

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**İNSANSI ROBOT KOLU TASARIMI VE YÖRÜNGE KONTROLÜ**

**YÜKSEK LİSANS TEZİ**

**Musa Özgün GÜLEÇ**

**Makina Mühendisliği Anabilim Dalı**

**Sistem Dinamiği ve Kontrol Programı**

**MAYIS 2014**



**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**İNSANSI ROBOT KOLU TASARIMI VE YÖRÜNGE KONTROLÜ**

**YÜKSEK LİSANS TEZİ**

**Musa Özgün GÜLEÇ**

**(503121612)**

**Makina Mühendisliği Anabilim Dalı**

**Sistem Dinamiği ve Kontrol Programı**

**Tez Danışmanı : Prof. Dr. Şeniz ERTUĞRUL**

**MAYIS 2014**



İTÜ, Fen Bilimleri Enstitüsü'nün 503121612 numaralı Yüksek Lisans / Doktora Öğrencisi **Musa Özgün GÜLEÇ**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**İNSANSI ROBOT KOLU TASARIMI VE YÖRÜNGE KONTROLÜ**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :**      **Prof. Dr. Şeniz ERTUĞRUL**      .....

İstanbul Teknik Üniversitesi

**Jüri Üyeleri :**      **Doç. Dr. Zeki Yağız BAYRAKTAROĞLU**      .....

İstanbul Teknik Üniversitesi

**Yrd. Doç. Dr. Ali Fuat ERGENÇ**      .....

İstanbul Teknik Üniversitesi

**Teslim Tarihi : 5 Mayıs 2014**

**Savunma Tarihi: 26 Mayıs 2014**



## ÖNSÖZ

Günümüzde üzerindeki çalışmaların çok yoğun olduğu ve belki de çağımızın ismini alacağı robot teknolojisi, sanayinin içindeki birçok sektörde karşımıza çıkmaktadır.

Robot teknolojisinin en son hali diyebileceğimiz insansı robotlar ise gerçekten bir insanı taklit edebilecek, iletişim kurabilecek yapay zekaya sahip bir hal almaya başlamıştır. Ülkemizde de bu alanda, gerek özel sektördeki firmalar, gerek üniversiteler çalışmalarını hızlandırmıştır. Yüksek lisans tezimde robot kolu tasarımı ve yörunge kontrolü üzerinde yaptığım çalışmalar ile İTÜ insansı robot projesine, aynı zamanda ülkemde yapılan bu çalışma sayesinde ülkeme, elimden geldiği kadar destek vermeye çalıştım.

Yüksek lisans tez çalışmamda başta değerli danışman hocam Prof. Dr. Şeniz Ertuğrul'a bütün proje boyunca benden yardımlarını hiç esirgemediği için, değerli arkadaşlarım Cem Gürdal, Ufuk Yanbaş ve İlkay Meşeli 'ye projemdeki yardımları için, son olarak sevgili aileme ve nişanlıma bütün çalışmalarım sırasında bana her zaman varlıklarını hissettirdikleri ve destek verdikleri için teşekkür ederim.

Mayıs 2014

Musa Özgün GÜLEÇ  
(Makine Mühendisi)



## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖNSÖZ</b> .....	<b>v</b>
<b>KISALTMALAR</b> .....	<b>ix</b>
<b>ÇİZELGE LİSTESİ</b> .....	<b>xi</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>xiii</b>
<b>ÖZET</b> .....	<b>xvii</b>
<b>SUMMARY</b> .....	<b>xix</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1 Tezin Amacı .....	1
1.2 Literatür Araştırması .....	2
<b>2. ROBOT KOLU TASARIMI</b> .....	<