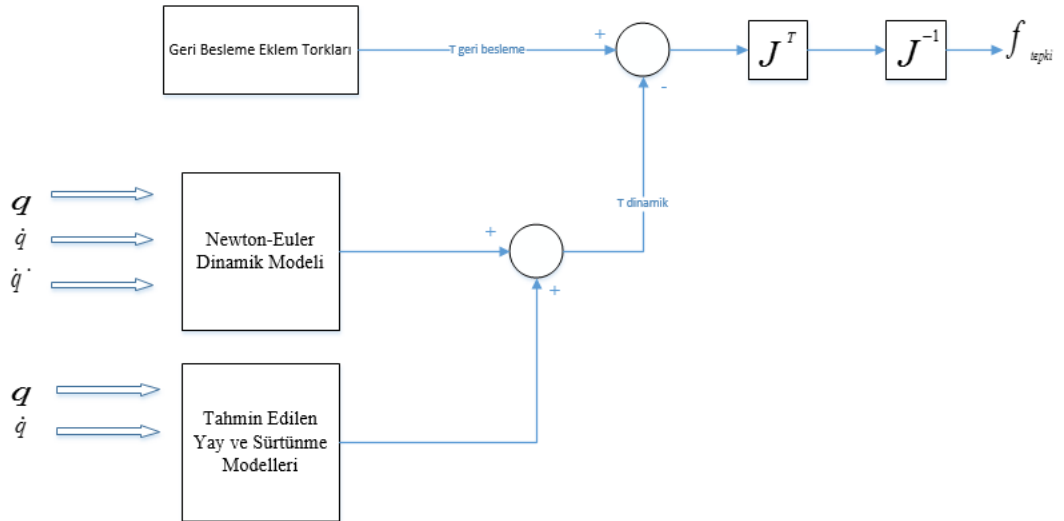
Çevresi ile teması halinde robot uç noktasında oluşan tepki kuvveti ile eklemlerde oluşan tepki torkları arasındaki ilişki denklem 3.52 ile verilmektedir.

$$\tau_{\text{tepki}} = J(q)^T F_{\text{tepki}} \quad (3.52)$$

Denklem 3.52’de yer alan $J(q)$, ileri kinematik model ile elde edilen 6×6 Jakobiyen matrisi temsil etmektedir. Aynı denklemde yer alan 6×1 F_{tepki} vektörünün ilk üç elemanı sırasıyla x, y ve z eksenlerinde oluşan kuvvetleri, son üç elemanı ise aynı eksenlerde oluşan moment değerlerini vermektedir.

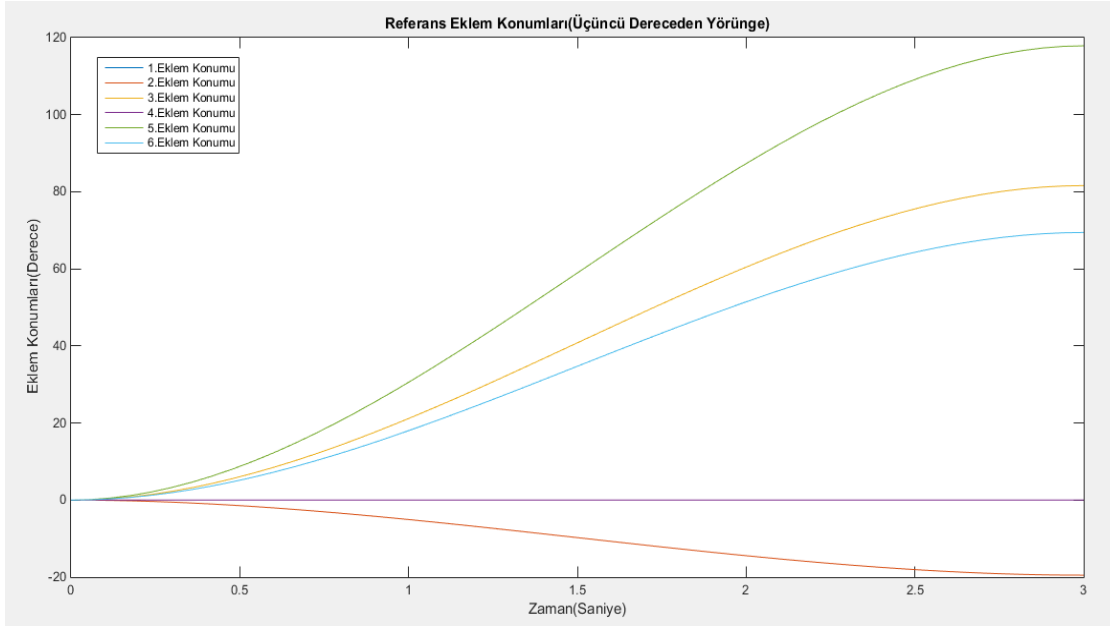
Tepki kuvvetini bulmak için Jakobiyen matrisi diğer tarafa atılıp, tepki kuvveti yalnız bırakılırsa denklem 3.53 oluşmaktadır.

$$F_{\text{tepki}} = J(q)^{-T} \tau_{\text{tepki}} \quad (3.53)$$

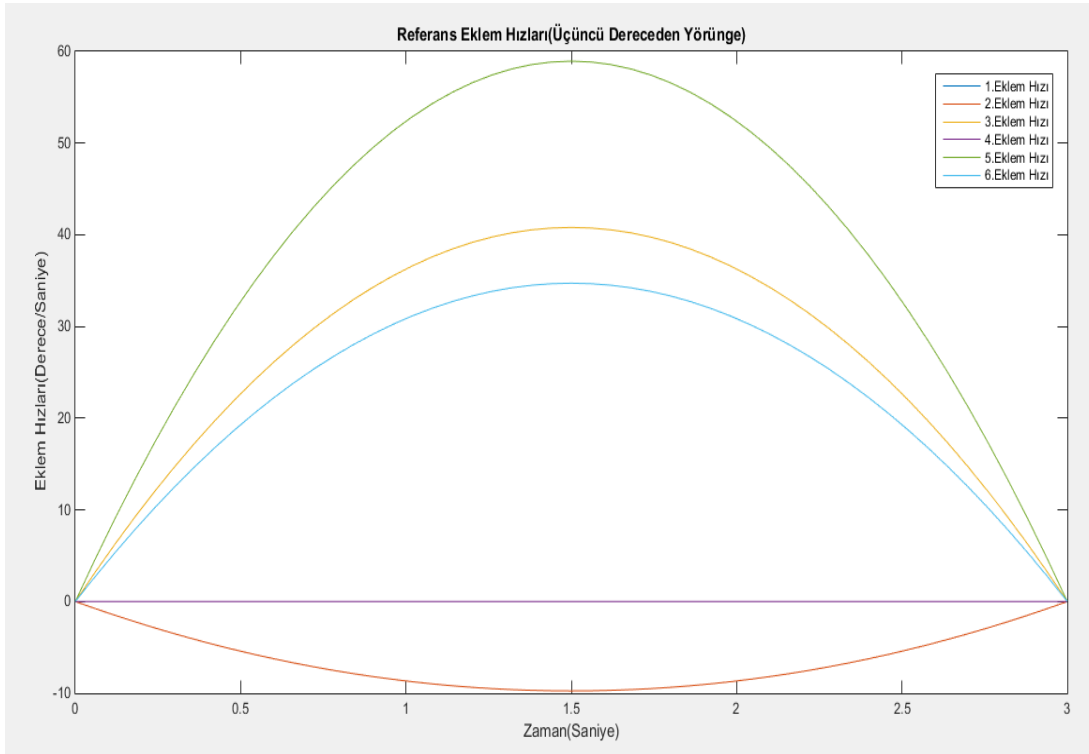


Şekil 3.20: Tepki kuvvetinin sensör kullanılmadan hesaplanması.

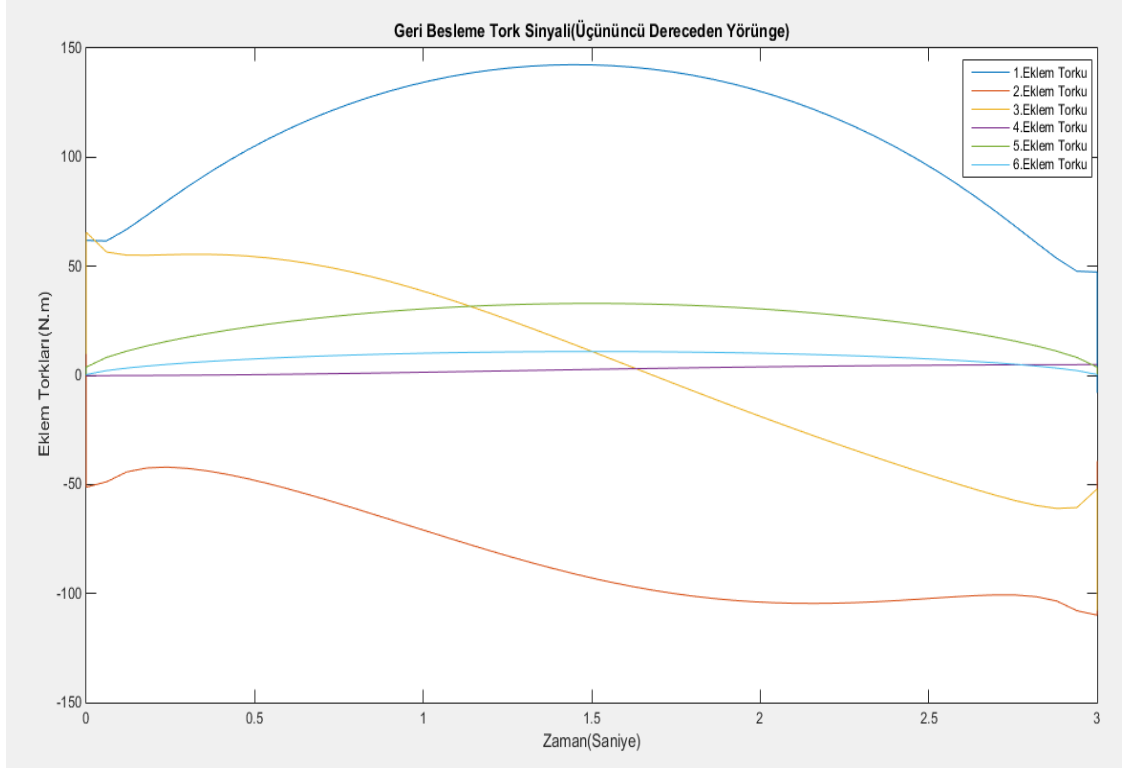
Böylece, denklem 3.49’dan denklem 3.53’e kadar olan denklemler kullanılarak, sensör kullanılmadan tepki kuvveti bulunmaktadır. Robot yazılımı LLI ile her 4 mili saniyede, enkoderlerden robot kontrolcüsüne gönderilen geri besleme eklemler torkları kullanılarak, uç noktada oluşan tepki kuvvet ve moment değerleri elde edilmektedir. Aşağıda yer verilen benzetim sonuçları ile sensör kullanılmadan robot uç noktasında oluşmaktada olan kuvvetlerin ve momentlerin bulunabileceği ispat edilmektedir.



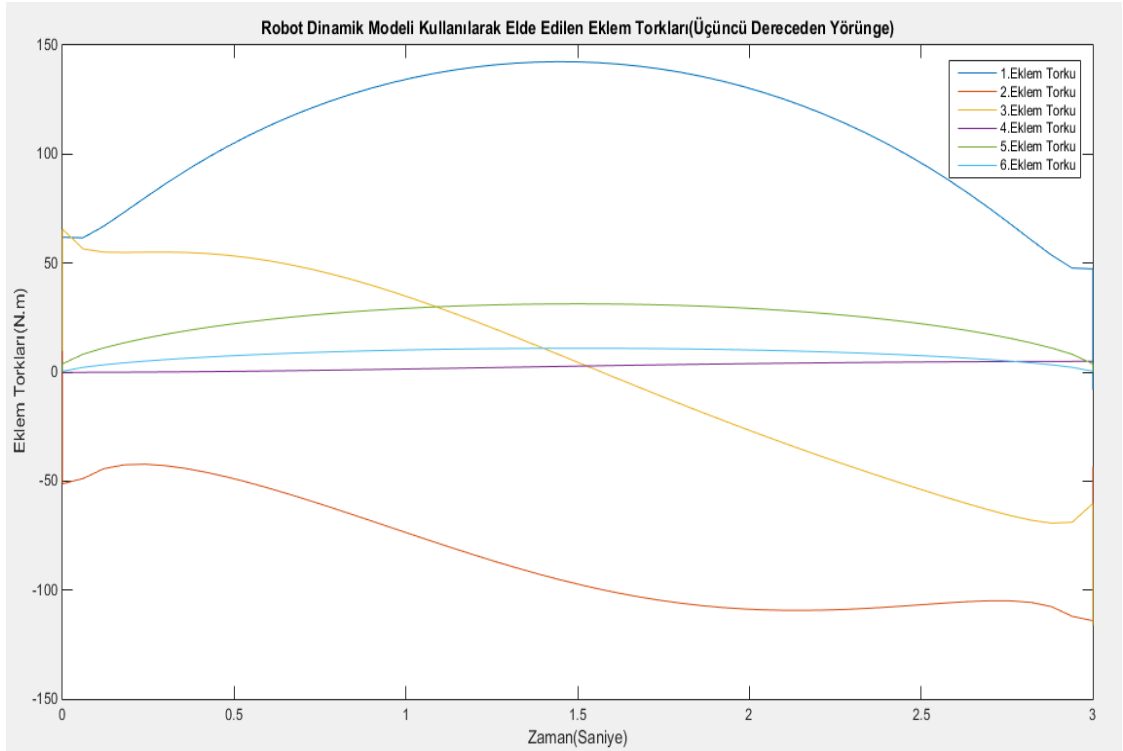
Şekil 3.21: Referans eklem konumları(üçüncü dereceden yörünge).



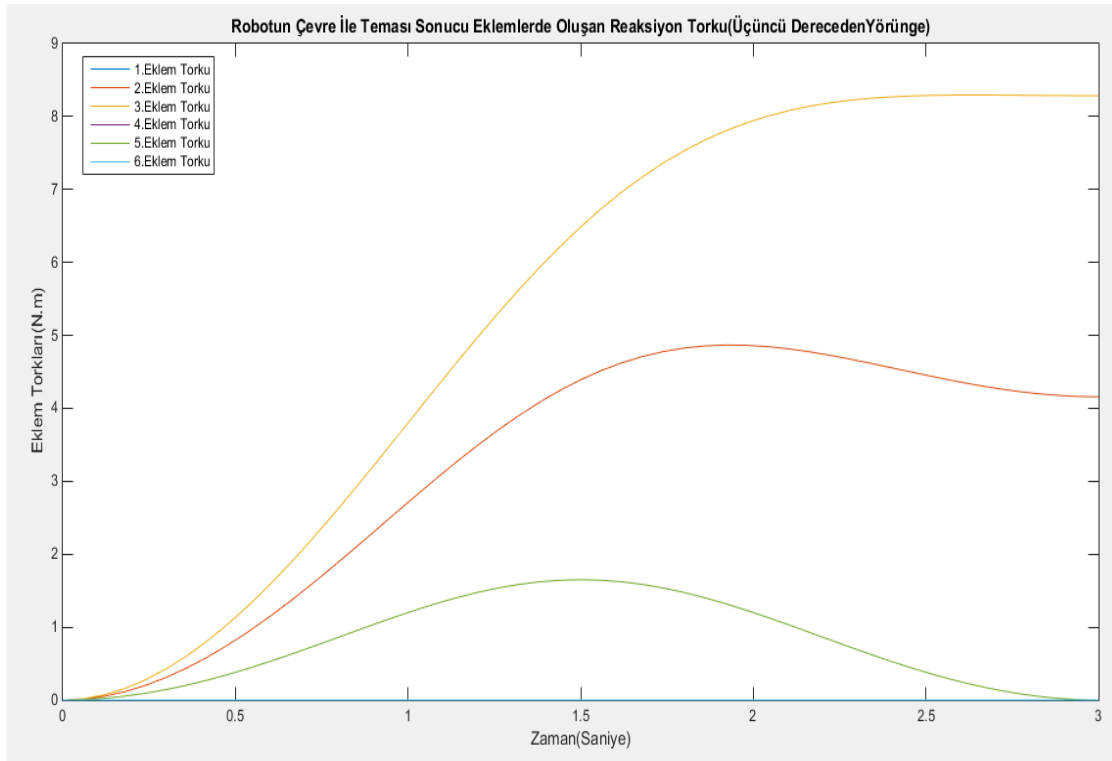
Şekil 3.22: Referans eklem hızları(üçüncü dereceden yörünge).



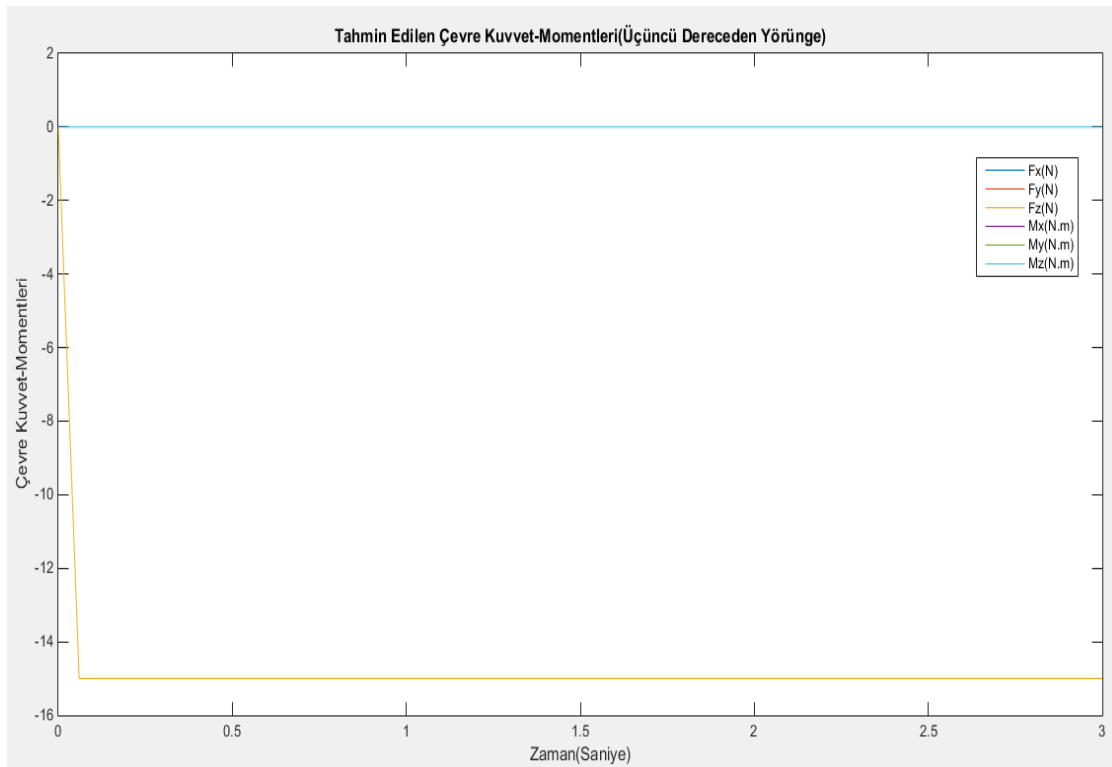
Şekil 3.23: Geri besleme tork sinyali(üçüncü dereceden yörünge).



Şekil 3.24: Robot dinamik modeli kullanılarak elde edilen eklem torkları(üçüncü dereceden yörünge).



Şekil 3.25: Robotun çevre ile teması sonucu eklemlerde oluşan tepki torku(üçüncü dereceden yörünge).



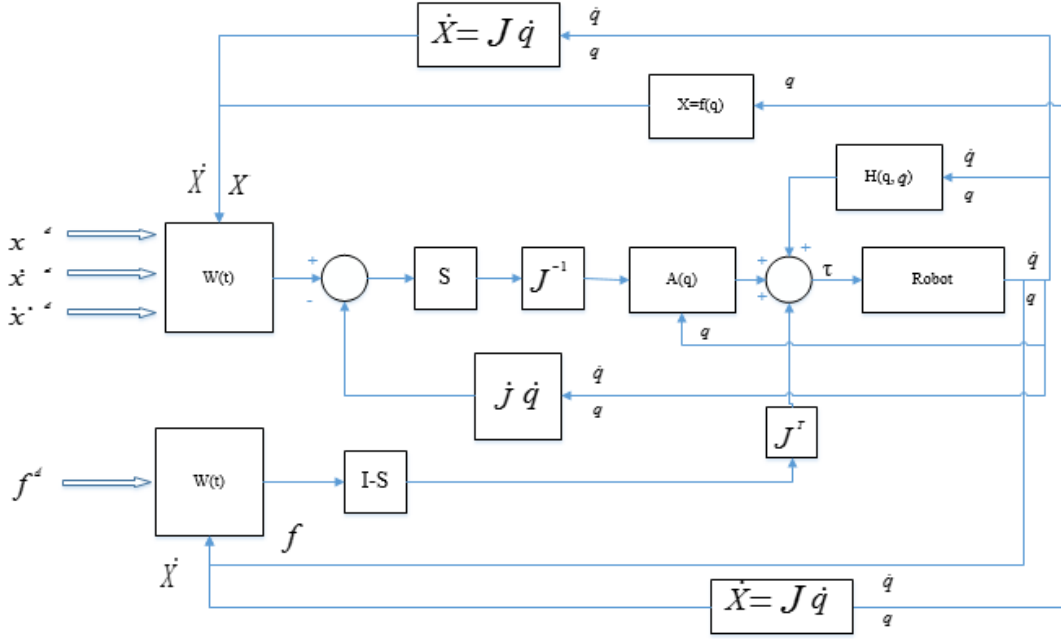
Şekil 3.26: Tahmin edilen çevre kuvvet-momentleri(üçüncü dereceden yörünge).

Elde edilen tepki kuvveti ile birlikte Staubli RX-160 robotunun hem konumunun hem de tepki kuvvetinin kontrol edildiği Hibrid kontrol algoritması gerçekleştirilmektedir. İkinci olarak ise robota, sensörsüz empedans kontrolü uygulanmaktadır.

3.2.1 Hibrid konum/kuvvet kontrolü

Hibrid konum/kuvvet kontrolü ile hem konum hem de robot ile çevresi arasında oluşmakta olan kuvvet kontrol edilebilmektedir. Konum açısından sınırlandırılmış yönlerde kuvvet, kuvvet açısından sınırlandırılmış yönlerde ise konum kontrolü gerçekleştirilmektedir. Hibrid kontrol, iki tane kapalı çevrim kontrol algoritmasından oluşmaktadır. İlk kontrol algoritmasında görev uzayında robot uç noktasına ait konum kontrolü gerçekleştirilmektedir. Konum kontrolü gerçekleştirilirken robot eklemlerinde bulunan sensörler vasıtası ile elde edilen konum ve hız geri besleme sinyalleri kullanılmaktadır. İkinci kapalı çevrim kontrol algoritması ise robot ile çevresi arasında oluşan tepki kuvvet ve momentlerinin kontrol edildiği algoritmadır. Kuvvet kontrol algoritmasında geri besleme sinyali olarak bir önceki bölümde sensör kullanılmadan elde edilen tepki kuvveti kullanılmaktadır. Bu iki kontrol algoritması tarafından oluşturulan kontrol sinyalleri robot eklemlerine gönderilmeden önce toplanmaktadırlar. Staubli RX-160 manipülatörü uç noktası görev uzayında 6 serbestlik derecesine sahiptir. Hibrid konum/kuvvet kontrolü ile bu serbestlik derecelerinden bazısında konum kontrolü gerçekleştirilirken, diğerlerinde ise kuvvet kontrolü gerçekleştirilmektedir. Bu ayrım, 6×6 köşegen seçim matrisi S ile yapılmaktadır. Konumun kontrol edildiği serbestlik derecelerinde seçim matrisi S 1 iken, kuvvet kontrolünün yapıldığı serbestlik derecelerinde 0'dır.

Şekil 3.27'de hibrid konum/kuvvet algoritması gösterilmektedir.



Şekil 3.27: Hibrid konum/kuvvet kontrolü.

Şekil 3.27’de üstte yer alan kontrol döngüsü görev uzayında konum kontrolüdür. $W(t)$ ile verilmekte olan kontrol vektörü, denklem 3.54 ile verilmektedir.

$$W(t) = \ddot{X}^d + K_d(\dot{X}^d - \dot{X}) + K_p(X^d - X) \quad (3.54)$$

Şekilde yer alan konum kontrol algoritması sonucu robota referans olarak gönderilen eklem ivmesi denklem 3.55 ile verilmektedir.

$$\ddot{q} = J^{-1}S[\ddot{X}^d + K_d(\dot{X}^d - \dot{X}) + K_p(X^d - X) - \dot{J}\dot{q}] \quad (3.55)$$

Denklem 3.55’de verilen S matrisi seçim matrisini temsil edip, konumun kontrol edildiği serbestlik dereceleri için ilgili köşegen elemanları(S_j) 1 değerini, kuvvet kontrolünün yapıldığı serbestlik dereceleri için ise 0 değerini almaktadırlar. Denklem 3.55 kullanılarak bulunan referans eklem ivmesi, eylemsizlik matrisi ile çarpılarak robota gönderilmektedir.

Şekil 3.27’de altta yer alan kontrol algoritması ise, kuvvet kontrol algoritmasıdır. Şekilde görülmekte olan $I-S$ seçim matrisi, S seçim matrisinin eşneliğidir. Kuvvet kontrolünün yapıldığı serbestlik dereceleri için $I-S$ köşegen matrisleri 1 değerini almaktadır. Kuvvet kontrolünde yer alan kontrol vektörü $W(t)$ 3.56 ile verilmektedir.

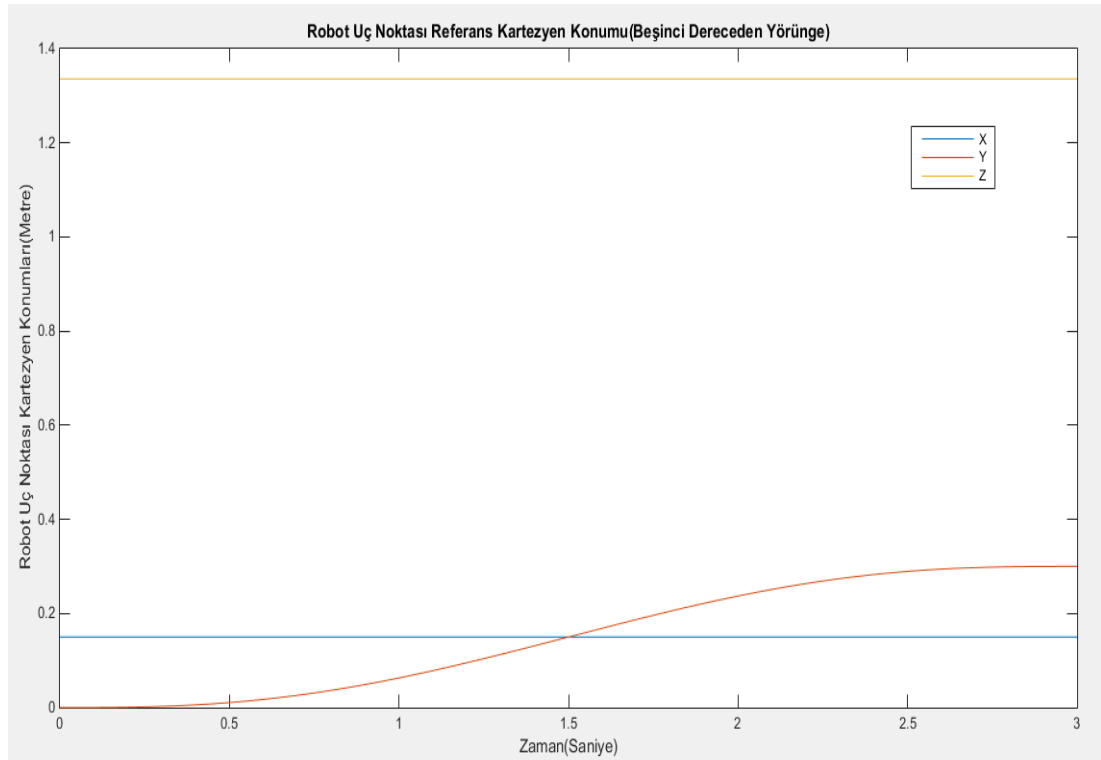
$$W(t) = F_{reak}^d - K_{fd}\dot{X} + K_f(F_{reak}^d - F_{reak.}) + K_{fl} \int_{t_0}^t (F_{reak}^d - F_{reak.}) dt \quad (3.56)$$

Yukarıdaki denklemde robotun çevre ile teması sırasında oluşan tepki kuvveti

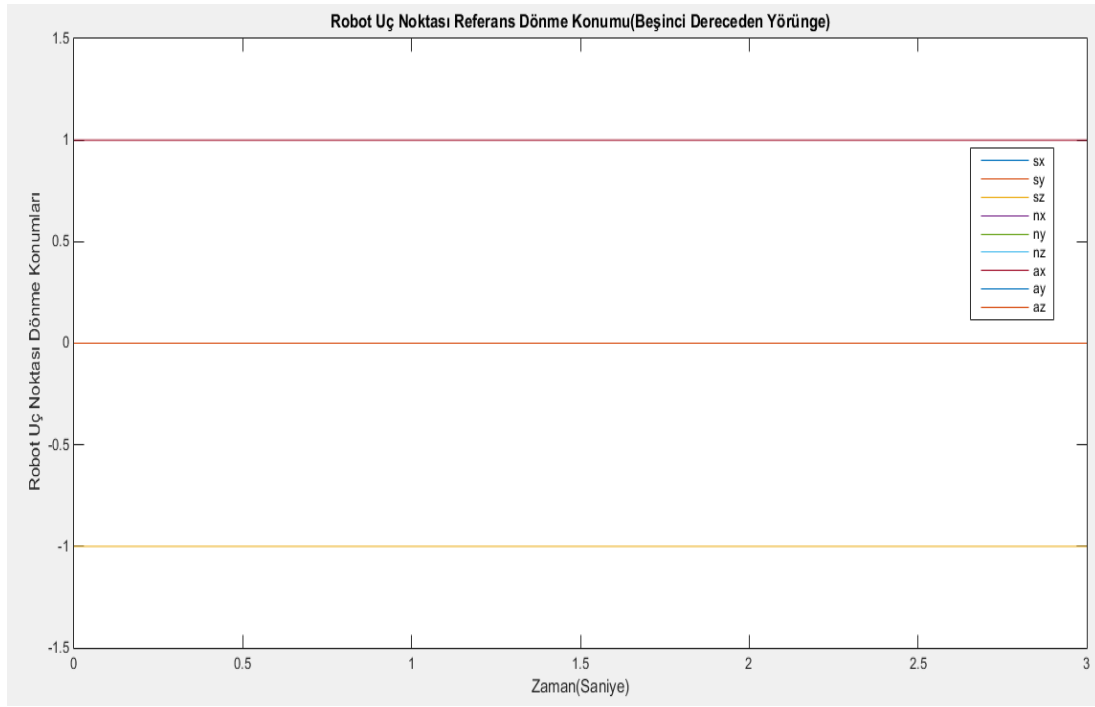
bozucu tork vektörü kullanılarak sensör kullanılmaksızın hesaplandıktan sonra kuvvet kontrol algoritmasına geri besleme sinyali olarak gönderilmektedir. Denklemde tepki kuvveti vektörünün türevi yerine konum vektörünün türevi kullanılmaktadır. Bunun nedeni, tepki kuvvetinin türevinin gürültülü olmasıdır. Son olarak elde edilen $W(t)$ kontrol vektörü seçim matrisi ve Jakobiyen matrisinin transpozu ile çarpılarak robota gönderilmektedir. Kuvvet kontrol algoritması ile robotun çevreye uygulaması istendiği kuvvet ve momentler uygulanabilmektedir.

3.2.1.1 Hibrid konum/kuvvet kontrolü benzetimi

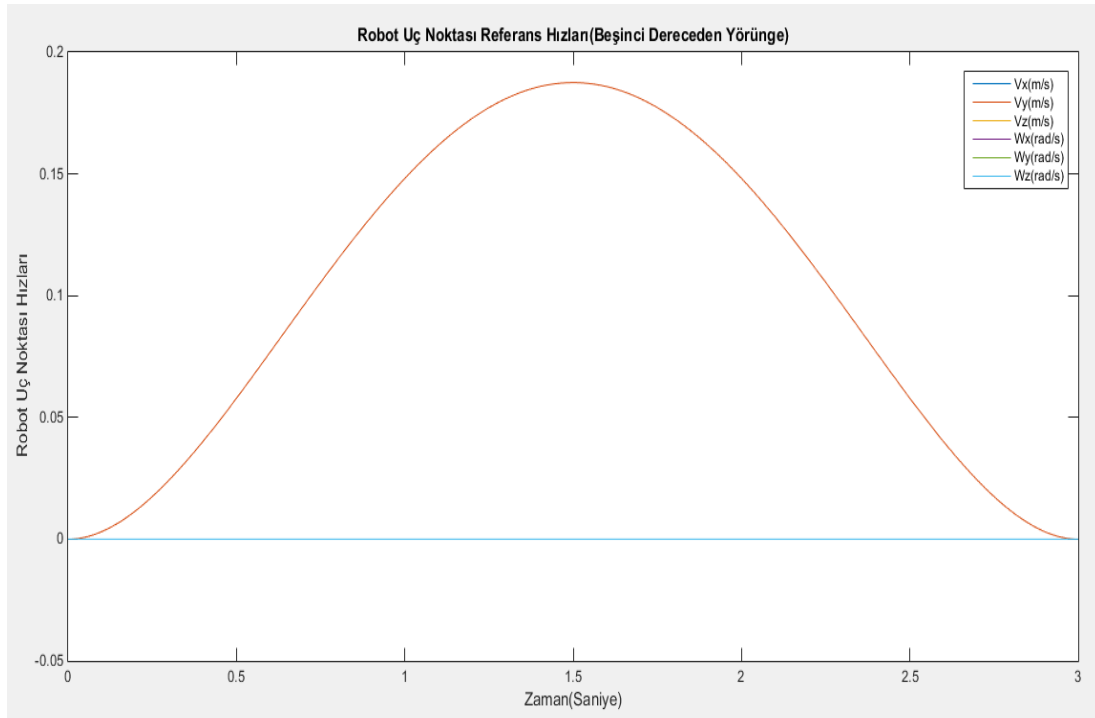
Hibrid konum/kuvvet kontrol algoritması robota uygulanmadan önce Matlab Simulink programı kullanılarak benzetimi gerçekleştirilmektedir. Benzetim sırasında referans olarak beşinci dereceden polinom kullanılmaktadır. Robota ait 6 serbestlik derecesinden 5 tanesinde konum kontrolü, z ekseninde ise -15 N'luk kuvvet çevreye uygulanmaktadır. Benzetim sırasında robot uç noktası ile çevre arasında sabit olarak -15 N'luk bir kuvvet değil, dalgalı bir kuvvet oluştuğu varsayılmaktadır. Böylelikle, Hibrid kontrolün güvenilirliği ve hassaslığı daha iyi bir şekilde ölçülebilmektedir.



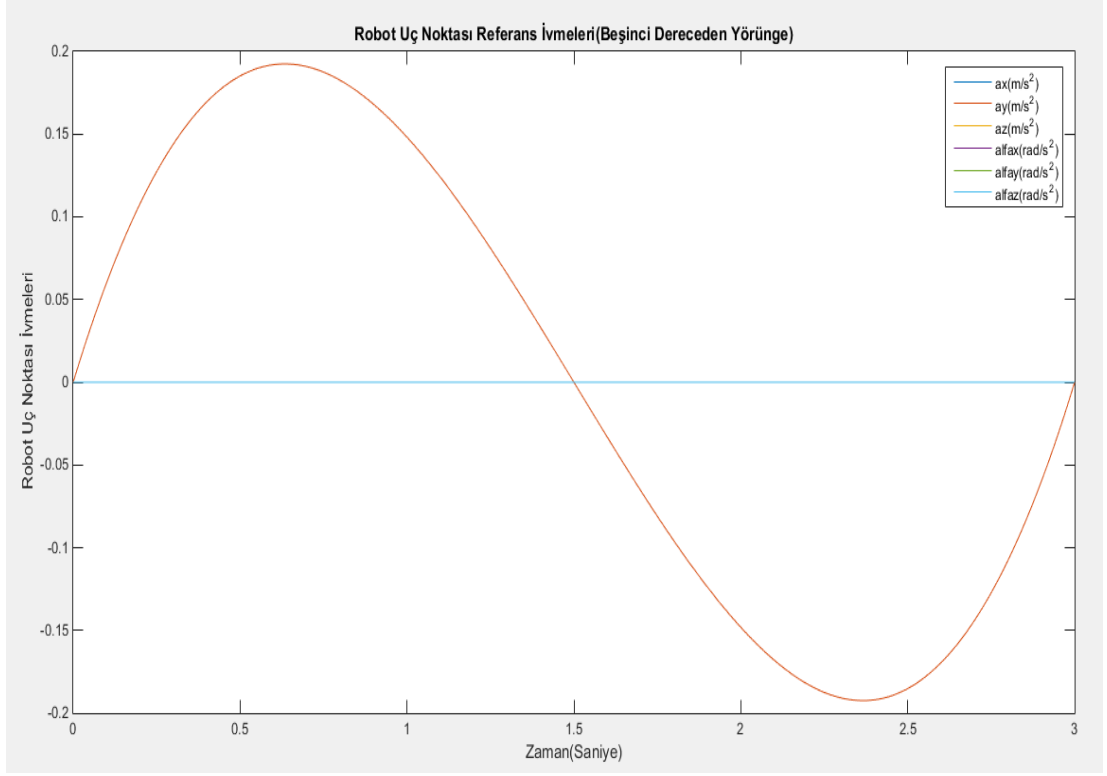
Şekil 3.28: Robot uç noktası referans kartezyen konumu(beşinci dereceden yörünge).



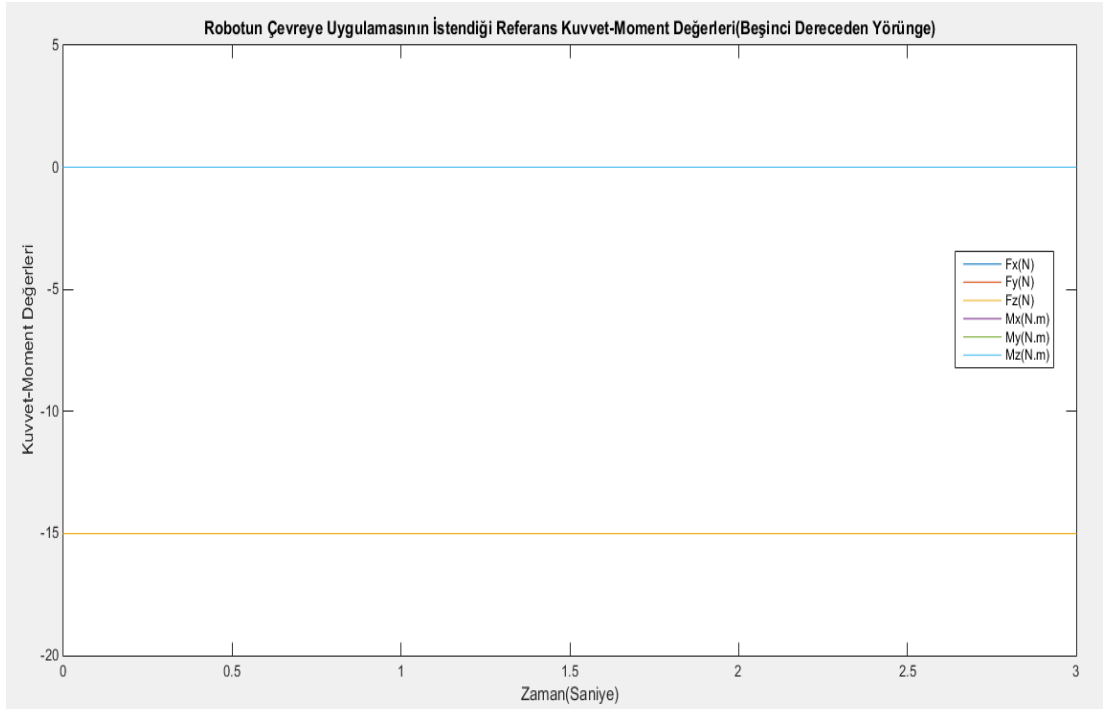
Şekil 3.29: Robot uç noktası referans dönme konumu(beşinci dereceden yörünge).



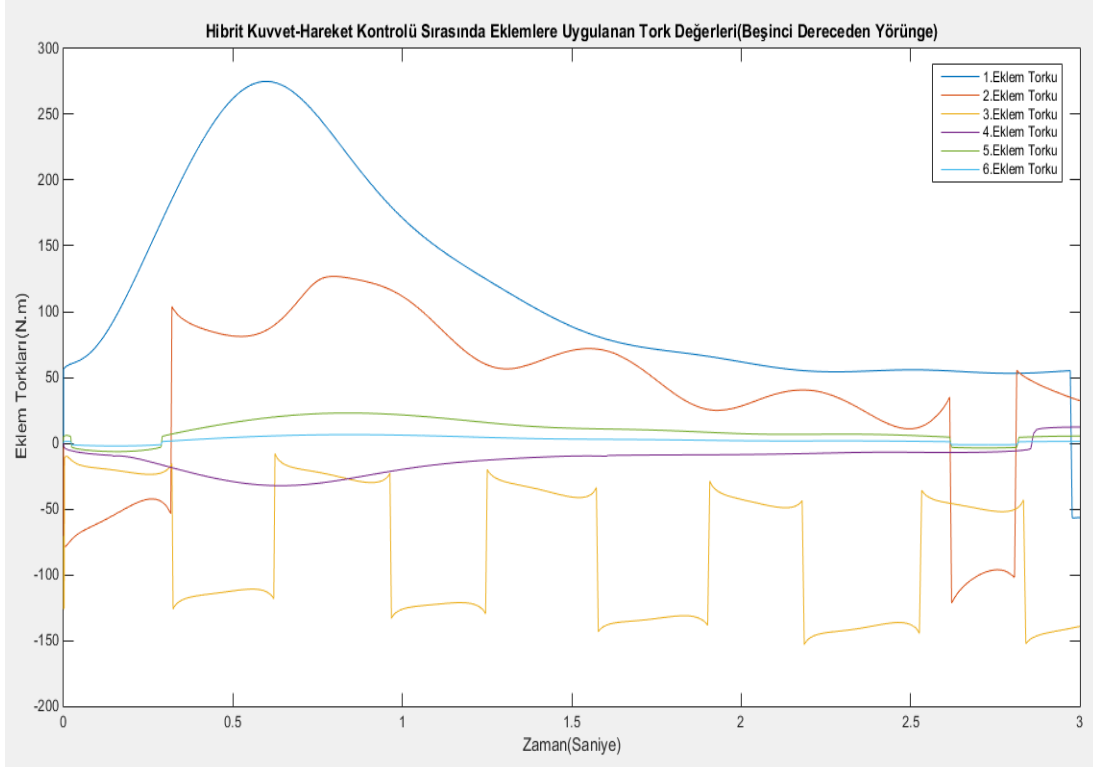
Şekil 3.30: Robot uç noktası referans hızları(beşinci dereceden yörünge).



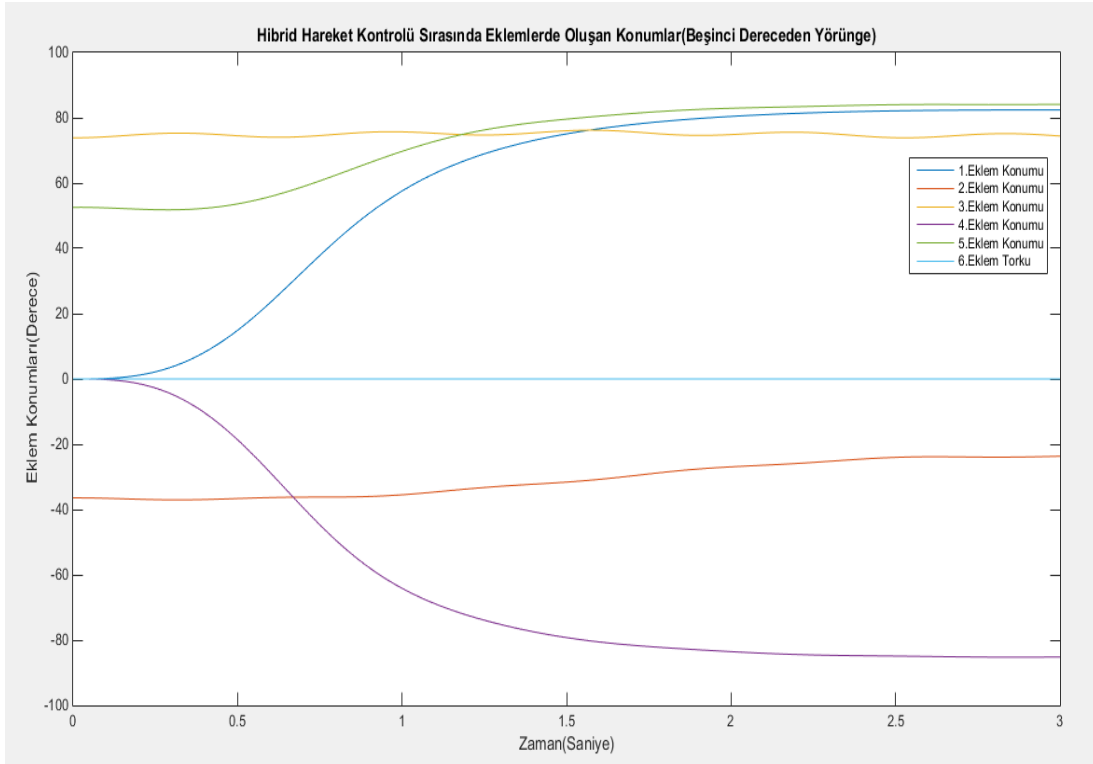
Şekil 3.31: Robot uç noktası referans ivmeleri(beşinci dereceden yörünge).



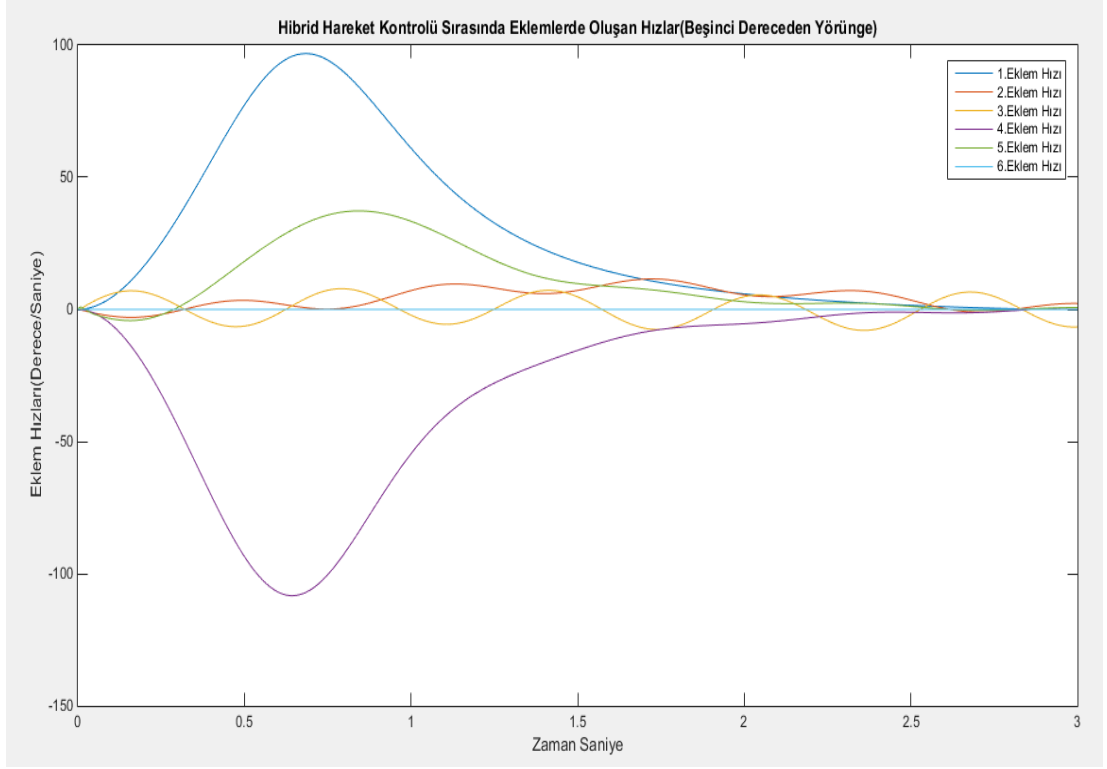
Şekil 3.32: Robotun çevreye uygulamasının istendiği referans kuvvet-moment değerleri(beşinci dereceden yörünge).



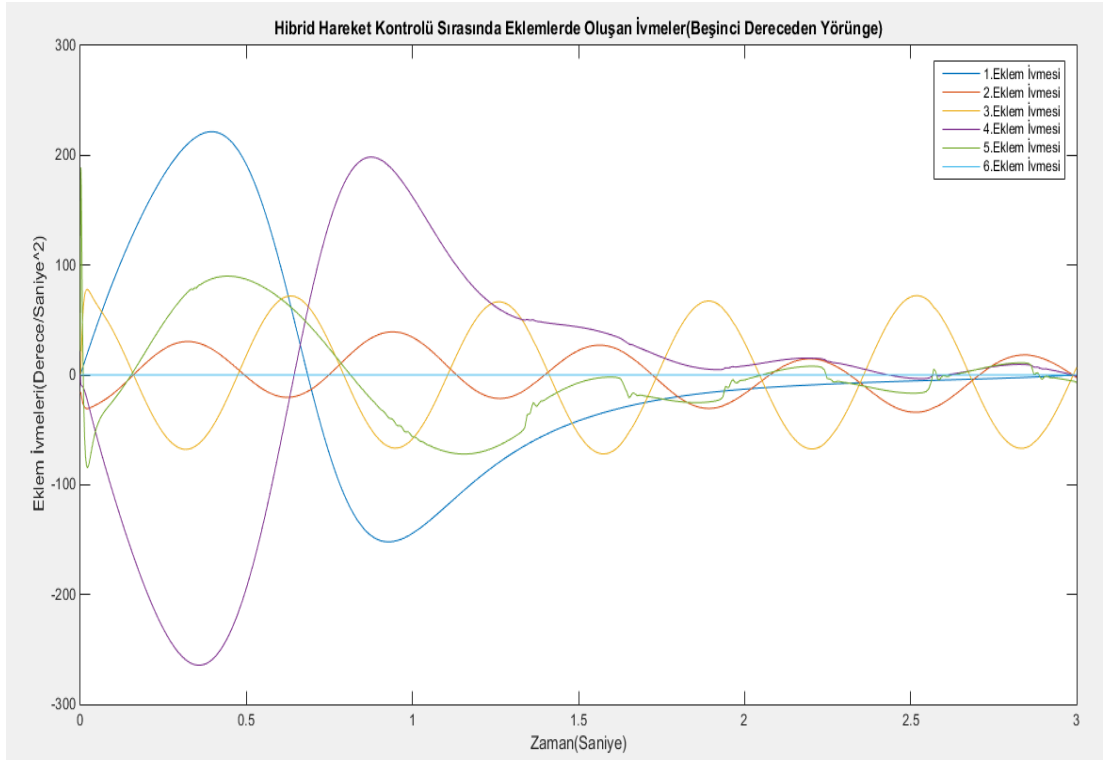
Şekil 3.33: Hibrit kuvvet-hareket kontrolü sırasında eklemlere uygulanan tork değerleri(beşinci dereceden yörünge).



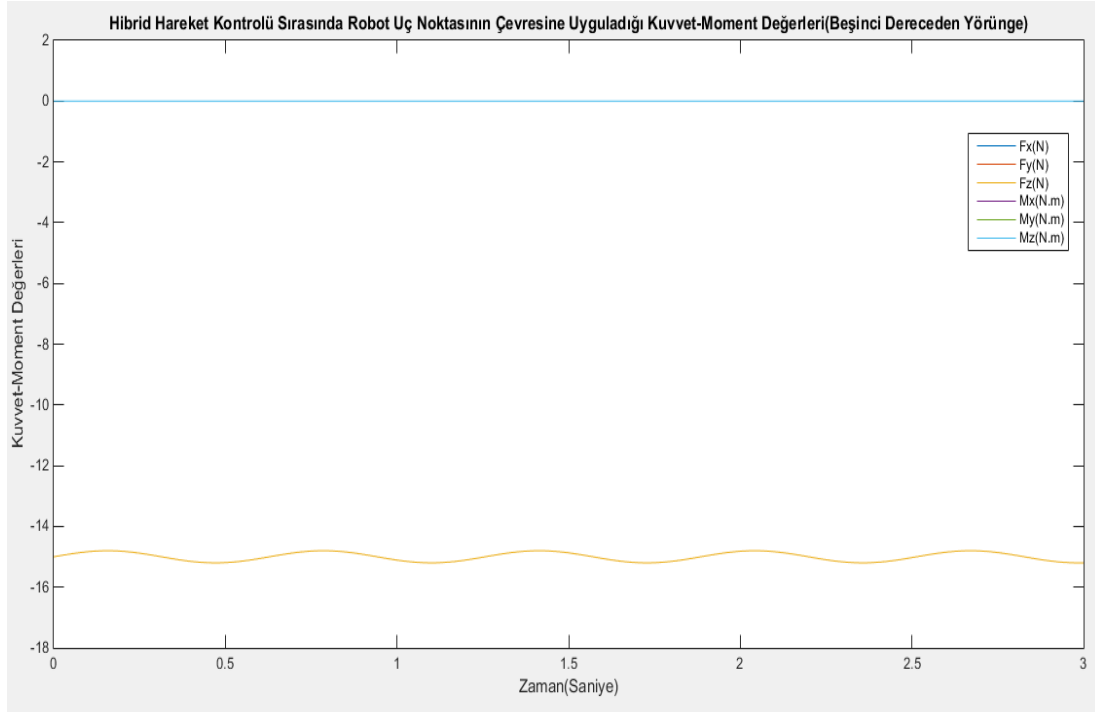
Şekil 3.34: Hibrid hareket kontrolü sırasında eklemlerde oluşan konumlar(beşinci dereceden yörünge).



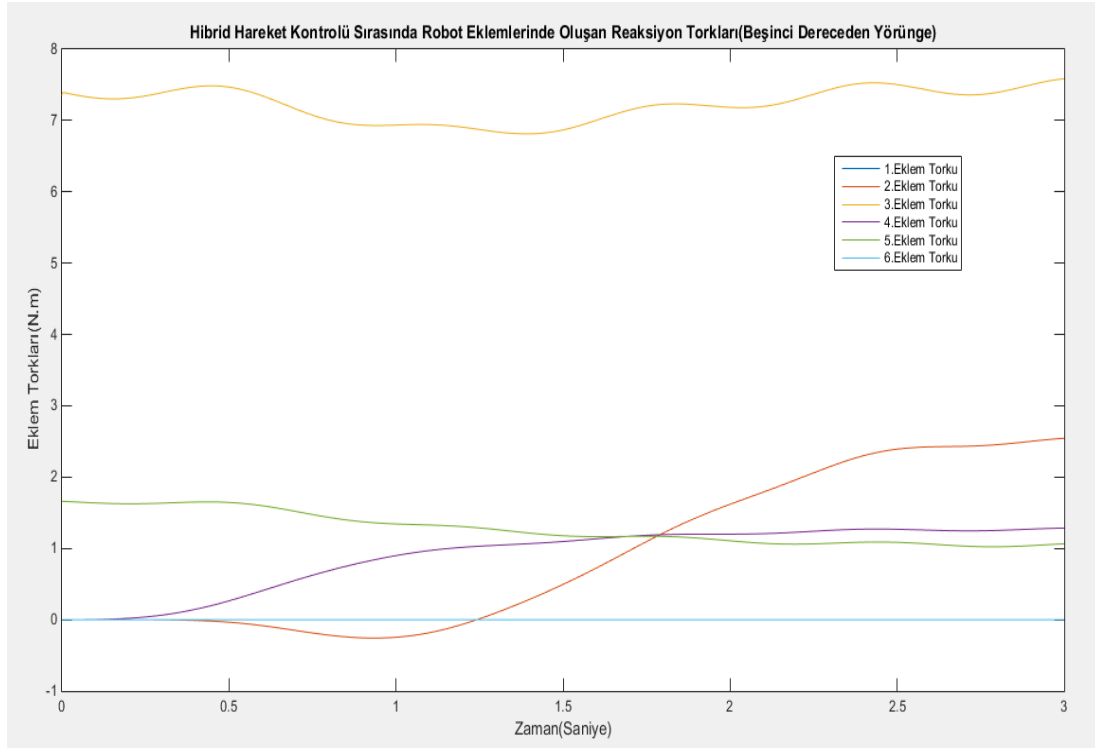
Şekil 3.35: Hibrid Hareket Kontrolü Sırasında Eklemlerde Oluşan Hızlar(Beşinci Dereceden Yörünge).



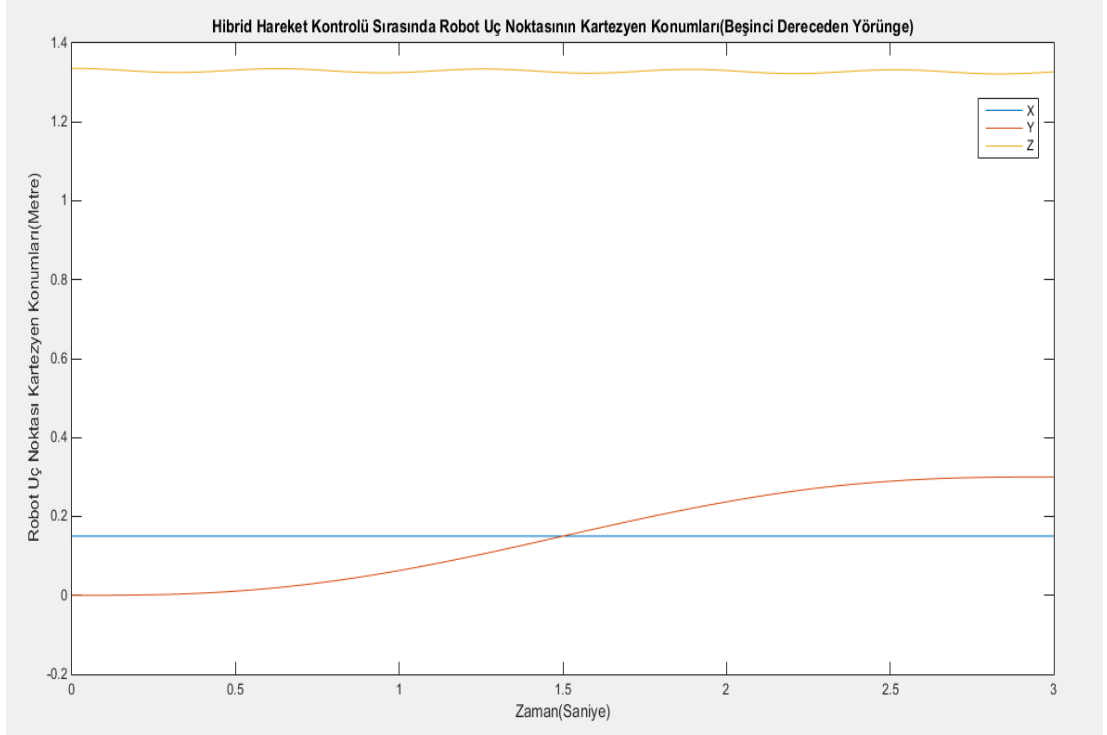
Şekil 3.36: Hibrid hareket kontrolü sırasında eklemlerde oluşun ivmeler(beşinci dereceden yörünge).



Şekil 3.37: Hibrid hareket kontrolü sırasında robot uç noktasının çevresine uyguladığı kuvvet-moment değerleri(beşinci dereceden yörünge).



Şekil 3.38: Hibrid hareket kontrolü sırasında robot eklemlerinde oluşan tepki torkları(beşinci dereceden yörünge).



Şekil 3.39: Hibrid hareket kontrolü sırasında robot uç noktasının kartezyen konumları(beşinci dereceden yörünge).

3.2.2 Empedans kontrolü

Robot ucunun hızı $\dot{X}$ ile robot ve çevre arasında oluşan kuvvet F arasındaki ilişki denklem 3.57 ile verilmektedir.

$$F(s) = Z(s)\dot{X}(s) \quad (3.57)$$

Robot ucunun hızı yerine konumu kullanılırsa denklem 3.57, denklem 3.58'e dönüşmektedir.

$$F(s) = sZ(s)X(s) \quad (3.58)$$

Denklem 3.57 ve 3.58'de kullanılmakta olan $Z(s)$ robotun mekanik empedansını temsil etmektedir. Robot empedansı $Z(s)$ farklı görevler için farklı değerler alabilmektedir. Robot ikinci dereceden kütle-yay-damper sistemi olarak düşünülürse, robota ait transfer fonksiyonu denklem 3.59'da gösterildiği gibi olmaktadır.

$$\frac{F(s)}{X(s)} = \Lambda s^2 + Bs + K \quad (3.59)$$

Denklem 3.59'da yer alan Λ referans eylemsizliği matrisini, B referans sönümleme matrisini ve K referans katılık matrisini temsil etmektedir. Bu üç matrisde robotun

istenilen davranışları sağlaması için seçilebilmektedir. Örneğin robotun çevresi ile temasta olduğu yönlerde Λ matrisinin elemanlarına yüksek değerler verilmektedir. Kinetik enerjinin sönümlenmesinin istendiği yönlerde B matrisine ait elemanlar yüksek değerlerde seçilmektedir. Katılık matrisi K'nın elemanları ise konum kontrolünün gerçekleştirildiği serbestlik derecelerinde yüksek, kuvvet kontrolünün gerçekleştirildiği serbestlik derecelerinde ise düşük değerlerde seçilmektedir.

Empedans kontrolü hem kuvvet-moment sensörü kullanılarak, hem de sensörsüz uygulanabilmektedir. Tez sırasında kullanılan empedans kontrolünde kuvvet-moment sensörü kullanılmamaktadır. Robotun çevreye uyguladığı kuvvet daha önce bahsedildiği gibi tepki torkları kullanılarak elde edilmektedir.

Empedans kontrolü gerçekleştirilirken, ikinci bölümde robota ait dinamik modeli veren denkleme tepki kuvvetinin etkisi de eklenmektedir. Bu durumda oluşan yeni dinamik model denklem 3.60 ile verilmektedir.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_s(q) + \tau_f(q, \dot{q}) + J^T F_{\text{teпки}} = \tau \quad (3.60)$$

Robotun çevreye uyguladığı kuvvet ve momentlerin referans değerleri denklem 3.59'dan yararlanılarak oluşturulabilmektedir. Böylelikle, robot çevresi ile temasta olduğu durumlarda arzu edilen dinamik davranışları gösterebilmektedir.

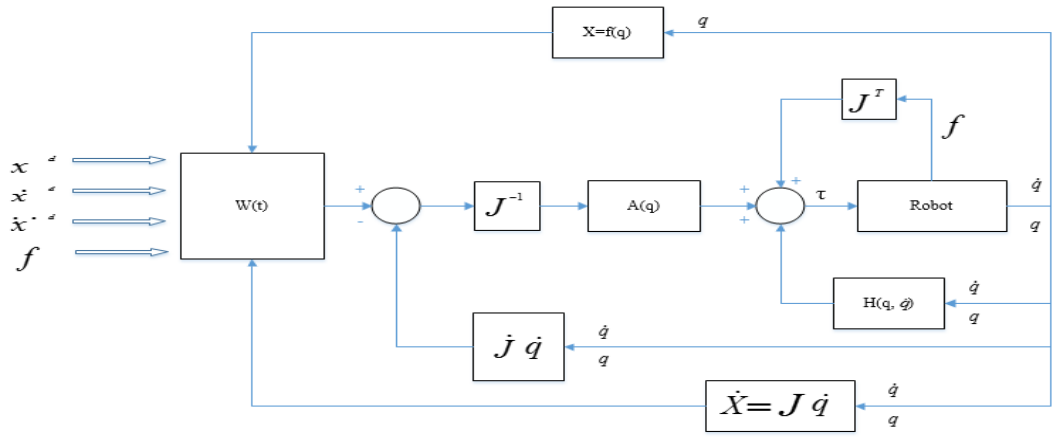
$$F_{\text{reak.}} = \Lambda(\ddot{X}^d - \ddot{X}) + B(\dot{X}^d - \dot{X}) + K(X^d - X) \quad (3.61)$$

Denklem 3.61'de yer alan ivme ifadesi $\ddot{X}$ tek başına bırakılırsa, denklem 3.62 oluşmaktadır.

$$\ddot{X} = \ddot{X}^d + \Lambda^{-1}[B(\dot{X}^d - \dot{X}) + K(X^d - X) - F_{\text{teпки}}] \quad (3.62)$$

Sensörsüz empedans kontrolünü gerçekleştirebilmek için kararlı ivme kontrolü algoritmasından yararlanılmaktadır. Kararlı ivme kontrolü algoritmasında yer alan $W(t)$ kontrol vektörü yerine $\ddot{X}$ ivme vektörü kullanılırsa denklem 3.63 oluşmaktadır.

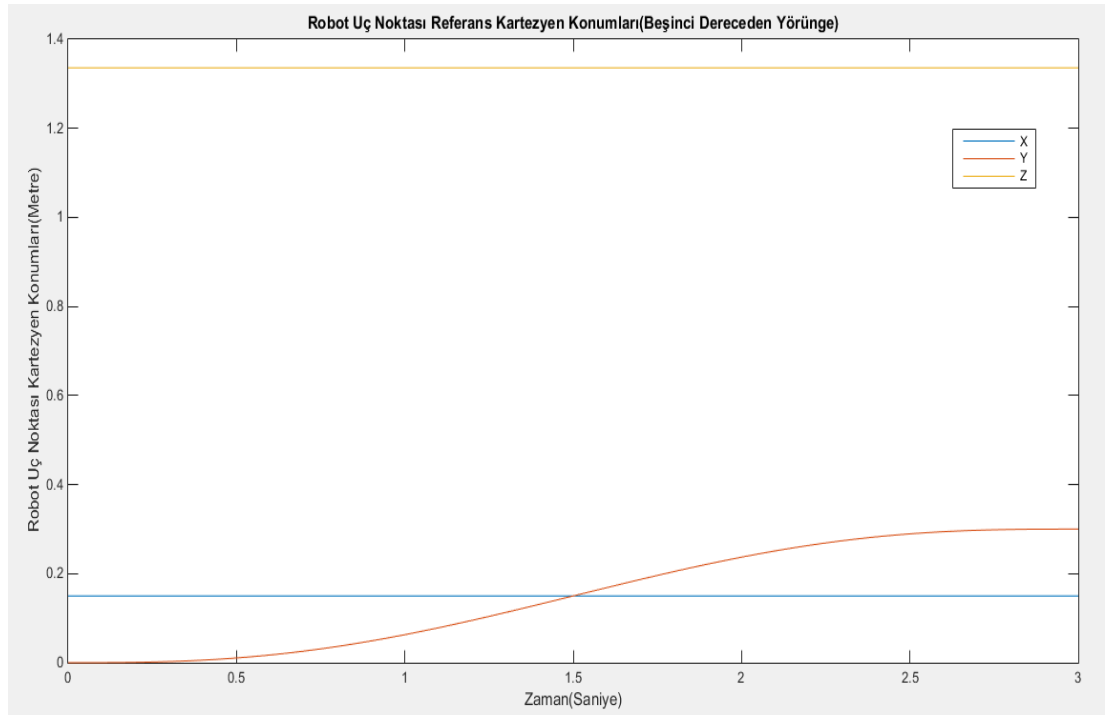
$$\tau = AJ^{-1}[\ddot{X}^d + \Lambda^{-1}(B(\dot{X}^d - \dot{X}) + K(X^d - X) - F) - \dot{J}\dot{q}] + H(q, \dot{q}) + J^T F \quad (3.63)$$



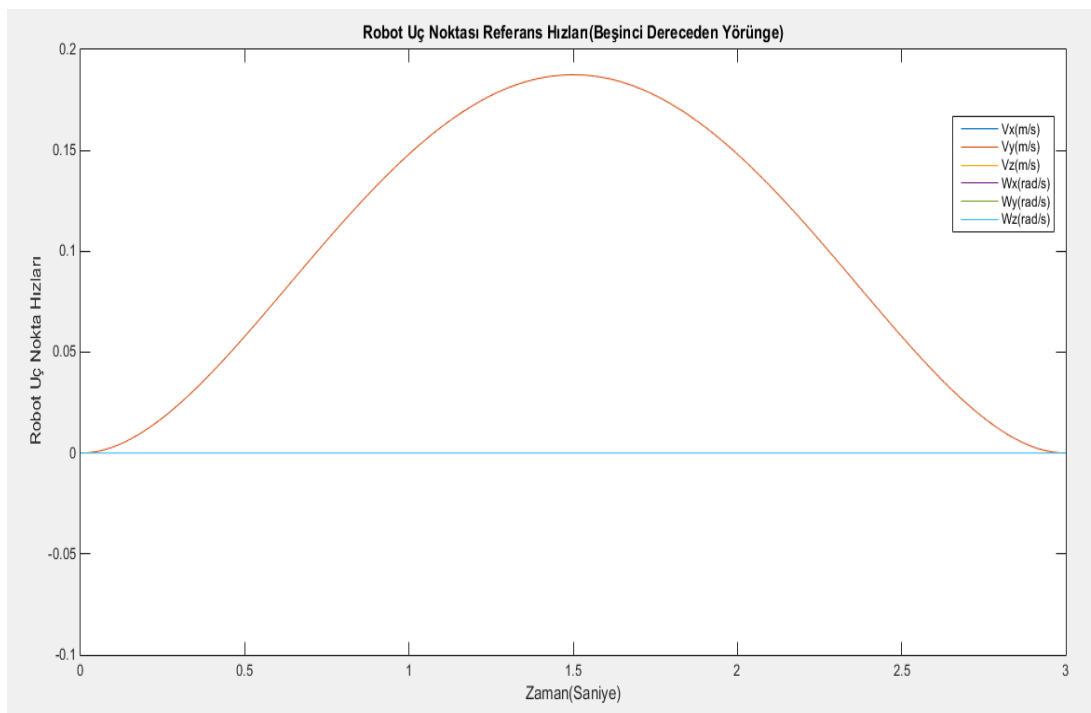
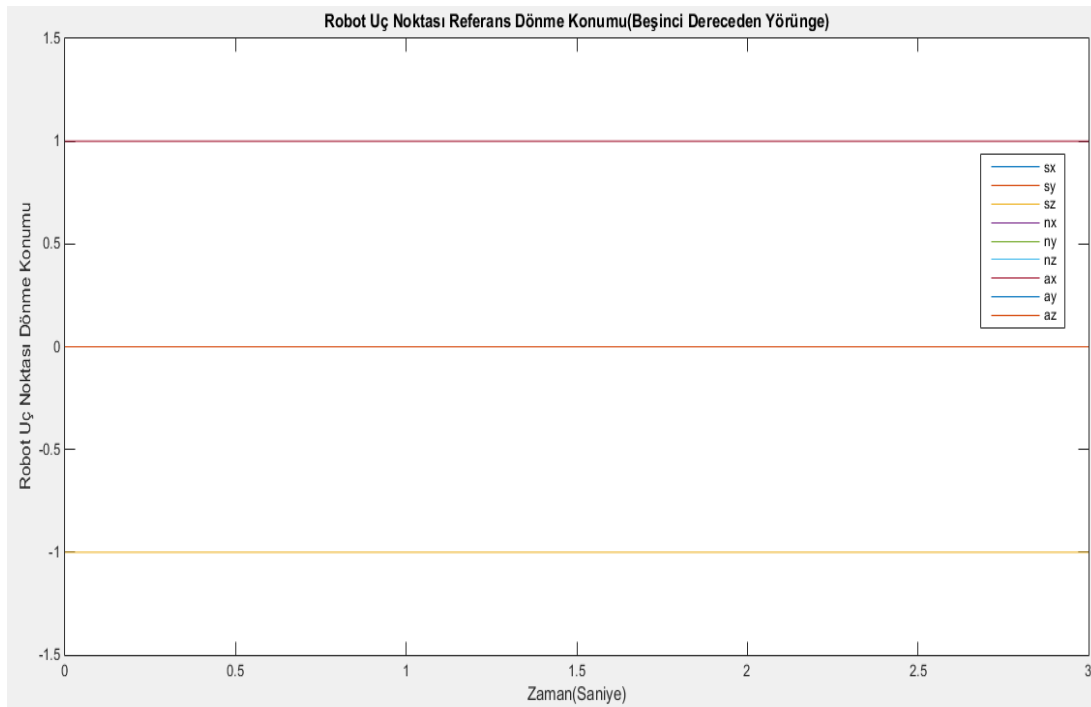
Şekil 3.40: Empedans kontrolü.

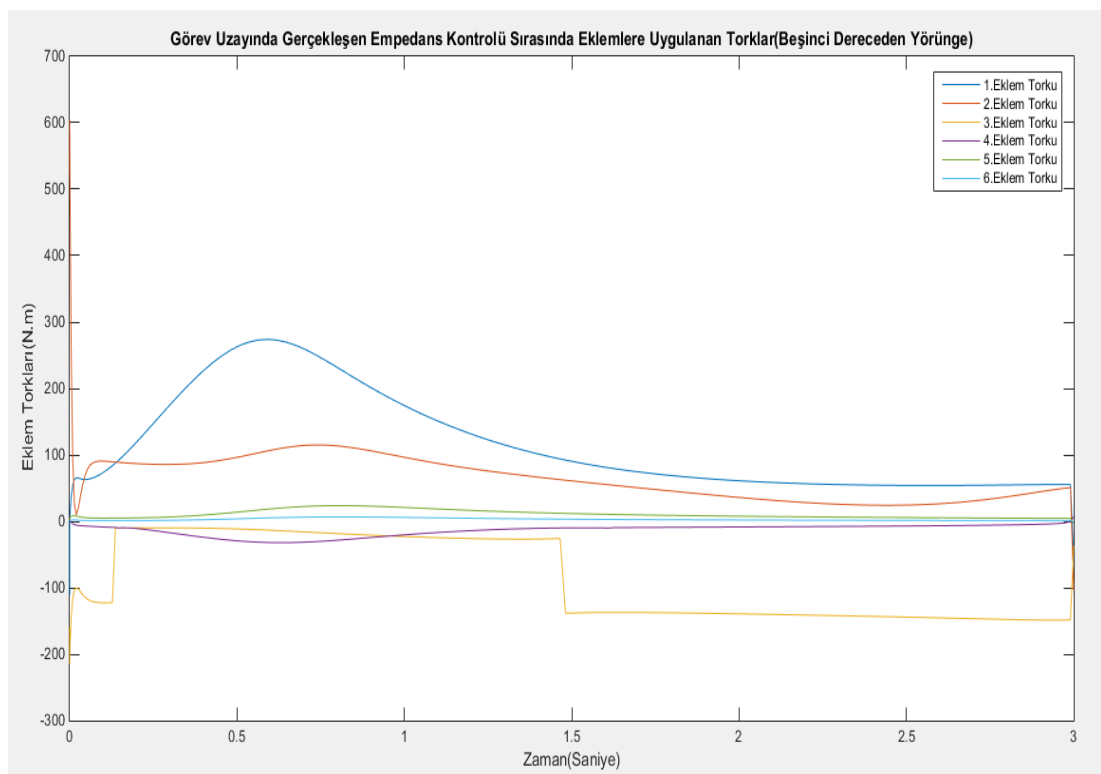
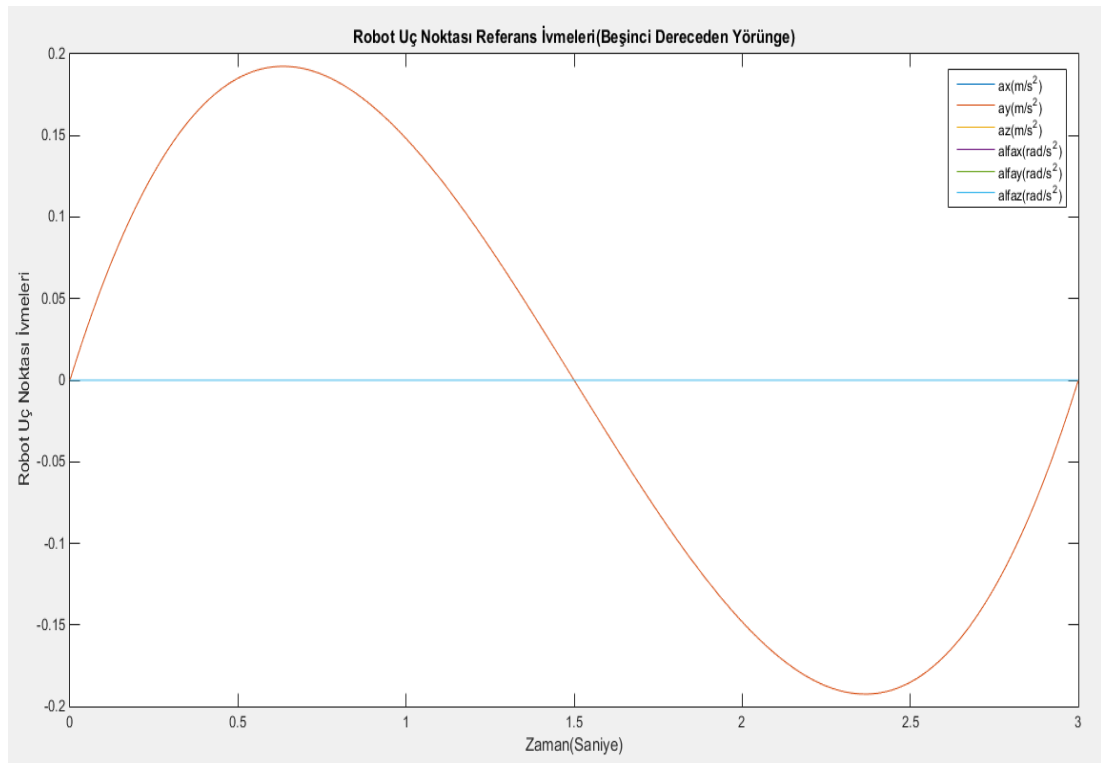
3.2.2.1 Empedans kontrolü benzetimi

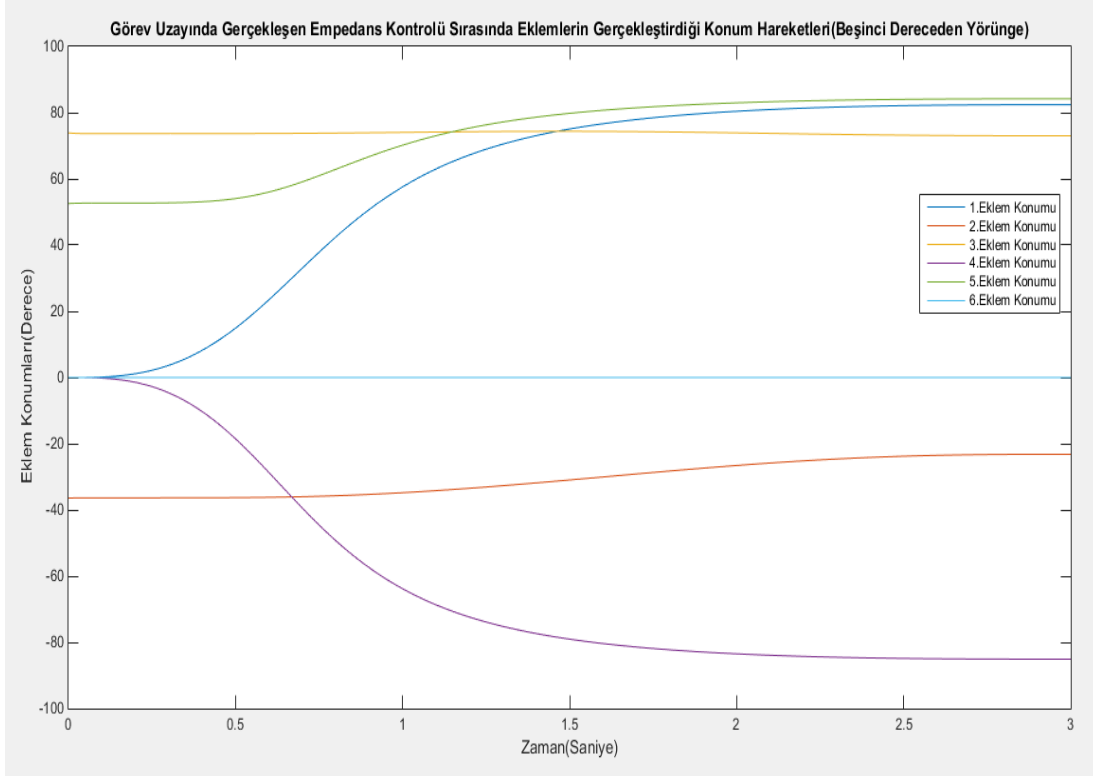
Bir önceki kontrol algoritmalarının benzetiminde olduğu gibi empedans kontrolünün benzetimi de Matlab Simulink programı kullanılarak yapılmaktadır. Benzetim sırasında robotun çevresine z yönünde 15 N'luk bir kuvvet uygulaması istenmektedir. Robot, konum olarak ise y yönünde 0 konumundan artı 0.3 metre konumuna gitmektedir. Hareket sırasında robotun x ve z yönlerinde ise konum değişmemektedir. Empedans kontrolüne ait referans girişler ve sonuçlar aşağıda verilmektedir.



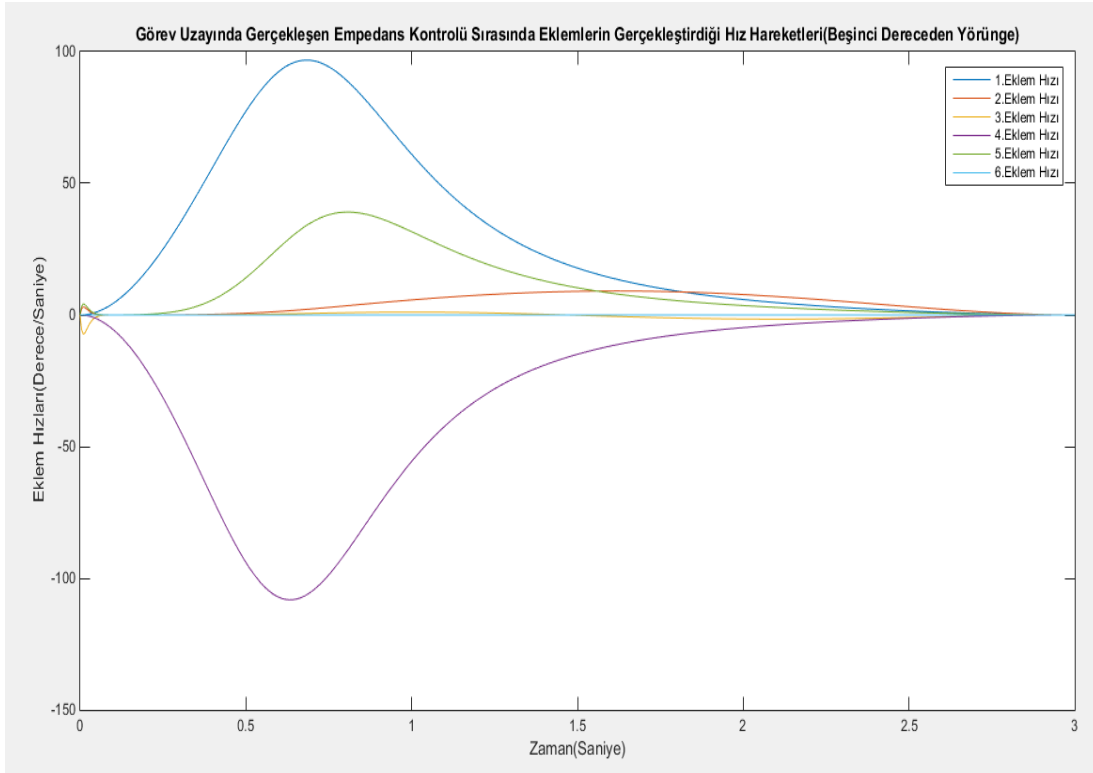
Şekil 3.41: Robot uç noktası referans kartezyen konumu (beşinci dereceden yörünge).



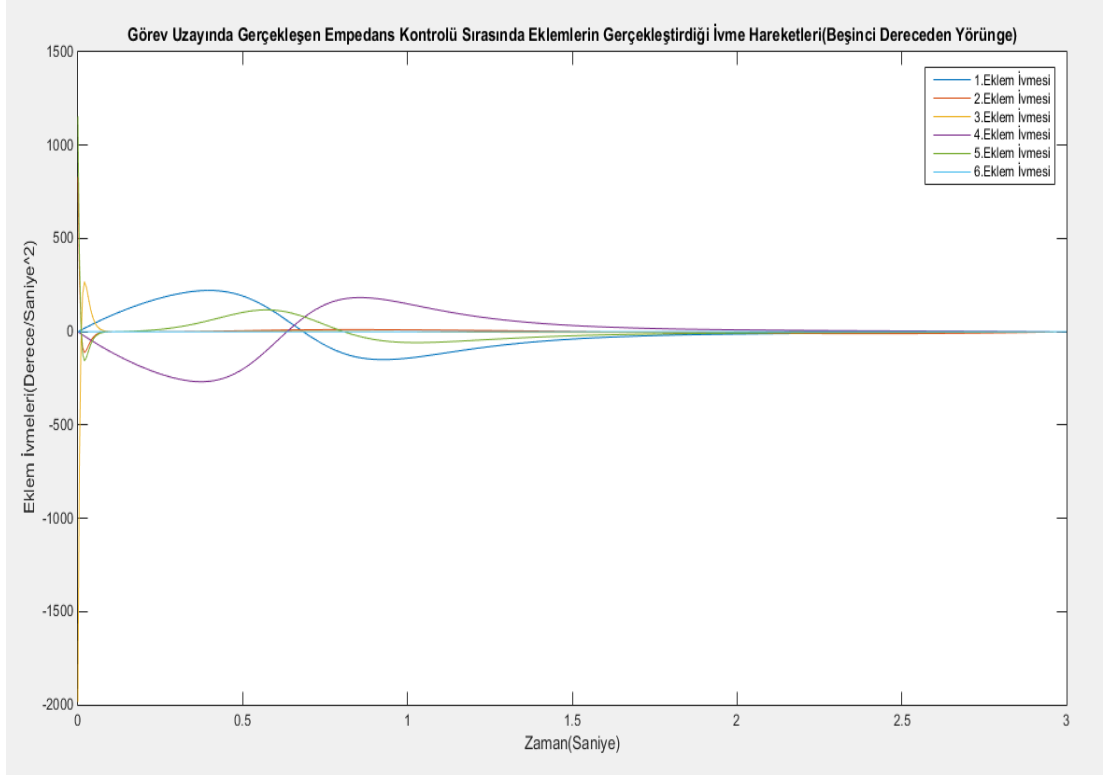




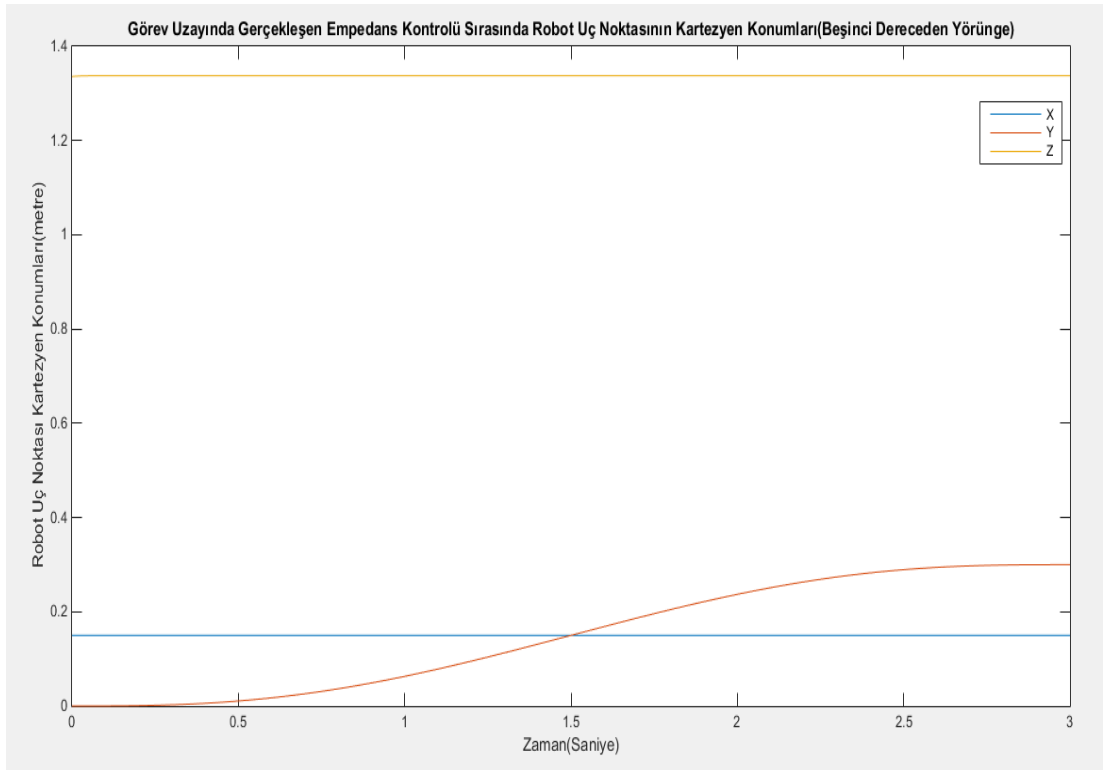
Şekil 3.46: Eklem konumları (beşinci dereceden yörünge).



Şekil 3.47: Eklem hızları (beşinci dereceden yörünge).



Şekil 3.48: Eklem ivmeleri(beşinci dereceden yörünge).



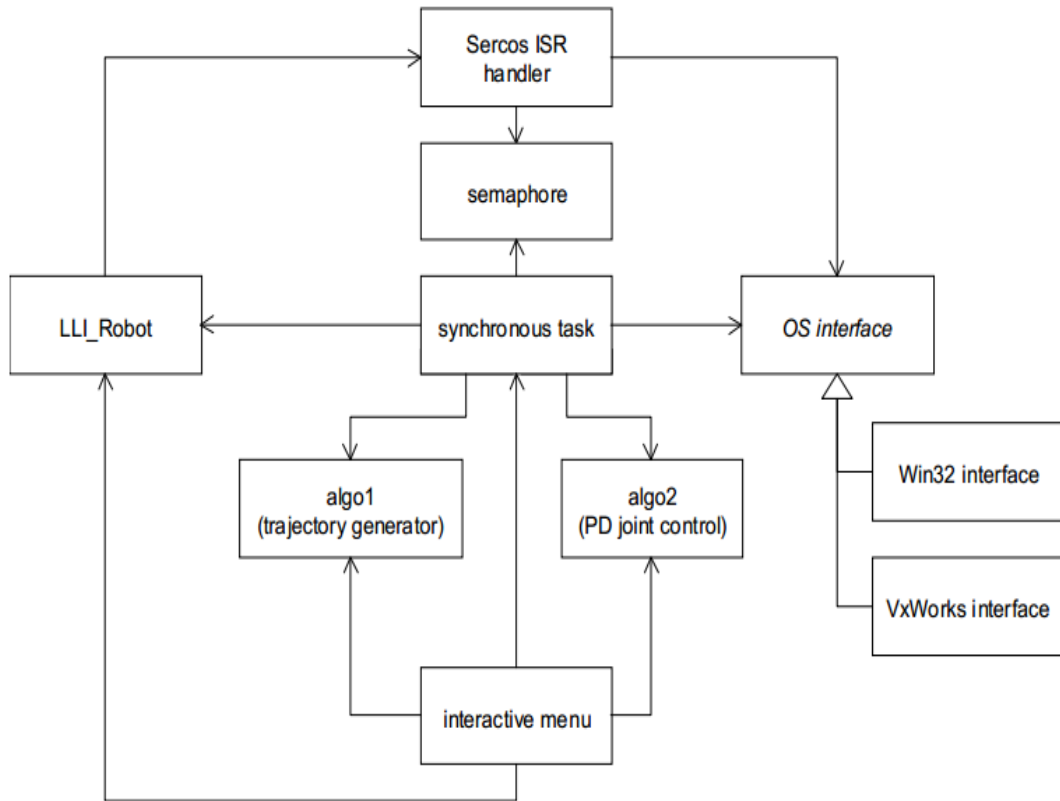
Şekil 3.49: Robot uç noktası kartezyen konumları(beşinci dereceden yörünge).

4.DENEYSEL ÇALIŞMALAR

4.1 Uygulama

2. ve 3. bölümlerde elde edilen modeller ve kontrol algoritmalarının deneysel sonuçları bu bölümde verilmektedir. Deneysel çalışmalar sırasında LLI programlama dili kullanılarak Staubli RX-160 manipülatörün kontrolü gerçekleştirilmektedir. LLI programı ile her 4 milisaniyede konum, hız ve tork geri besleme sinyalleri alınarak, konum, hız, ileri besleme hız ve ileri besleme tork sinyalleri robota gönderilmektedir.

Şekil 4.1’de LLI program akışı gösterilmektedir.

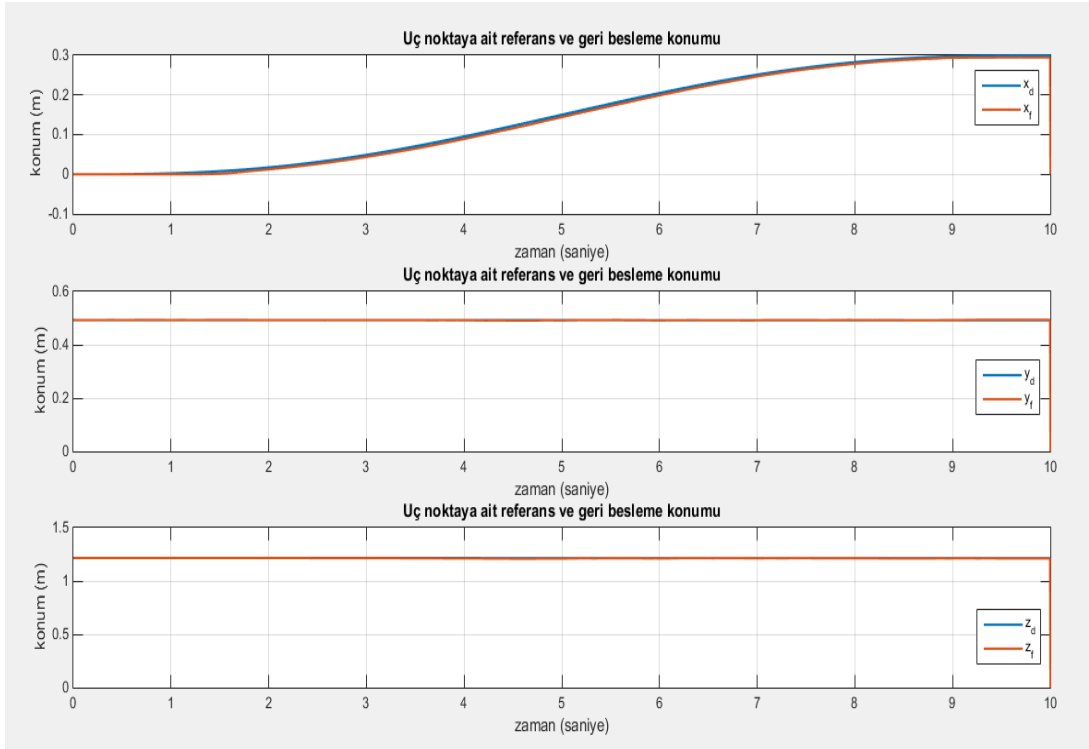


Şekil 4.1: LLI kullanıcı arayüz program akış şeması [2].

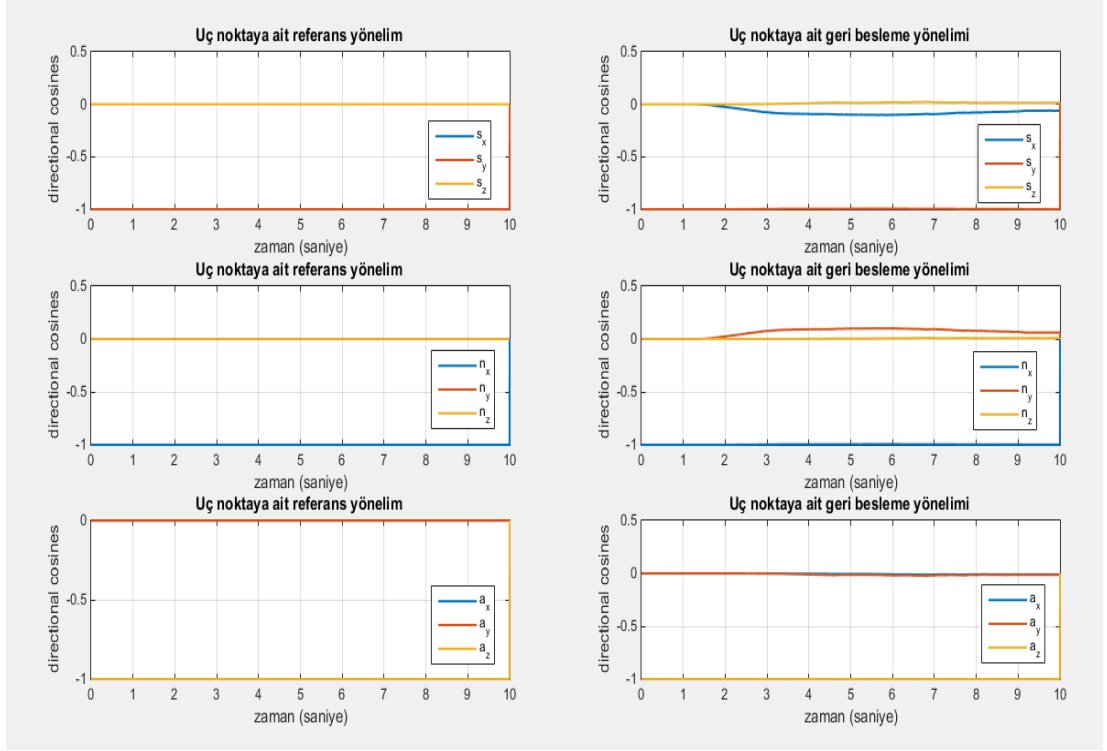
4.2 Görev Uzayında Konum Kontrolü

Gerçekleştirilen ilk deneyde robot uç noktası uzayda belirli bir noktadan, diğer bir noktaya götürülmektedir. Deneye başlamadan önce, uç nokta için görev uzayında tanımlı olan referans yörüngeler oluşturulmaktadır. Elde edilen bu yörüngeler sayesinde robot uç noktası, uzayda istenen konum, hız ve ivmelerde hareket edebilmektedir. Deneyde kontrolcü olarak görev uzayında tanımlanan ters dinamik kontrolcü şeması kullanılmaktadır. Robot kontrolü sırasında Newton-Euler denklemleri ile elde edilem dinamik model ve sistem tanıma ile elde edilen sürtünme ve yay modelleri kullanılmaktadır. Kontrol sırasında sensörler ile alınan eklemlere ait geri besleme konum ve hız değerleri Jakobiyen matrisi kullanılarak, uç nokta geri besleme konum ve hız değerlerine dönüştürülmektedir. Deneylerde uç nokta sırası ile x, y ve z yönlerinde hareket ettirilmektedir. Daha sonra hem x, hem y, hem de z yönünde birlikte hareket ettirilmiştir. Deneylerde uç nokta çok düşük bir konum hatası ile istenilen yönlerde hareket ettirilmektedir. Fakat sensörlerden gelen geri besleme hız verileri gürültülü olduğu için uç nokta hızları açısından küçük de olsa bir hata meydana gelmektedir.

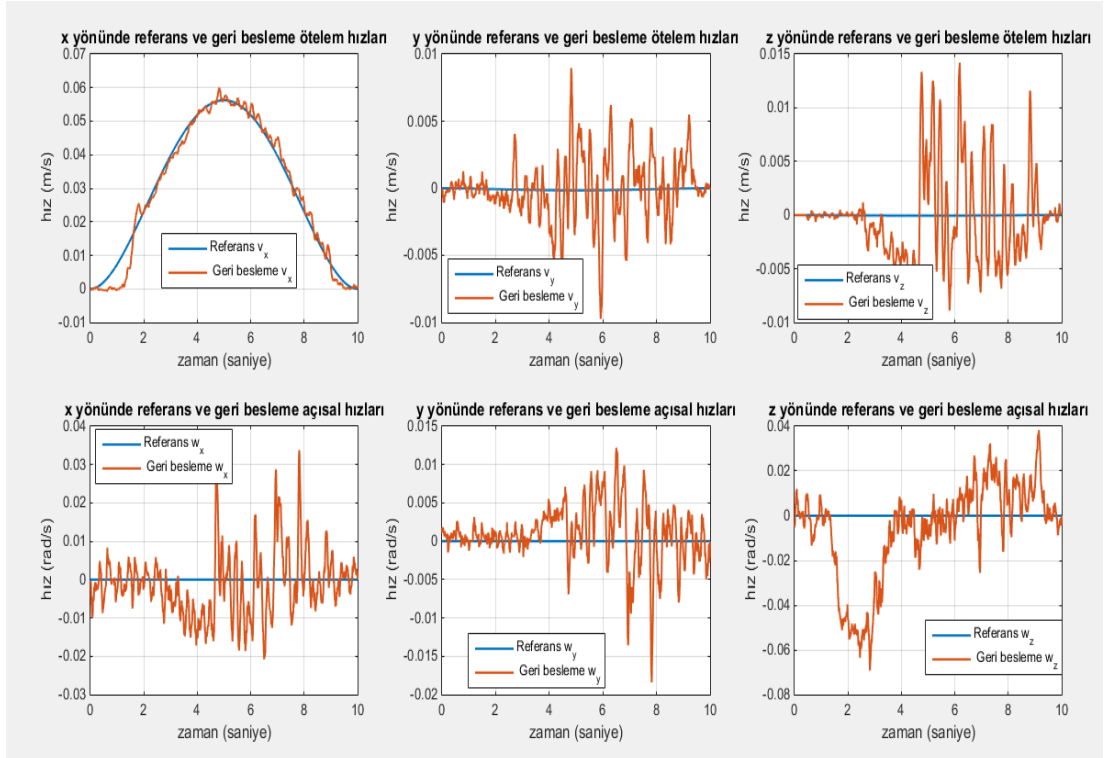
4.2.1 Doğrusal yörüngede hareket - X doğrultusu



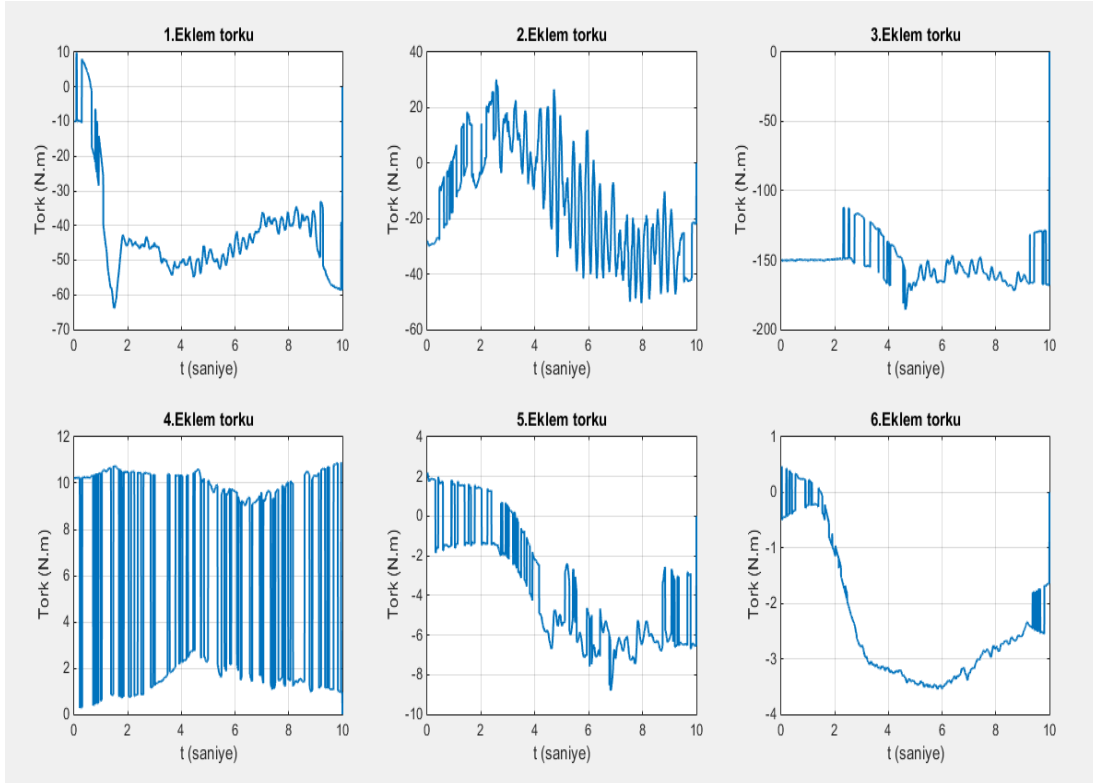
Şekil 4.2: Uç noktaya ait referans ve geri besleme çizgisel konumları.



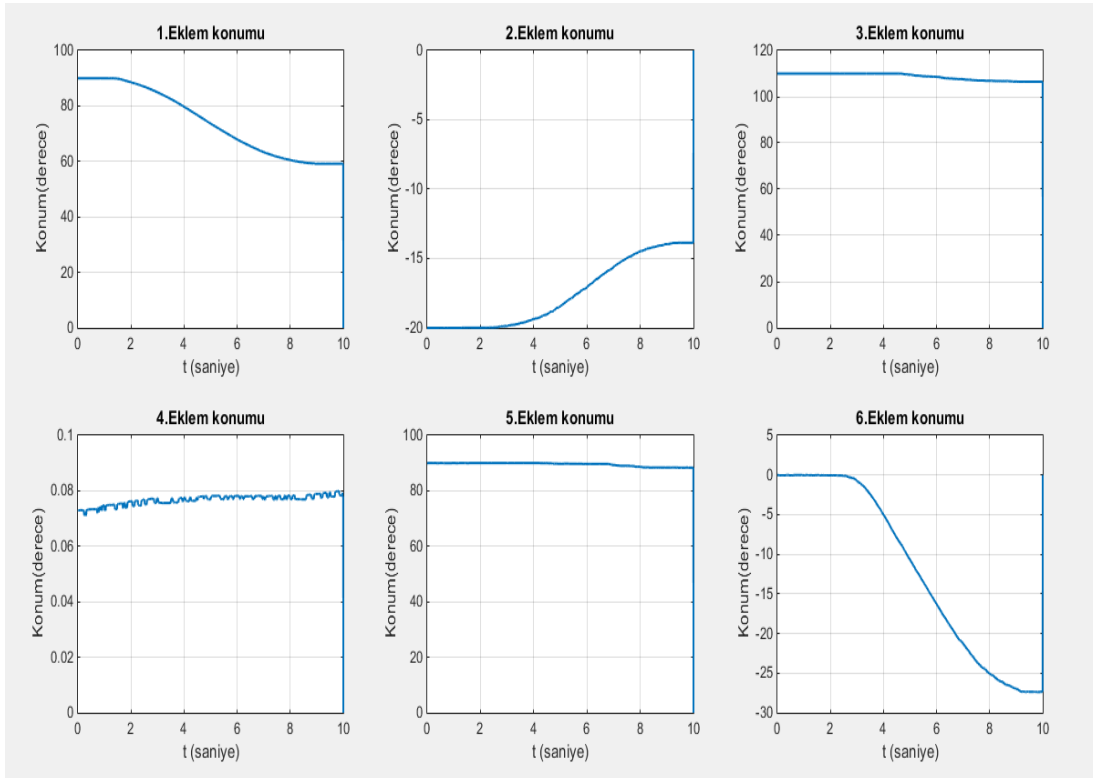
Şekil 4.3: Uç noktaya ait referans ve geri besleme dönme konumları.



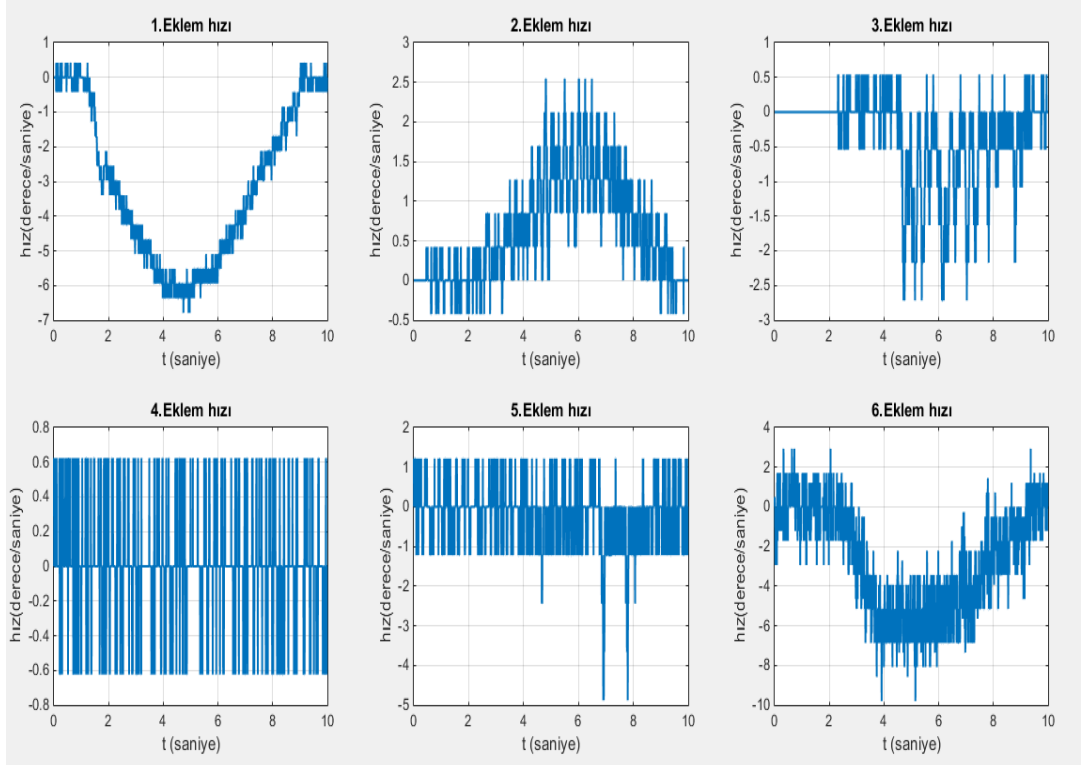
Şekil 4.4: Uç noktaya ait referans ve geri besleme hızları



Şekil 4.5: Hareket esnasında eklemlere uygulanan torklar.

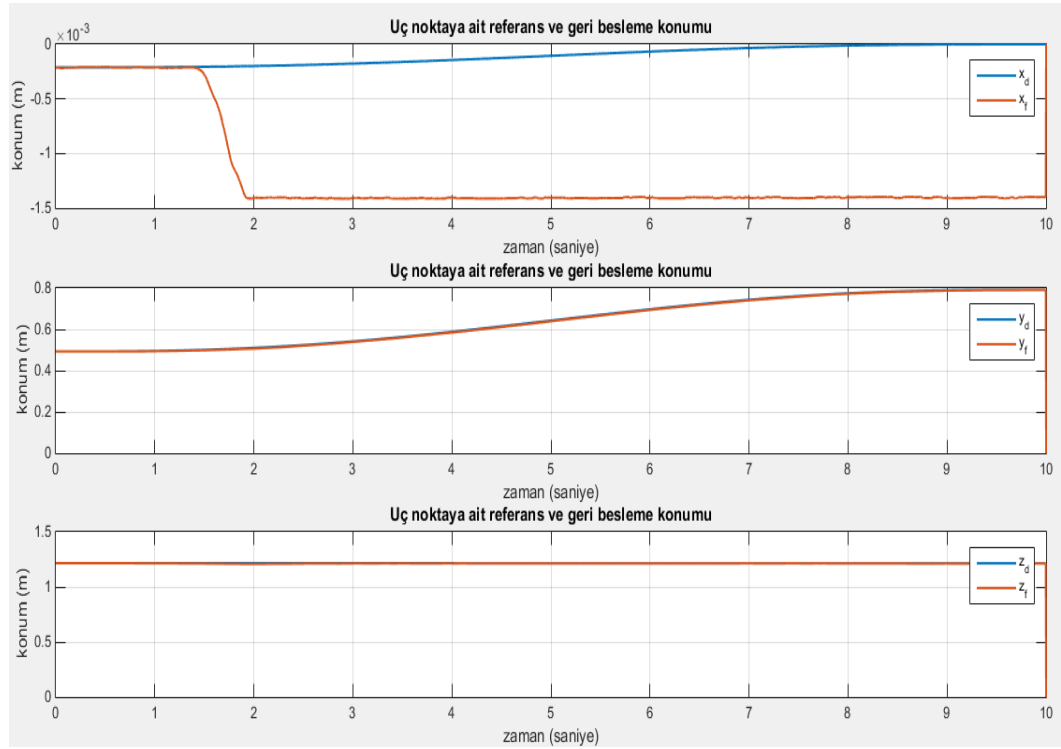


Şekil 4.6: Hareket esnasında eklemlerin yapmış olduğu konumları

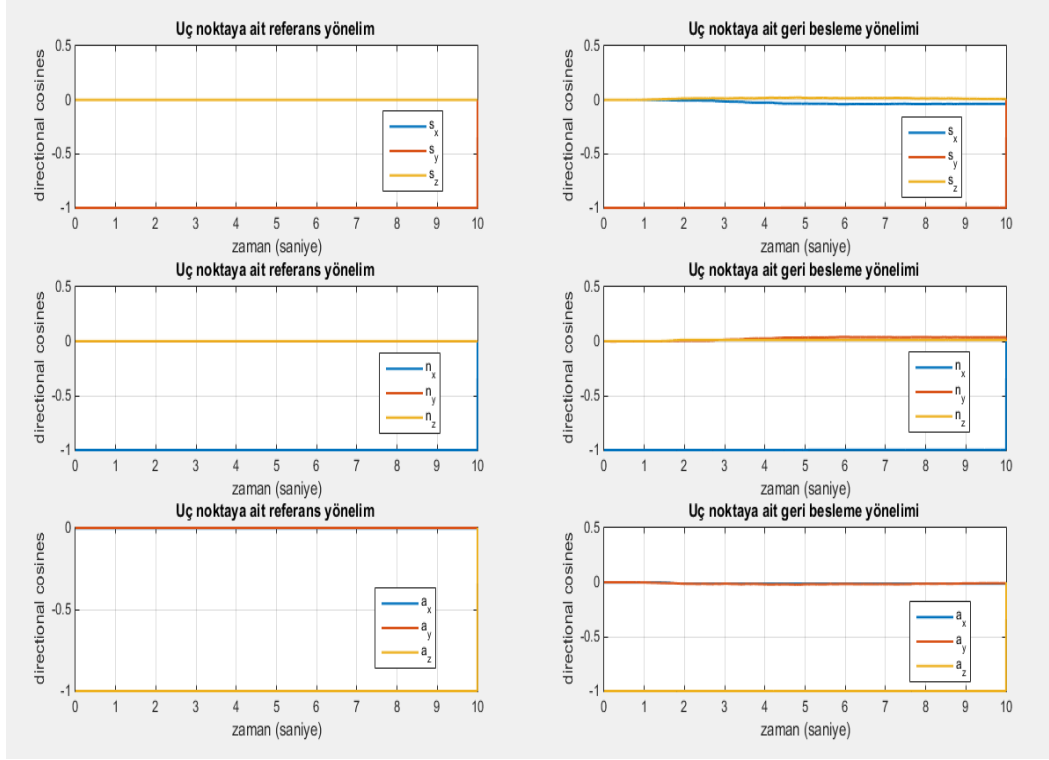


Şekil 4.7: Hareket esnasında eklemlerin yapmış olduğu hızları.

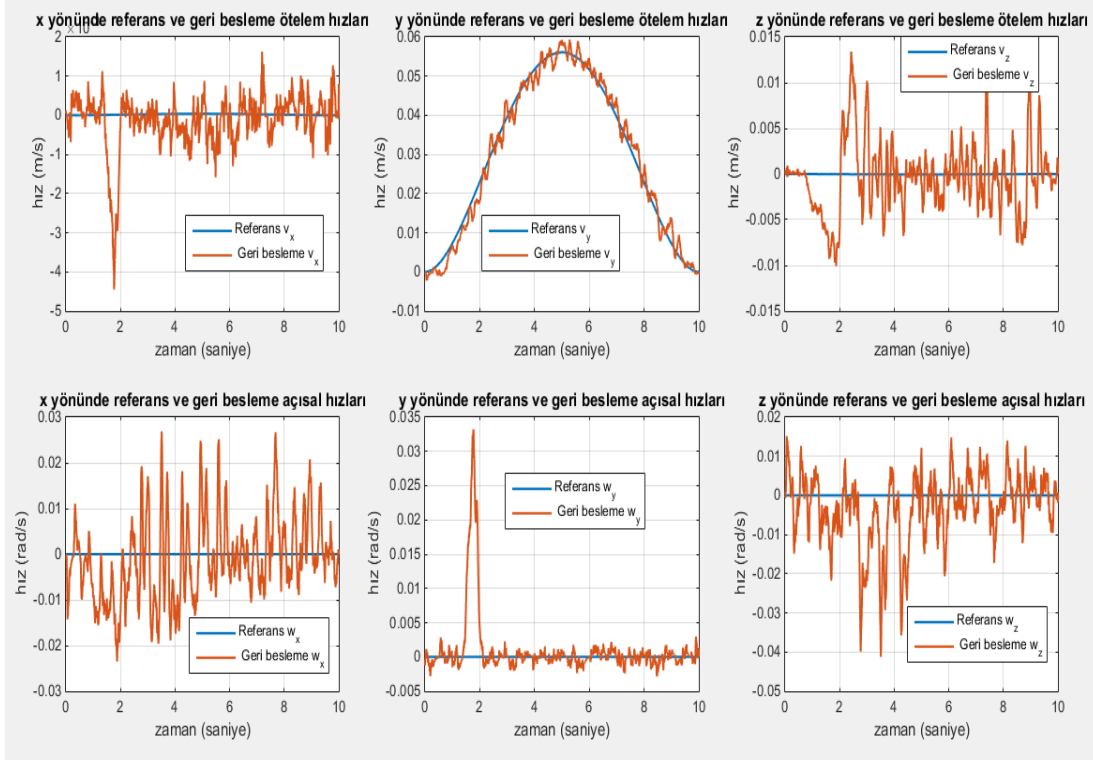
4.2.2 Doğrusal yörüngede hareket - Y doğrultusu



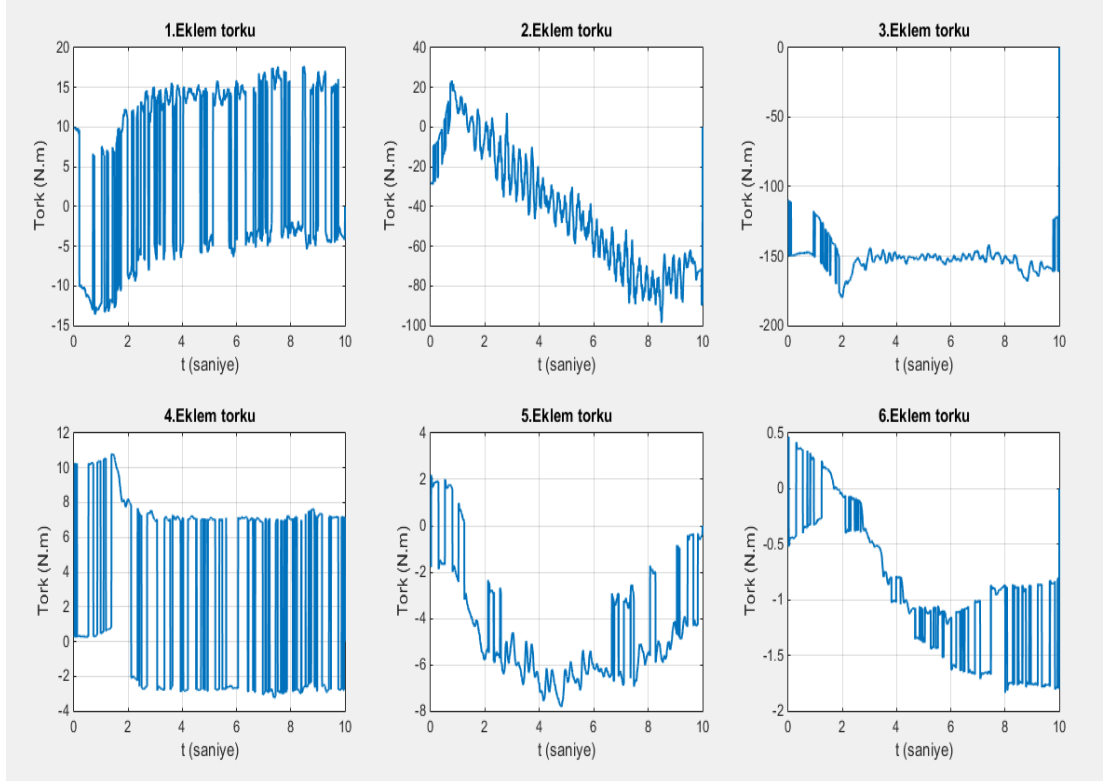
Şekil 4.8: Uç noktaya ait referans ve geri besleme çizgisel konumları.



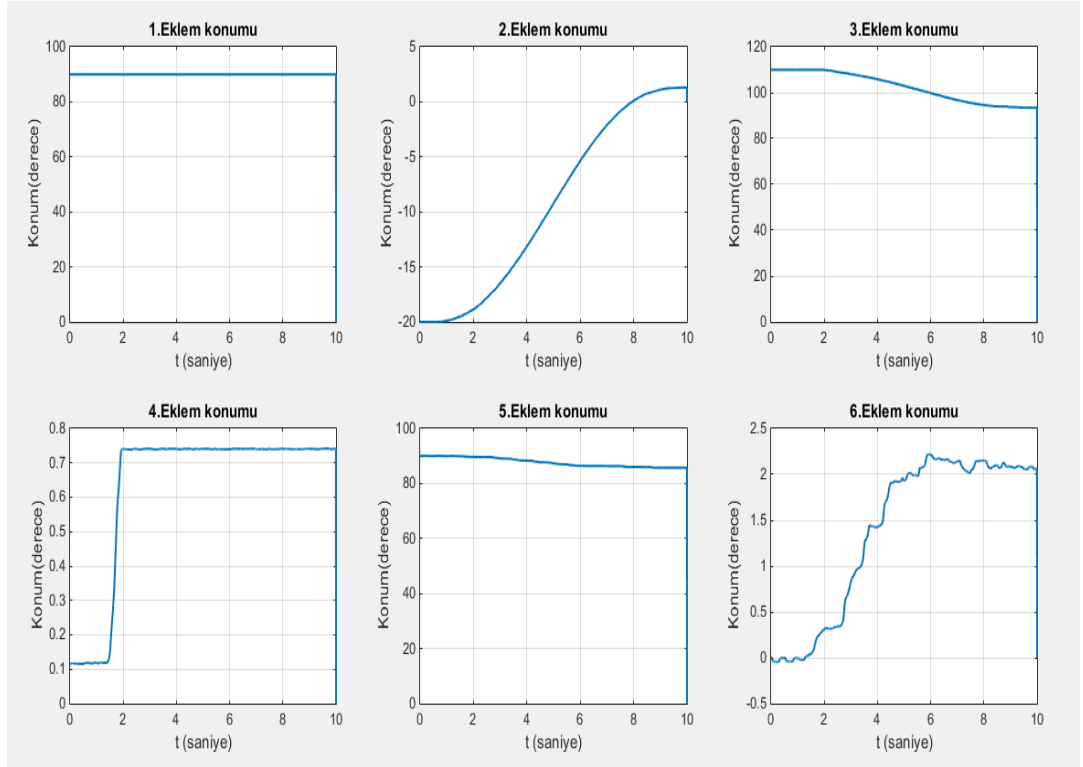
Şekil 4.9: Uç noktaya ait referans ve geri besleme dönme konumları.



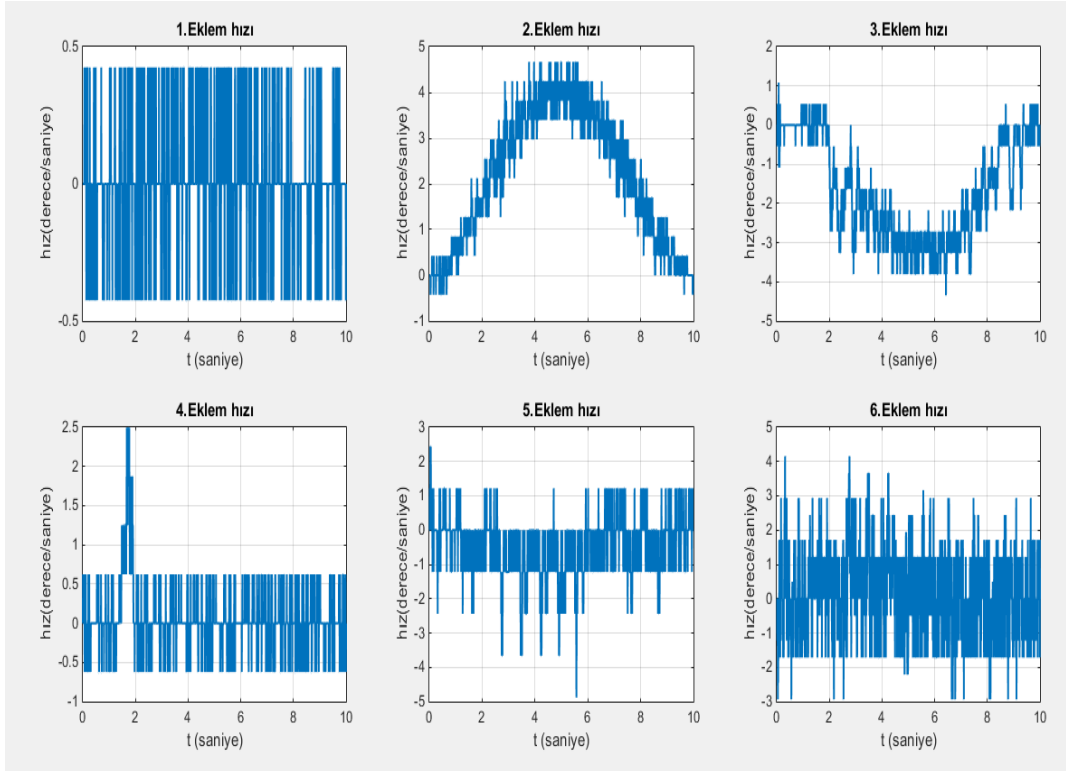
Şekil 4.10: Uç noktaya ait referans ve geri besleme hızları.



Şekil 4.11: Hareket esnasında eklemlere uygulanan torklar.

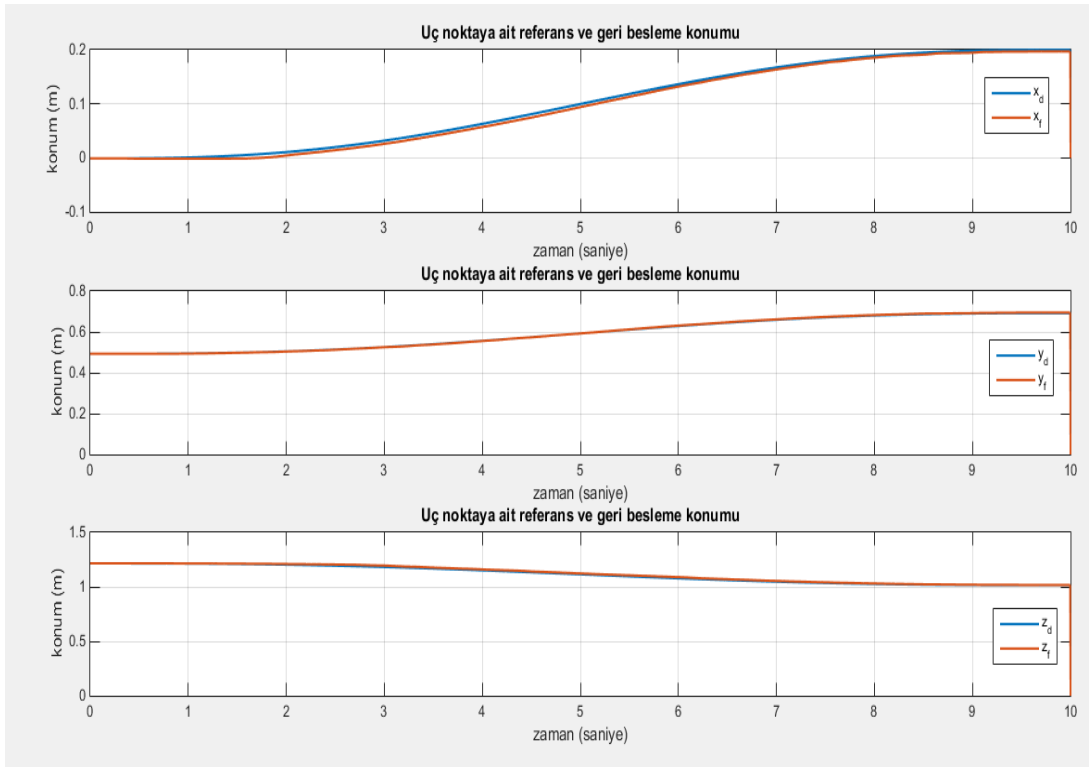


Şekil 4.12: Hareket esnasında eklemlerin yapmış olduğu konumları.

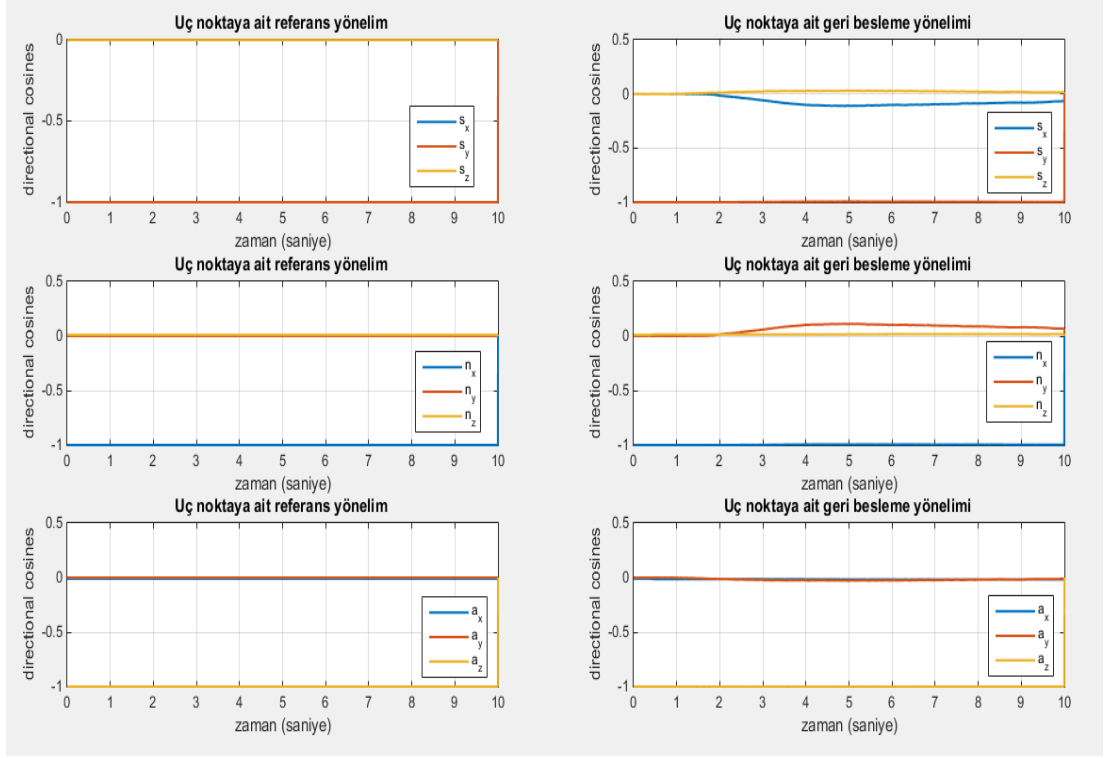


Şekil 4.13: Hareket esnasında eklemlerin yapmış olduğu hızları.

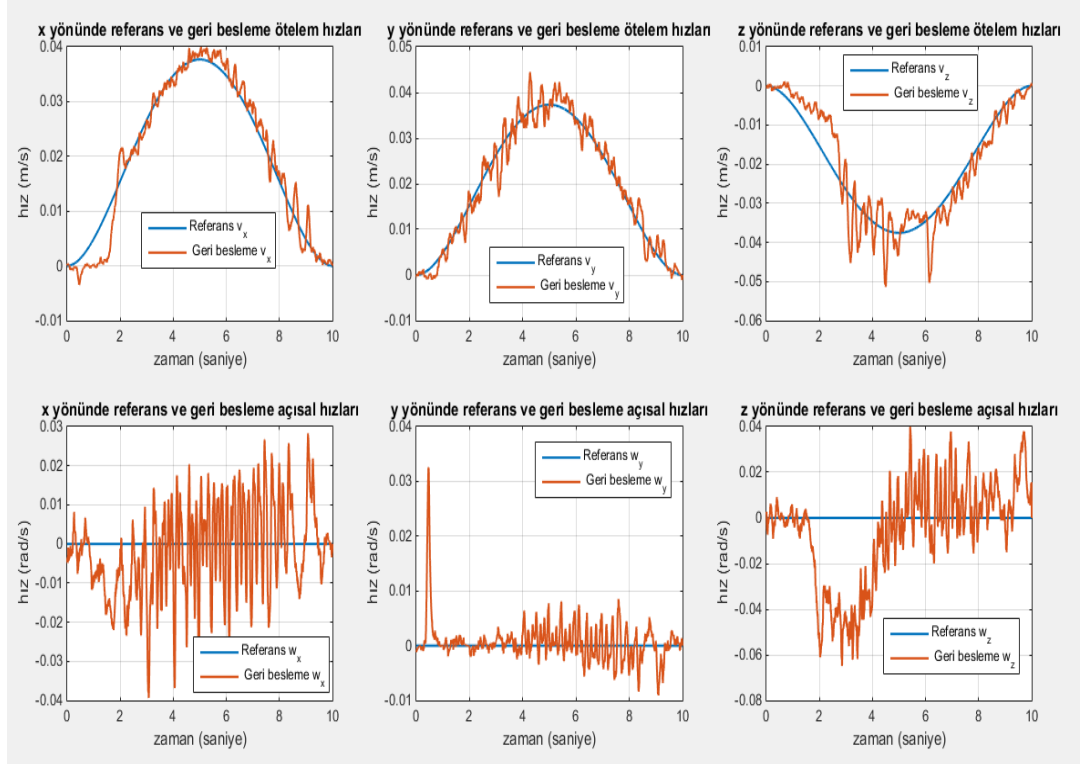
4.2.3 Doğrusal yörüngede hareket - X, Y ve Z doğrultusu



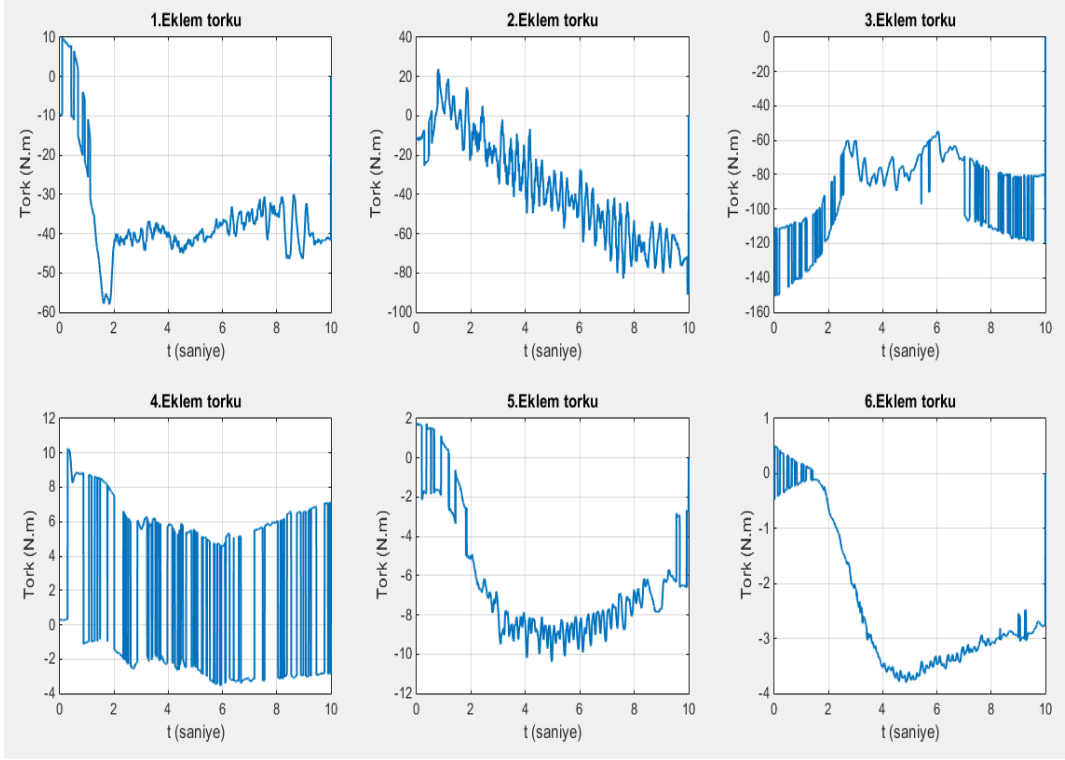
Şekil 4.14: Uç noktaya ait referans ve geri besleme çizgisel konumları.



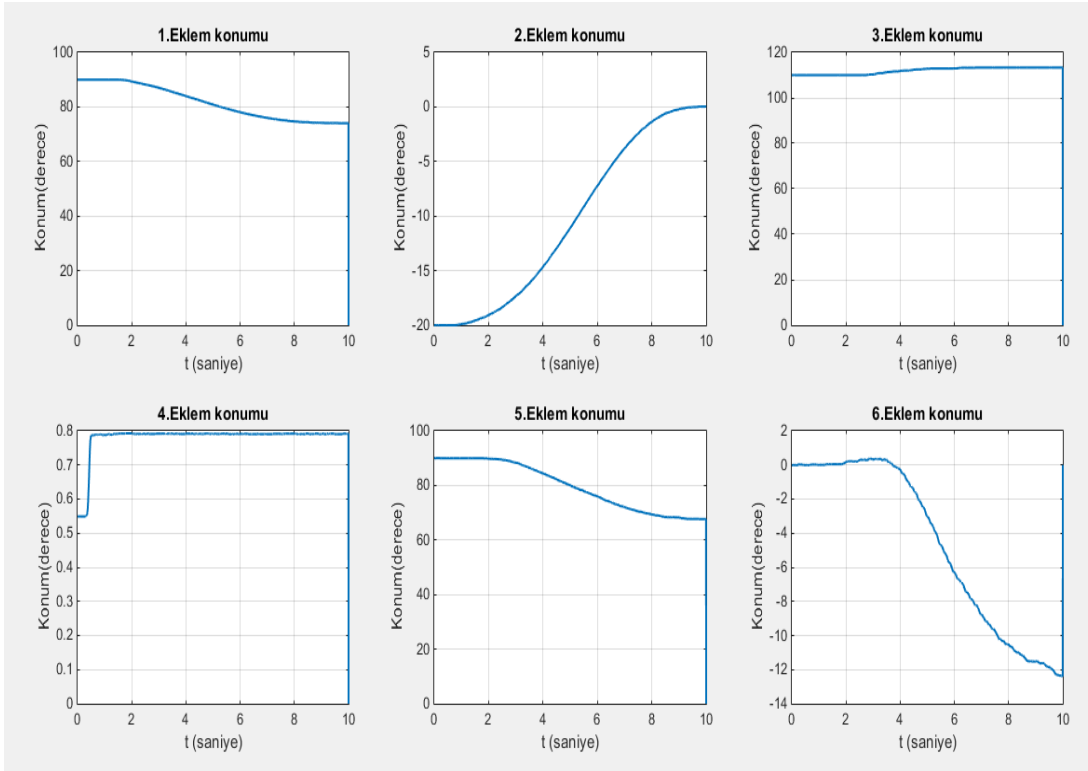
Şekil 4.15: Uç noktaya ait referans ve geri besleme dönme konumları.



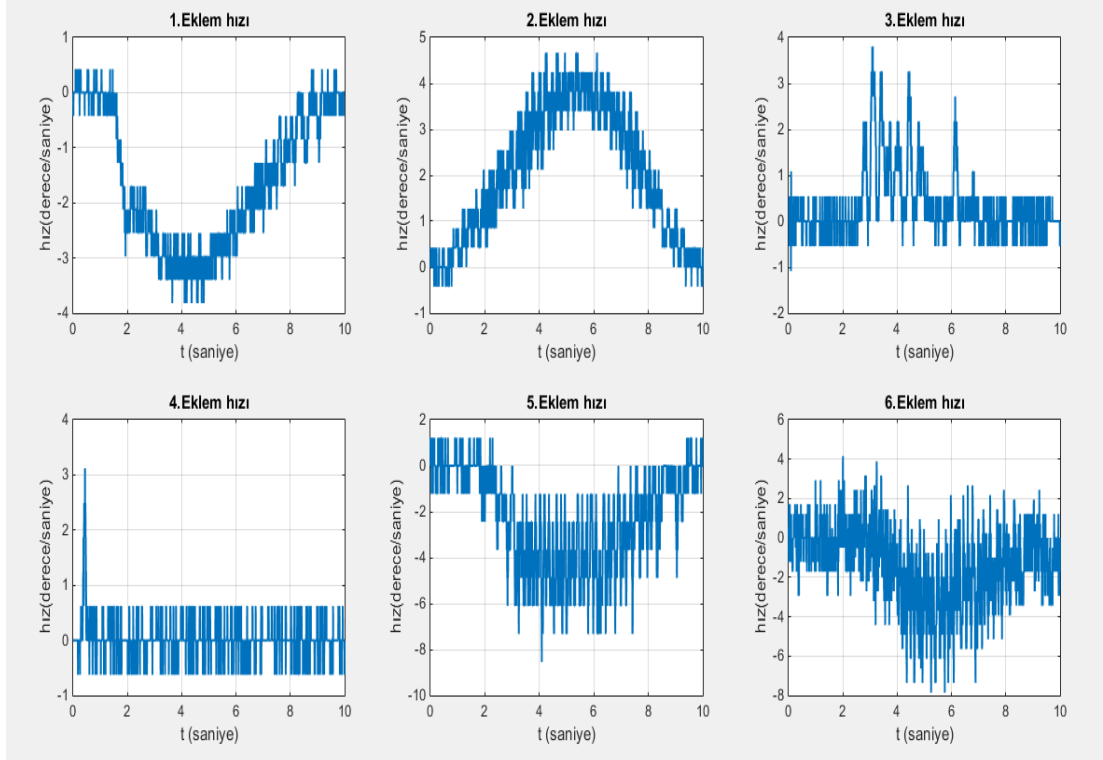
Şekil 4.16: Uç noktaya ait referans ve geri besleme hızları.



Şekil 4.17: Hareket esnasında eklemlere uygulanan torklar.



Şekil 4.18: Hareket esnasında eklemlerin yapmış olduğu konumları.



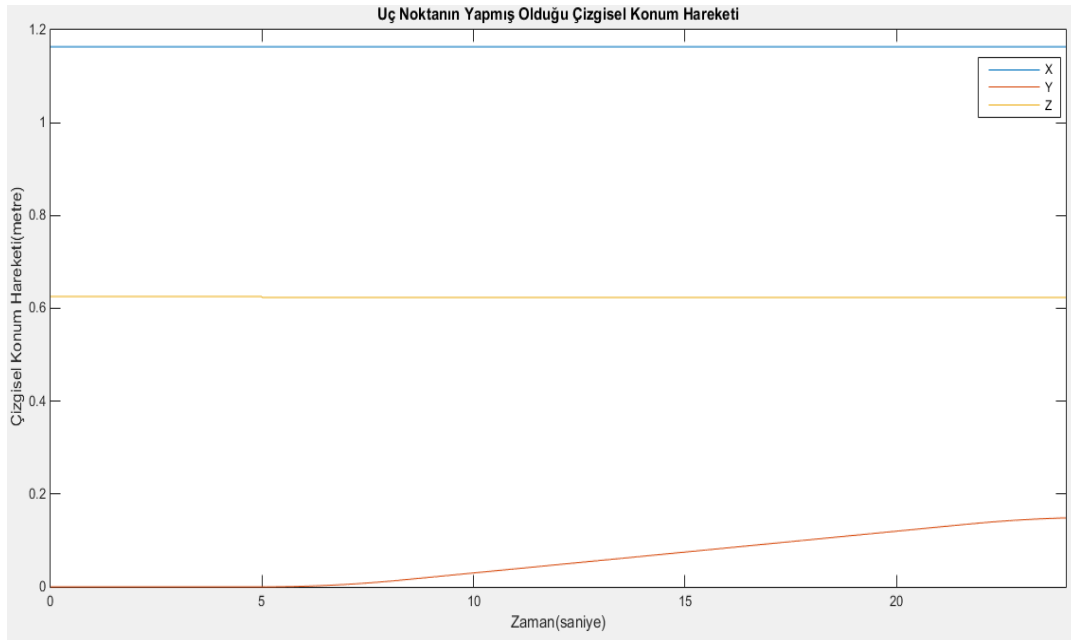
Şekil 4.19: Hareket esnasında eklemlerin yapmış olduğu hızları.

4.3 Görev Uzayında Sensörsüz Hibrid Konum/Kuvvet Kontrolü

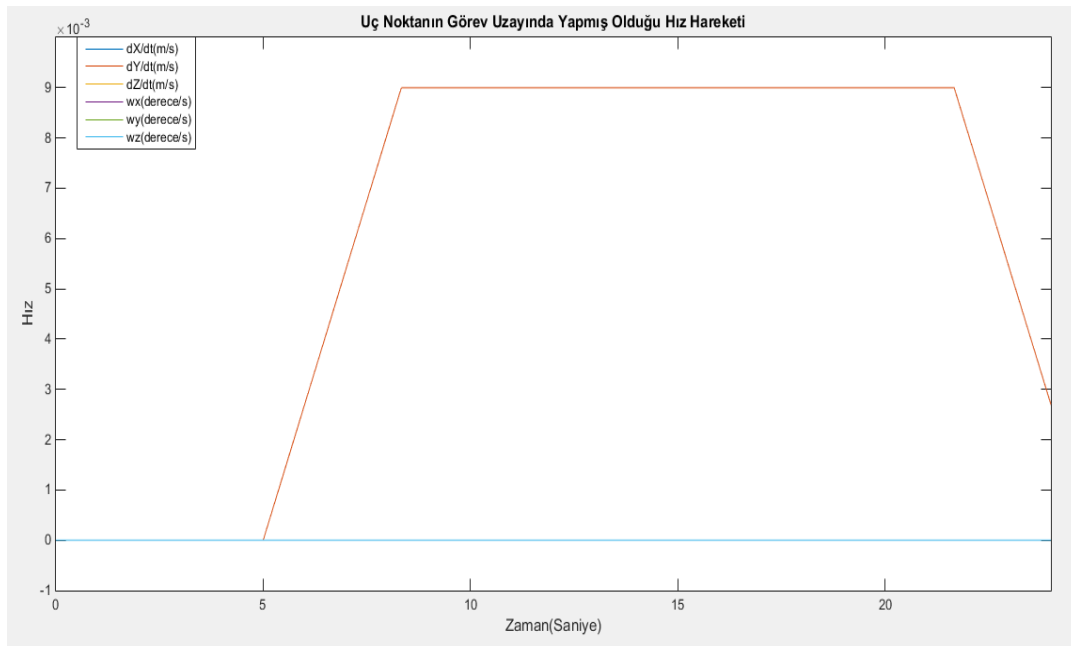
Gerçekleştirilen ikinci deney görev uzayında hibrid konum/kuvvet kontrolüdür. Robot çevre ile temas edene kadar uzayda bir konum hareketi gerçekleştirilir. Serbest uzayda bu konum hareketi, ters dinamik kontrol metodu kullanılarak gerçekleştirilmektedir. Robot uç noktası çevre ile temas eder etmez, robot hibrid konum/kuvvet algoritması ile kontrol edilmeye başlanmaktadır. Deney sırasında robot z yönünde çevresine 10 N'luk bir kuvvet uygularken x ve y yönlerinde çizgisel hareket gerçekleştirmektedir. Deney sırasında, uç nokta ile çevresi arasındaki tepki kuvveti hem kuvvet/tork sensörü ile ölçülmekte, hem de sensörsüz tahmin metodu ile hesaplanmaktadır. Deney sırasında z yönünde tahmin edilen kuvvet, sensörden ölçülen kuvvet değerinden bir miktar sapmaktadır. Bu sapmanın ilk nedeni, kullanılan sürtünme modelinin tam olarak eklemlerde oluşan sürtünmeleri hesaplayamamasıdır. Tahmin edilen kuvvet-moment değerinin başarısı doğrudan hesaplanan eklem tepki torkları ile ilişkilidir. Kullanılan sürtünme modelinin yetersiz olması nedeni ile hesaplanan tepki torku yalnız çevre teması nedeniyle oluşan torkları değil, aynı zamanda gerçekte oluşan sürtünme torkları ile modelden elde edilen sürtünme torkları arasındaki farkları içermektedir. İkinci ana neden ise,

geri besleme tork sensöründen gelen verilerin gürültülü olmasıdır. Bu sebepten dolayı tahmin edilen kuvvet değeri, sensörden ölçülen kuvvet değerinden daha gürültülü olmaktadır. Sensörsüz olarak yapılan ilk hibrid konum/kuvvet kontrolünün aksine ikinci hibrid kontrolünde geri besleme tork sensörlerinden gelen eklem torkları Kalman filtresi ile filtrelenmektedir. Bu sayede ikinci deneyde, tahmin edilen tepki kuvveti ile sensörden ölçülen tepki kuvveti arasındaki fark azalmaktadır.

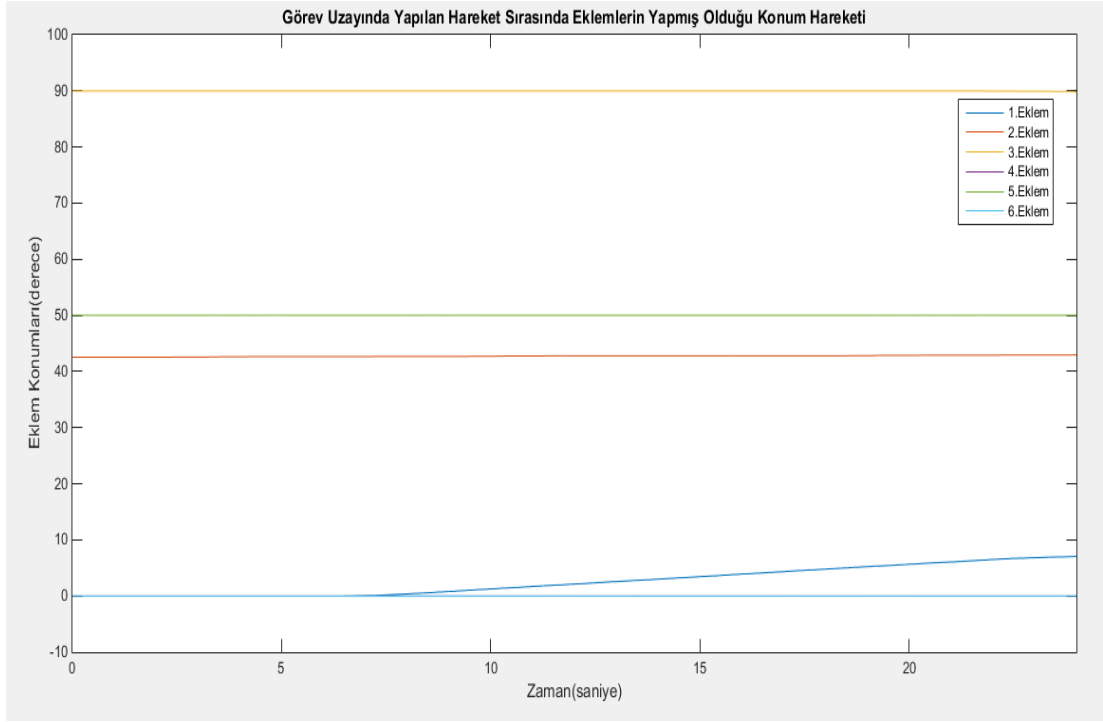
4.3.1 Doğrusal hareket - Y doğrultusu, kuvvet kontrolü - Z doğrultusu



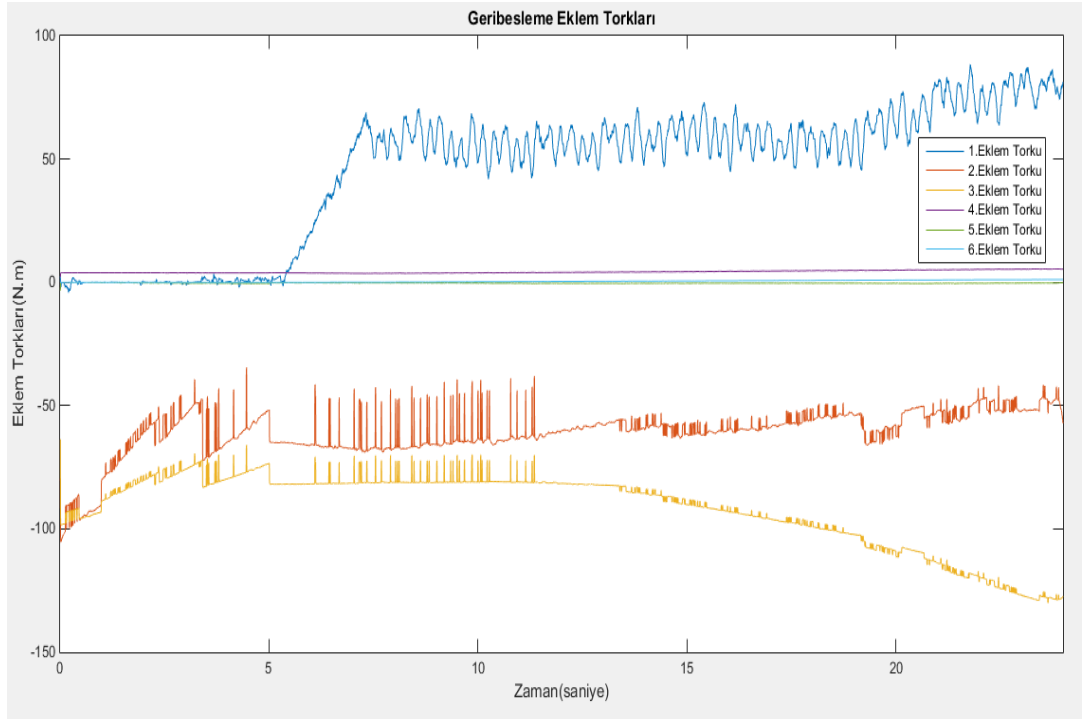
Şekil 4.20: Görev uzayında uç noktanın yapmış olduğu çizgisel hareket.



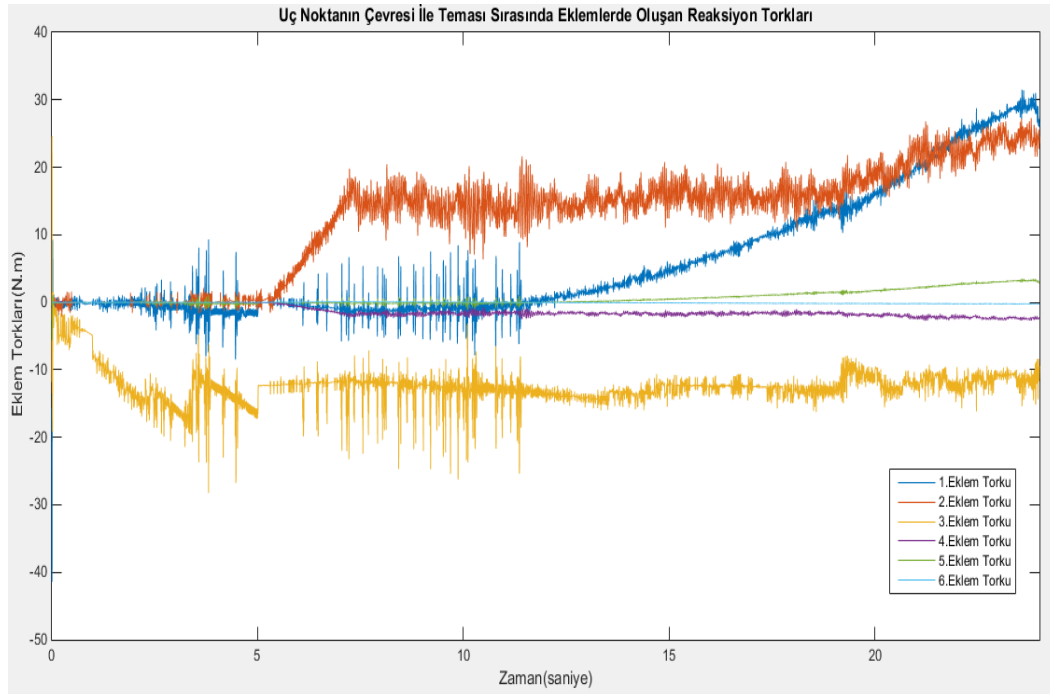
Şekil 4.21: Görev uzayında uç noktanın yapmış olduğu hızlar.



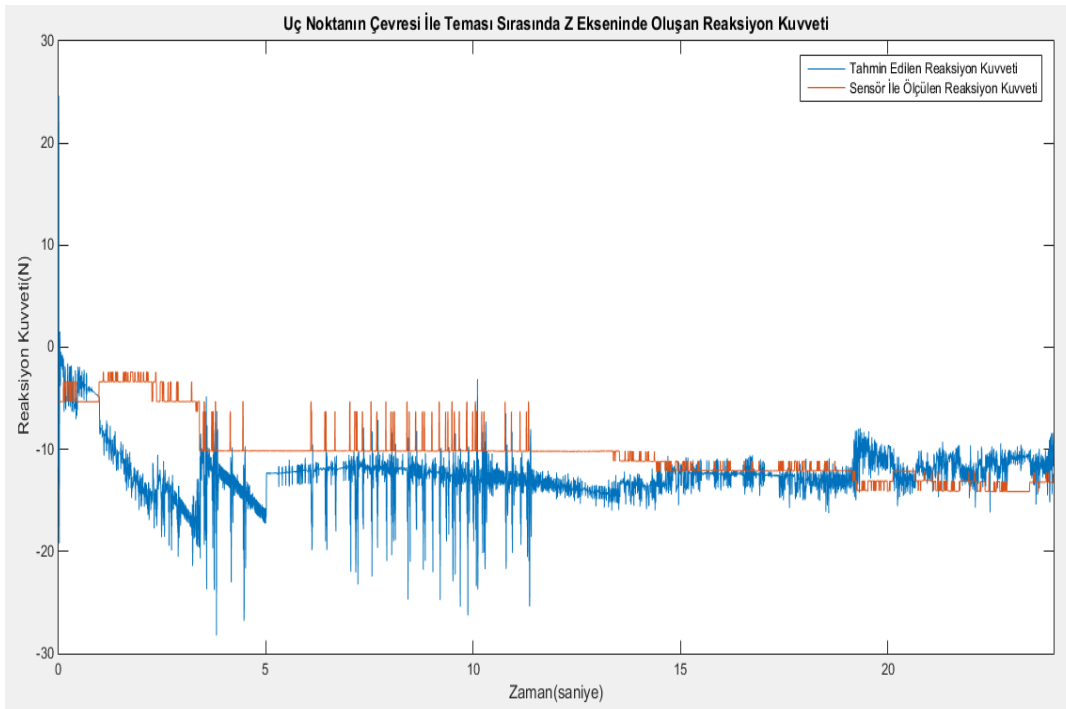
Şekil 4.22: Görev uzayında yapılan hareket sırasında eklem konumları.



Şekil 4.23: Hareket sırasında sensörlerden ölçülen eklem torkları.

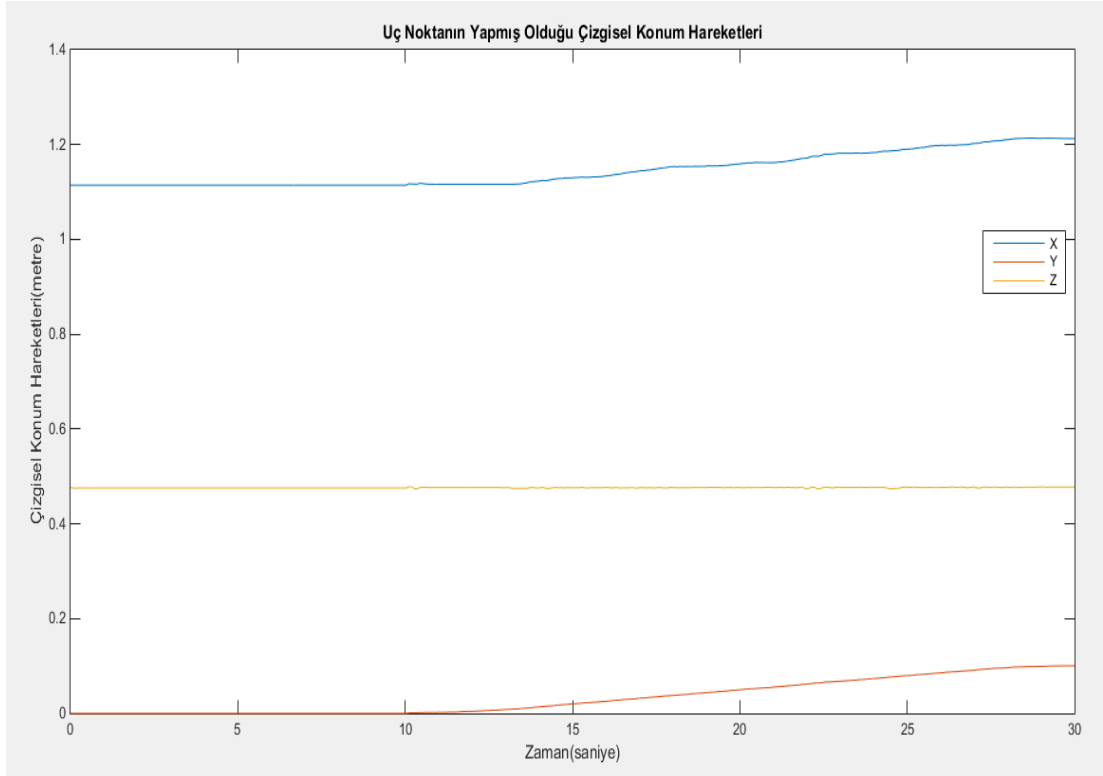


Şekil 4.24: Uç noktanın çevresi ile teması sırasında eklemlerde oluşan tepki torkları.

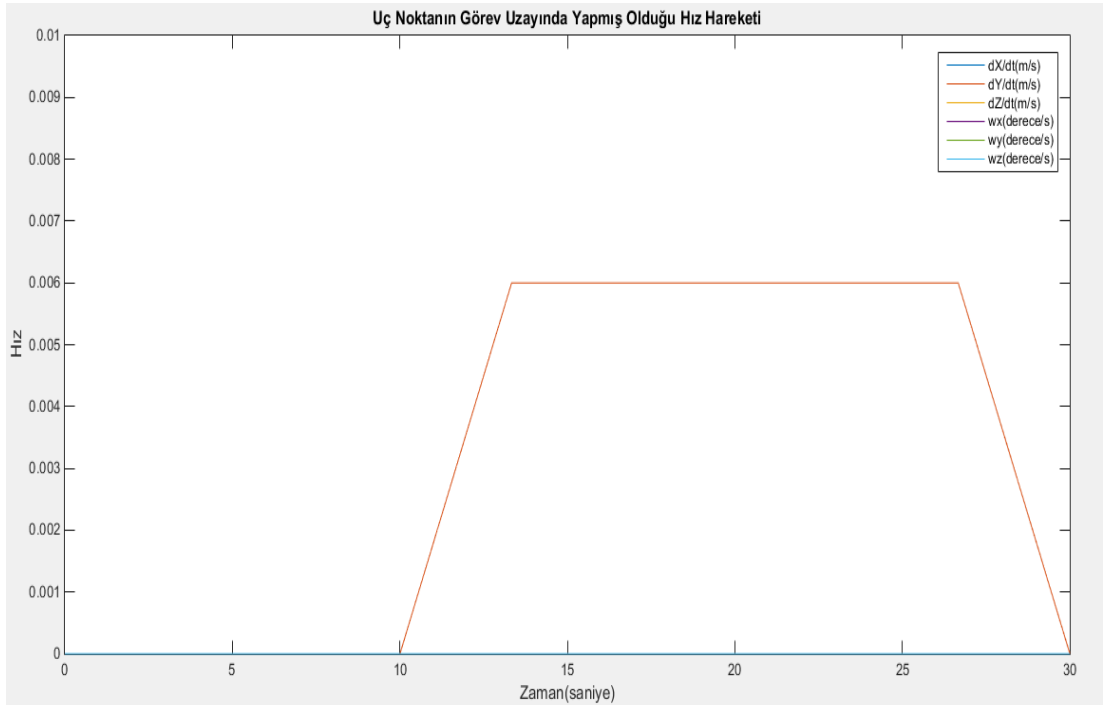


Şekil 4.25: Kuvvet-Tork sensöründen ölçülen ve tahmin edilen tepki kuvveti.

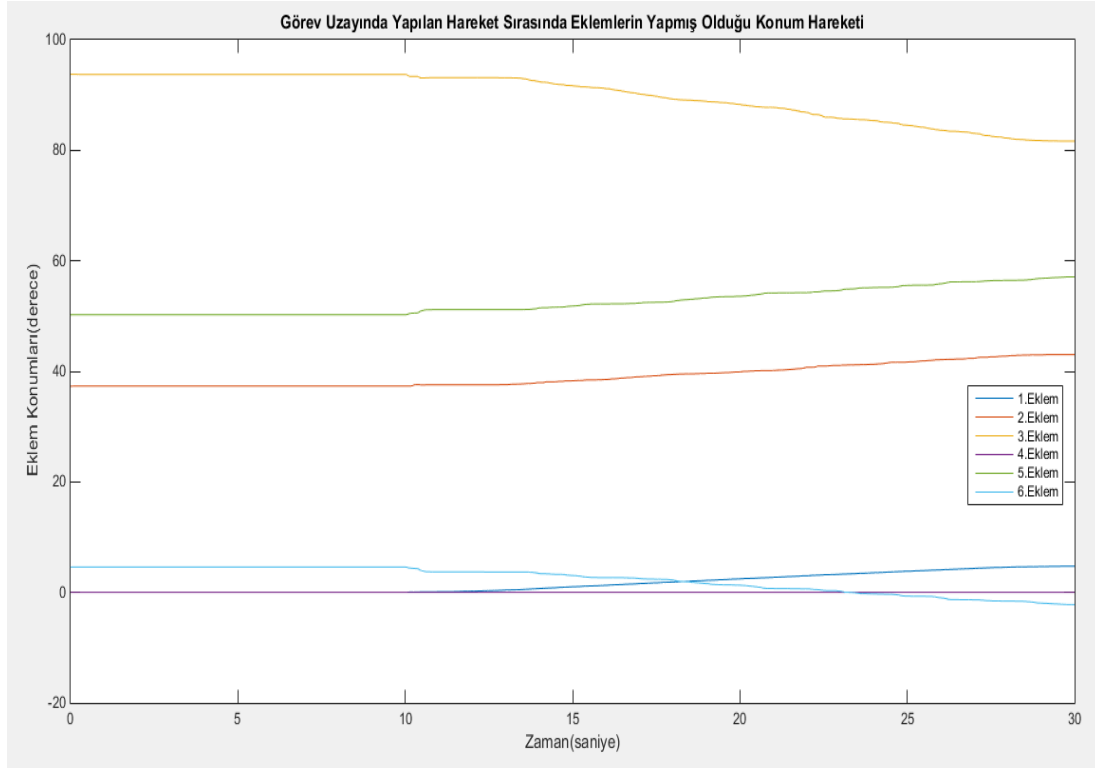
4.3.2 Doğrusal hareket - X ve Y doğrultusu, kuvvet kontrolü - Z doğrultusu



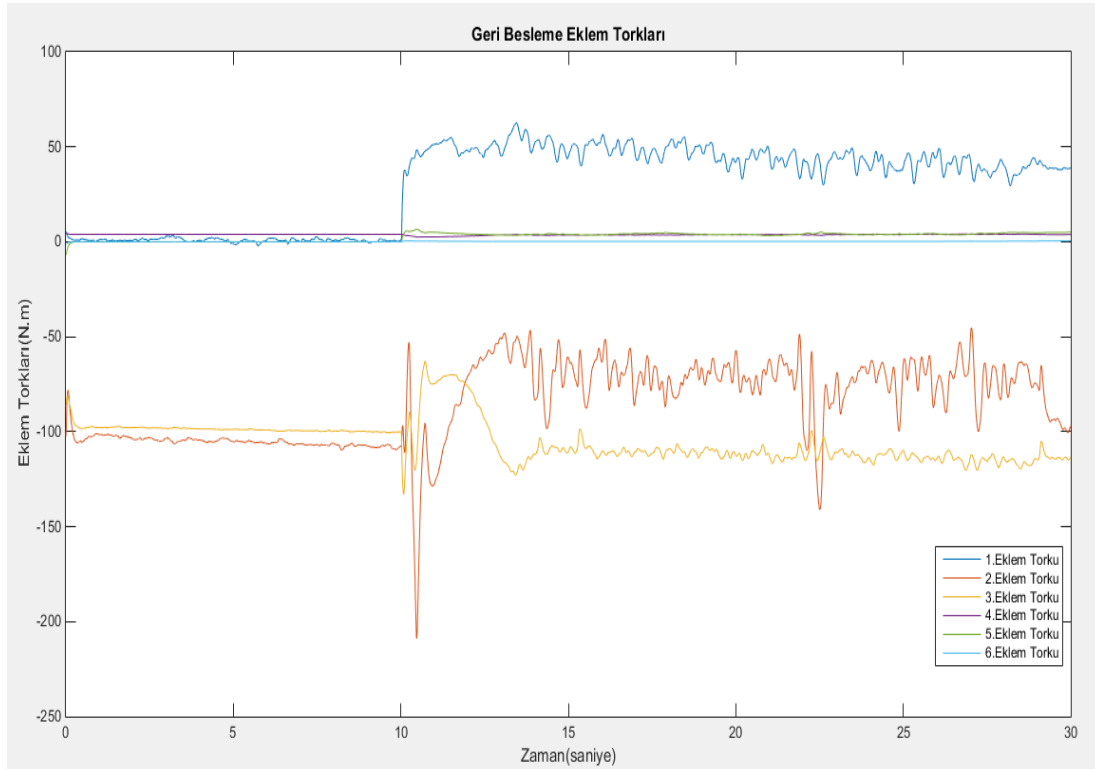
Şekil 4.26: Görev uzayında uç noktanın yapmış olduğu çizgisel hareket.



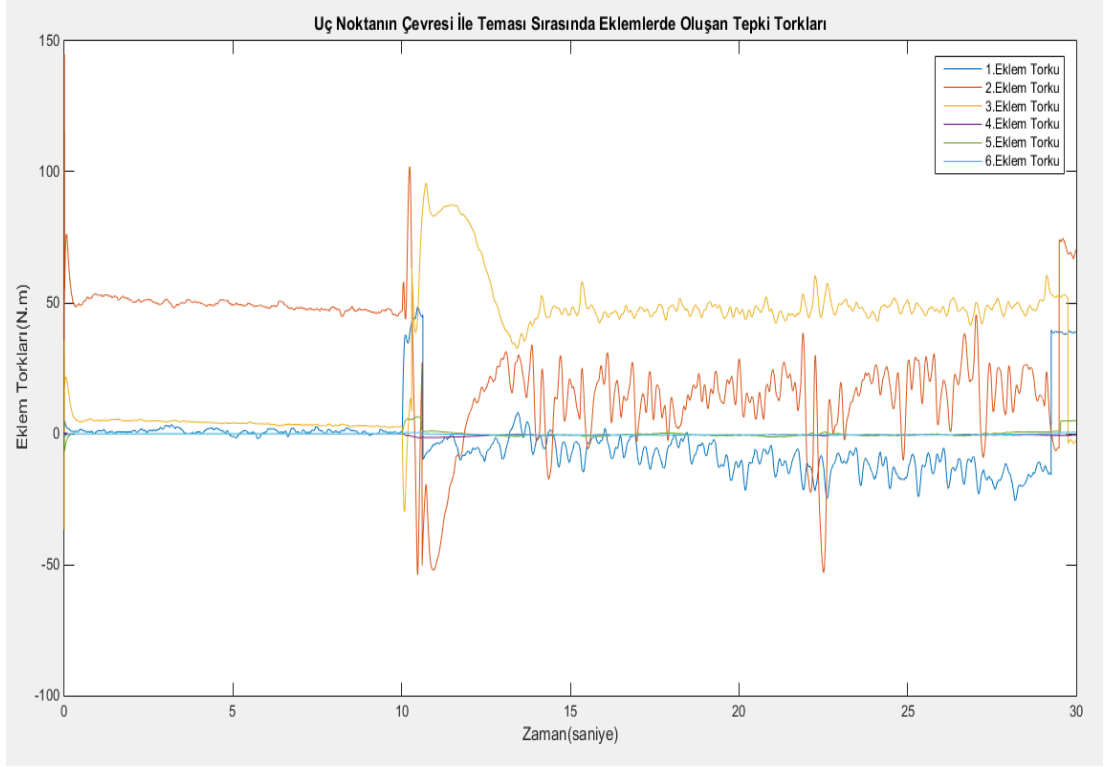
Şekil 4.27: Görev uzayında uç noktanın yapmış olduğu hızlar.



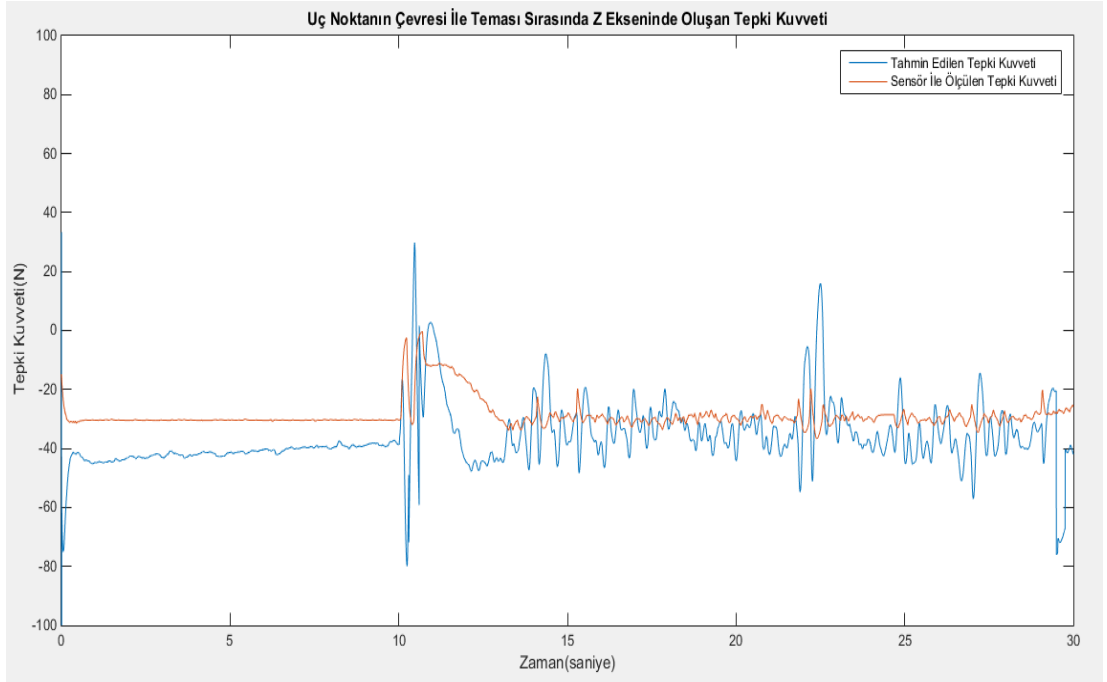
Şekil 4.28: Görev uzayında yapılan hareket sırasında eklemler konumları.



Şekil 4.29: Hareket sırasında sensörlerden ölçülen eklemler torkları.



Şekil 4.30: Uç noktanın çevresi ile teması sırasında eklemlerde oluşan tepki torkları.



Şekil 4. 31: Kuvvet-Tork sensöründen ölçülen ve tahmin edilen tepki kuvveti.

5. SONUÇ VE ÖNERİLER

Tez sırasında robota ait modellerin ve kontrol algoritmalarının benzetimleri Matlab Simulink programı kullanılarak yapılmıştır. Yapılan deneyler ile benzetim sonuçları karşılaştırılmıştır. Deneyler Staubli RX-160 manipülatörünün C kütüphanesi LLI kullanılarak yapılmıştır. Deneyler sırasında konum, hız, eklem torkları ve uç noktaya ait kuvvet ve moment verileri sensörler vasıtası ile her 4 milisaniyede alınıp, eklemlere ait referans tork değerleri motor sürücülerini vasıtası ile robot eklemlerine gönderilmiştir. Deneyler sırasında 4 milisaniyede Euler-Lagrange denklemleri kullanılarak hesaplanan Coriolis ve merkezkaç etkisini ifade eden $C(q, \dot{q})$ matrisi hesaplanamadığından robota ait dinamik model Newton-Euler denklemleri ile de türetilmiştir. Konum kontrolü gerçekleştirilirken eklem hız sensörlerinden gelen verilerin gürültülü olduğu gözlenmiş ve kalman filtresi uygulanmıştır. Kuvvet kontrolüne ait deneylere geçildiğinde laboratuvar ortamının ve robotun deneyler için yetersiz olduğu görülmüştür. Robot ucuna monte edilen kuvvet-moment sensörüne ait verilerin her 4 milisaniyede bir LLI programına gelmesi için mevcut robot kontrolörü güncellenmiştir. Kuvvet kontrolü uygulanırken robota zarar gelmemesi için deney düzeneği kurulmuştur. Robot uç noktasına yay sistemi monte edilerek olası zararlardan kaçınılmıştır. Son deneylerde tahmin edilen tepki kuvveti ile sensörden okunan tepki kuvveti karşılaştırılarak sensör kullanılmadan uç noktaya ait tepki kuvvetinin bulunabileceği gösterilmiştir. Bu tez, daha önce yapılan tezlerin devamı olup, gelecekte Staubli RX-160 manipülatörleri kullanılarak yapılacak olan çalışmalara da ışık tutabilecektir. Gelecekte yapılacak çalışmalar için sırası ile aşağıdaki öneriler yapılabilir.

- Hız sensörlerinden gelen gürültüyü azaltmak için mevcut Kalman filtresine ait parametreler değiştirilerek daha gürültüsüz sensör verisi elde edilebilir.
- Mevcut sürtünme modeli yapılacak kontrol senaryolarına göre daha hassas modellenebilir.

- Robot uç noktasında yer alan kuvvet-moment sensörünün kullanılmasına gerek duyulmayacağı için, sensör kullanılmadan doğrudan kuvvet kontrolüne ait uygulamalar geliştirilebilir.
- Gerek konum gerek kuvvet kontrol algoritmaları tez sırasında elde edildiği için doğrudan robotlar kullanılarak çapak alma, montaj gibi senaryolar gerçekleştirilebilir.

KAYNAKLAR

- [1] **Staubli Arm – RX series 160 family Instruction Manual** (2008)
- [2] **Egemen Zengin** (2013). Stäubli RX160 Manipülatörün Modellenmesi Tanınması ve Kontrolü (doktora tezi).
- [3] **Url-3** <<http://www.ati-ia.com/products/ft/sensors.aspx>>, alındığı tarih: 07:1:2015.
- [4] **F. Pertin and J.-M. Bonnet-des-Tuves** (2014). Real time robot controller abstraction layer. In Proc.Int. Symposium on Robots (ISR), Paris, France, Mart.
- [5] **Mark W. Spong, Seth Hutchinson, and M. Vidyasagar** Robot Modeling and Control, First Edition.
- [6] **John J. Craig**, Introduction to Robotics Mechanics and Control, Third Edition.
- [7] **W. Khalil, E. Dombre** (2002). Modeling, Identification and Control of Robots.
- [8] **Url-2** <<http://www.staubli.com/en/robotics/manipulator-support-data>>, alındığı tarih: 12:07:2014.
- [9] **Toshiyuki Murakami, Ryo Nakamura, Fangming Yu, and Kouhei Ohnishi** (1993). Force Sensorless Impedance Control By Disturbance Observer.
- [10] **Toshiyuki Murakami, Fangming Yu, and Kouhei Ohnishi** (1993). Torque Sensorless Control in Multidegree-of-Freedom Manipulator. IEEE Transactions On Industrial Electronics, Vol. 40, No. 2.

EKLER

EK A1: İleri geometrik model benzetimi

EK A2: İleri geometrik model Matlab kodu

EK B: Ters geometrik model benzetimi

EK C1: İleri kinematik model benzetimi

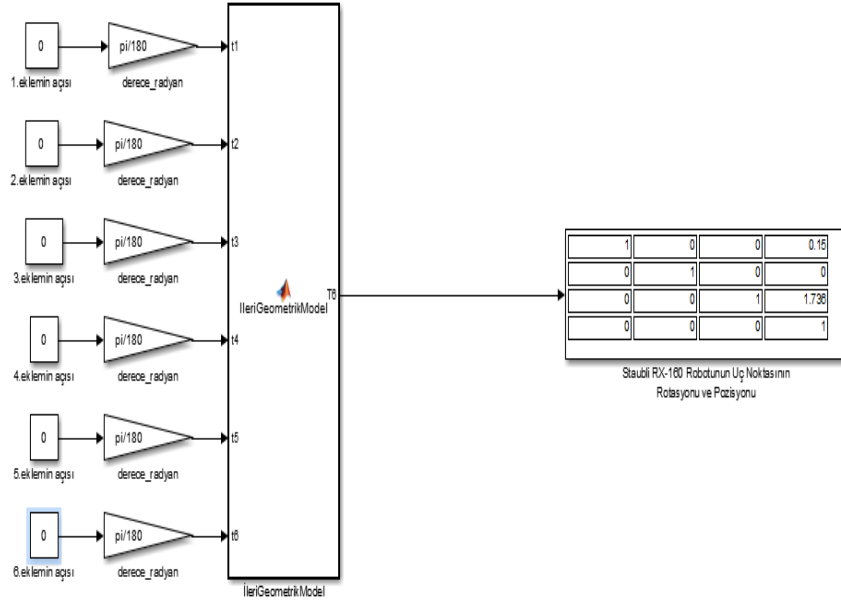
EK C2: İleri kinematik model Matlab kodu

EK D: Dinamik model benzetimi

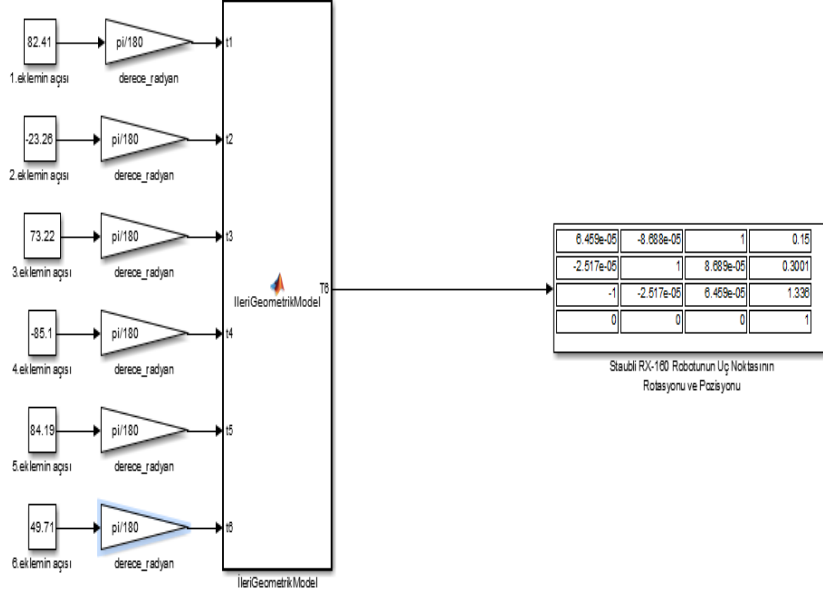
EK E: Konum kontrolü Simulink blok diyagramı

EK F: Sensör kullanılmadan çevre kuvvetinin bulunuşu Simulink blok diyagramı

EK A1 İleri geometrik model benzetimi



Şekil A1.1: İleri geometrik model benzetimi(Robot ilk konumunda)



Şekil A1.2: İleri geometrik model benzetimi(Robot ikinci konumunda).

EK A2: İleri geometrik model Matlab kodu

```
syms t1 t2 t3 t4 t5 t6; % t1=teta1 t2=teta2.....
%Staubli RX-160 manipölatörün ileri geometrik modeli
%alfa1=-pi/2,alfa3=pi/2,alfa4=-pi/2,alfa5=pi/2
%0. ve 1. eksen takımları arasındaki dönme ve öteleme
hareketleri
Rx1=[1 0 0 0; 0 0 1 0;0 -1 0 0;0 0 0 1]
Rz1=[cos(t1) -sin(t1) 0 0;sin(t1) cos(t1) 0 0;0 0 1 0;0 0
0 1]
Tx1=[1 0 0 0.150;0 1 0 0;0 0 1 0;0 0 0 1]
Ty1=[1 0 0 0;0 1 0 0.126;0 0 1 0;0 0 0 1]
Tz1=[1 0 0 0;0 1 0 0;0 0 1 0.176;0 0 0 1]
%1. ve 2. eksen takımları arasındaki dönme ve öteleme
hareketleri
Rz2=[cos(t2) -sin(t2) 0 0;sin(t2) cos(t2) 0 0;0 0 1 0;0 0
0 1]
Ty2=[1 0 0 0;0 1 0 -0.825;0 0 1 0;0 0 0 1]
%2. ve 3. eksen takımları arasındaki dönme ve öteleme
hareketleri
Rx3=[1 0 0 0; 0 0 -1 0;0 1 0 0;0 0 0 1]
Rz3=[cos(t3) -sin(t3) 0 0;sin(t3) cos(t3) 0 0;0 0 1 0;0 0
0 1]
Ty3=[1 0 0 0;0 1 0 -0.105;0 0 1 0;0 0 0 1]
Tz3=[1 0 0 0;0 1 0 0;0 0 1 -0.126;0 0 0 1]
%3. ve 4. eksen takımları arasındaki dönme ve öteleme
hareketleri
Rx4=[1 0 0 0; 0 0 1 0;0 -1 0 0;0 0 0 1]
Rz4=[cos(t4) -sin(t4) 0 0;sin(t4) cos(t4) 0 0;0 0 1 0;0 0
0 1]
Tz4=[1 0 0 0;0 1 0 0;0 0 1 0.520;0 0 0 1]
%4. ve 5. eksen takımları arasındaki dönme ve öteleme
hareketleri
Rx5=[1 0 0 0; 0 0 -1 0;0 1 0 0;0 0 0 1]
Rz5=[cos(t5) -sin(t5) 0 0;sin(t5) cos(t5) 0 0;0 0 1 0;0 0
0 1]
%5. ve 6. eksen takımları arasındaki dönme ve öteleme
hareketleri
Rz6=[cos(t6) -sin(t6) 0 0;sin(t6) cos(t6) 0 0;0 0 1 0;0 0
0 1]
Tz6=[1 0 0 0;0 1 0 0;0 0 1 0.110;0 0 0 1]
%i. ve i+1. eksen takımları arasındaki homojen matrisleri
A1=simplify(Rz1*Tz1*Ty1*Tx1*Rx1)
A2=simplify(Rz2*Ty2)
A3=simplify(Rz3*Tz3*Ty3*Rx3)
A4=simplify(Rz4*Tz4*Rx4)
A5=simplify(Rz5*Rx5)
A6=simplify(Rz6*Tz6)
%i. ve i+1. eksen takımları arasındaki dönme matrisleri
R1=[A1(1,1) A1(1,2) A1(1,3);A1(2,1) A1(2,2)
A1(2,3);A1(3,1) A1(3,2) A1(3,3)]
```

```

R2=[A2(1,1) A2(1,2) A2(1,3);A2(2,1) A2(2,2)
A2(2,3);A2(3,1) A2(3,2) A2(3,3)]
R3=[A3(1,1) A3(1,2) A3(1,3);A3(2,1) A3(2,2)
A3(2,3);A3(3,1) A3(3,2) A3(3,3)]
R4=[A4(1,1) A4(1,2) A4(1,3);A4(2,1) A4(2,2)
A4(2,3);A4(3,1) A4(3,2) A4(3,3)]
R5=[A5(1,1) A5(1,2) A5(1,3);A5(2,1) A5(2,2)
A5(2,3);A5(3,1) A5(3,2) A5(3,3)]
R6=[A6(1,1) A6(1,2) A6(1,3);A6(2,1) A6(2,2)
A6(2,3);A6(3,1) A6(3,2) A6(3,3)]

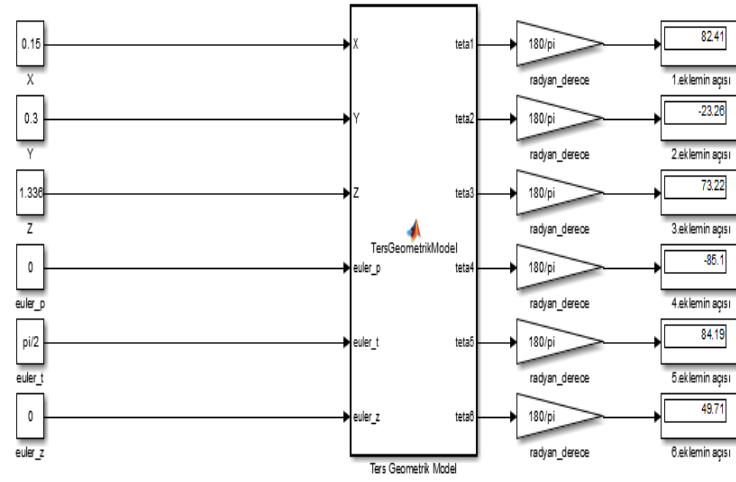
```

```

T1=A1 %0. ve 1. eksen takımları
arasındaki dönüşüm matrisi
T2=A1*A2 %0. ve 2. eksen takımları
arasındaki dönüşüm matrisi
T3=A1*A2*A3 %0. ve 3. eksen takımları
arasındaki dönüşüm matrisi
T4=A1*A2*A3*A4 %0. ve 4. eksen takımları
arasındaki dönüşüm matrisi
T5=A1*A2*A3*A4*A5 %0. ve 5. eksen takımları
arasındaki dönüşüm matrisi
T6=vpa(A1*A2*A3*A4*A5*A6) %0. ve 6. eksen takımları
arasındaki dönüşüm matrisi

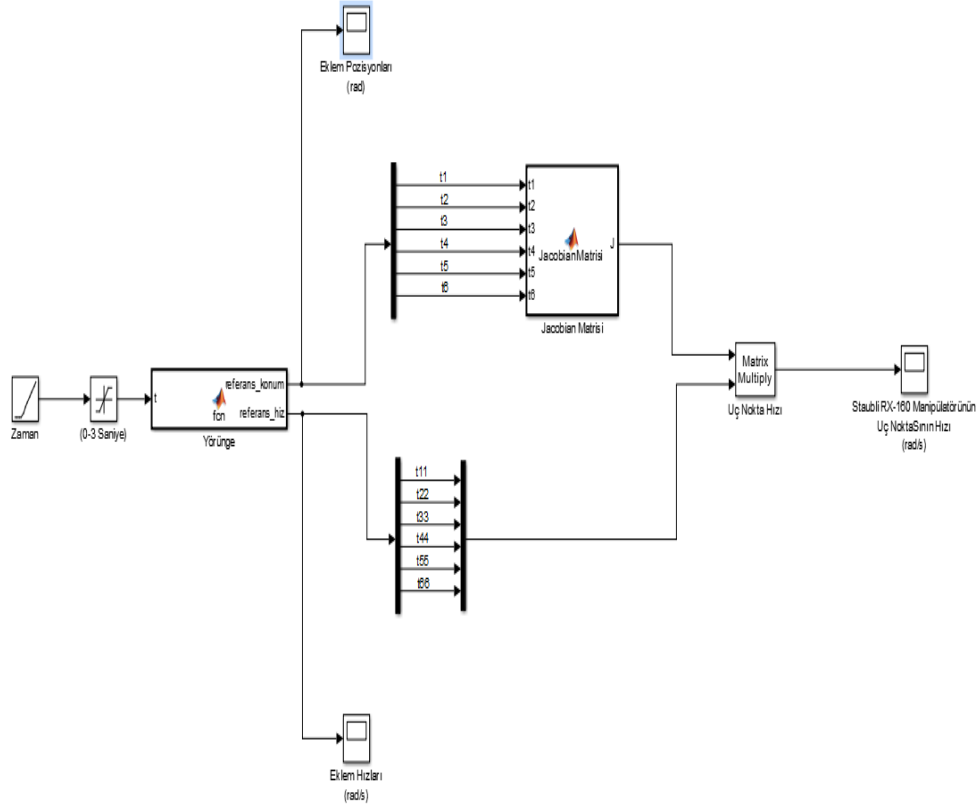
```

EK B: Ters geometrik model benzetimi



Şekil B.1: Ters geometrik model benzetimi.

EK C1: İleri kinematik model benzetimi

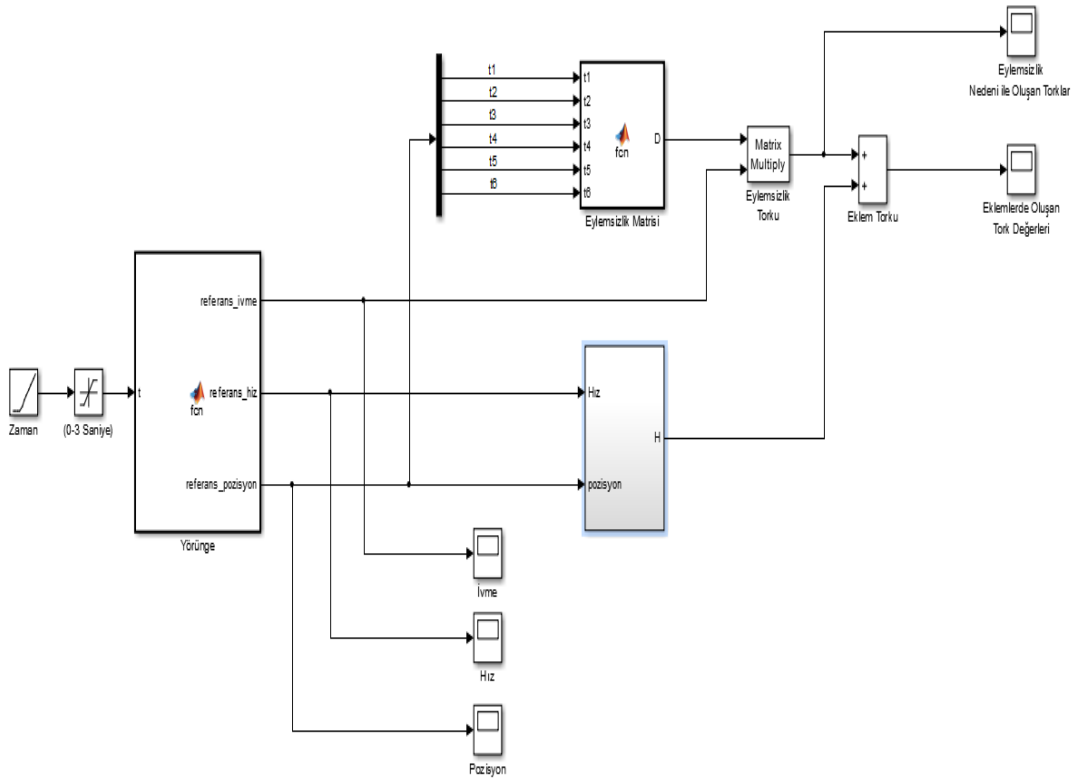


Şekil C1.1: İleri kinematik model benzetimi.

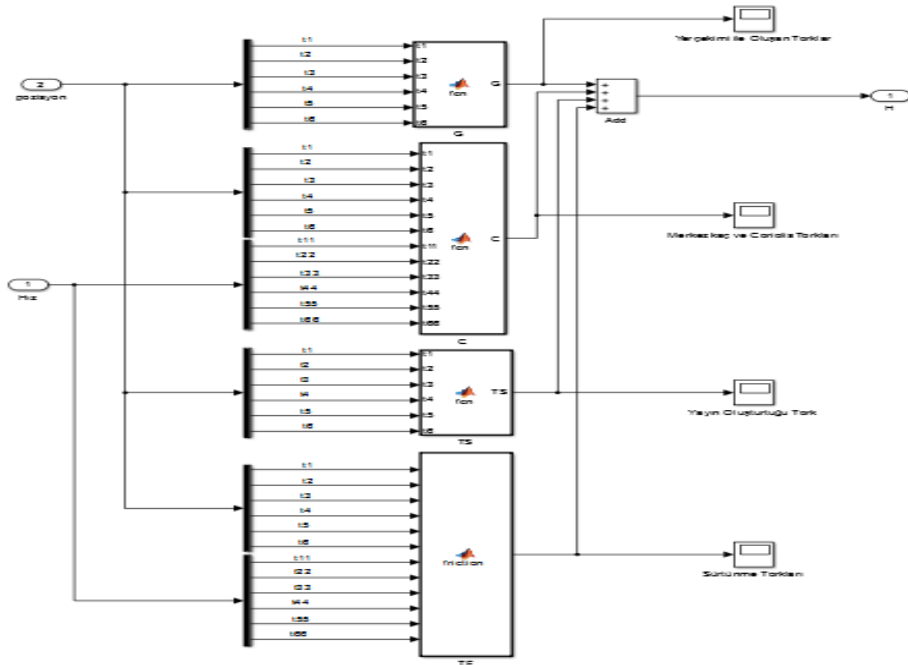
EK C2: İleri kinematik model Matlab kodu

```
% 6*6 Jakobiyen Matrisi
z0=[0;0;1]
z1=T1(1:3,3)
z2=T2(1:3,3)
z3=T3(1:3,3)
z4=T4(1:3,3)
z5=T5(1:3,3)
z6=T6(1:3,3)
o0=[0;0;0]
o1=T1(1:3,4)
o2=T2(1:3,4)
o3=T3(1:3,4)
o4=T4(1:3,4)
o5=T5(1:3,4)
o6=T6(1:3,4)
jw1=z0
jw2=simplify(z1)
jw3=simplify(z2)
jw4=simplify(z3)
jw5=simplify(z4)
jw6=simplify(z5)
jv1=simplify(cross(z0,(o6-o0)))
jv2=simplify(cross(z1,(o6-o1)))
jv3=simplify(cross(z2,(o6-o2)))
jv4=simplify(cross(z3,(o6-o3)))
jv5=simplify(cross(z4,(o6-o4)))
jv6=simplify(cross(z5,(o6-o5)))
J=vpa([jv1 jv2 jv3 jv4 jv5 jv6;jw1 jw2 jw3 jw4 jw5 jw6])
```

EK D: Dinamik model benzetimi

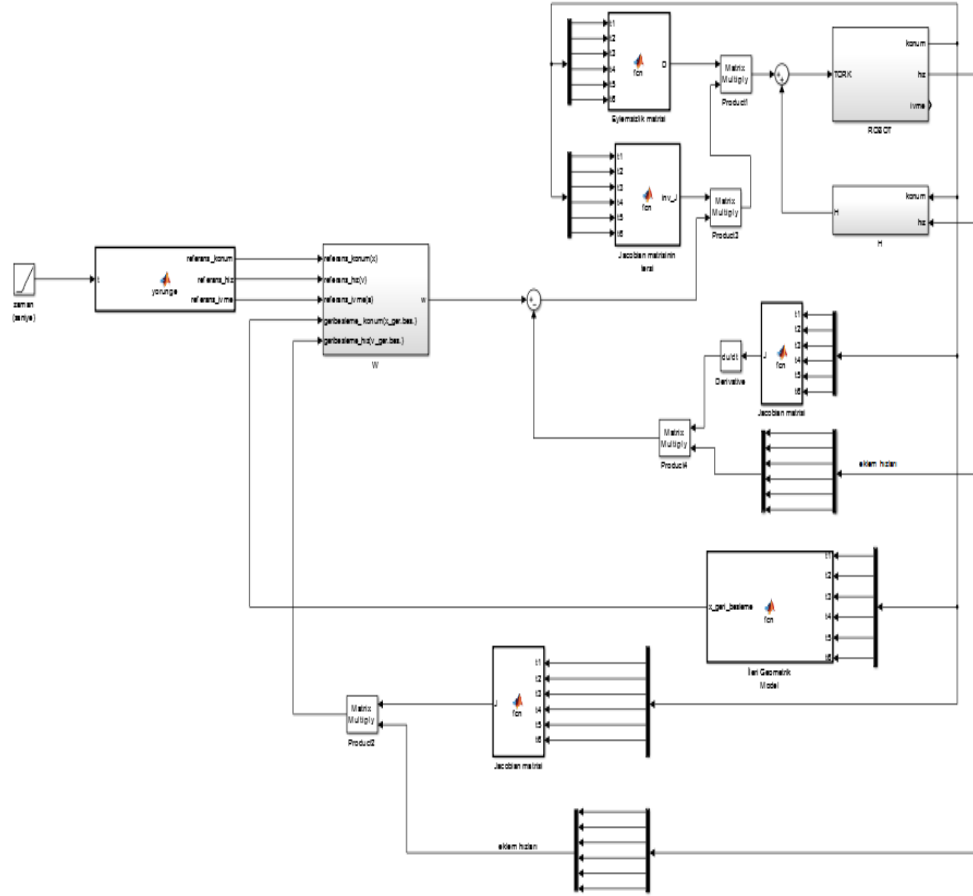


Şekil D.1: Ters dinamik model benzetimi.



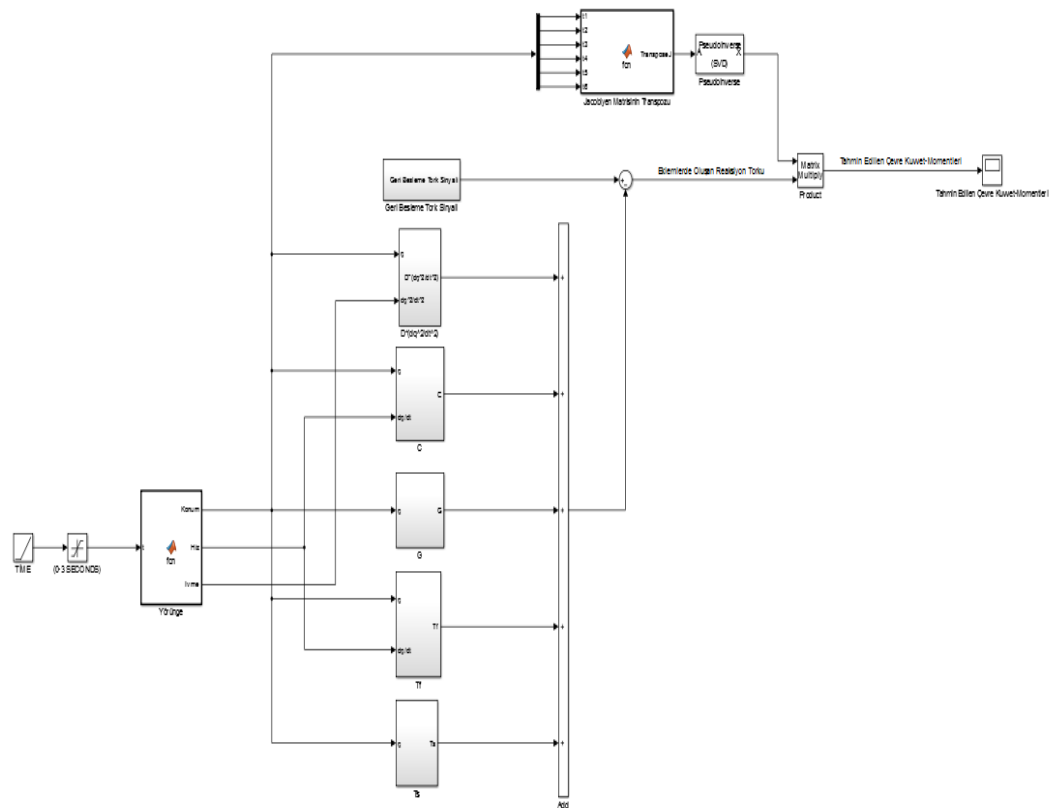
Şekil D.2: H bloğu içerisinde oluşan torklar.

EK E: Konum kontrolü Simulink blok diyagramı



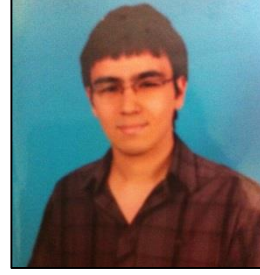
Şekil E.1: Görev uzayında konum kontrol benzetimi.

EK F: Sensör kullanılmadan çevre kuvvetinin bulunuşu Simulink blok diyagramı



Şekil F.1: Sensör kullanılmadan çevre kuvvetinin bulunuşuna ait benzetim.

ÖZGEÇMİŞ



Ad Soyad : Onur TAŞKIRAN

Doğum Yeri ve Tarihi : Kocaeli 15.04.1989

E-Posta : onurtaskiran89@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2012, Yıldız Teknik Üniversitesi, Makine Fakültesi, Makine Mühendisliği

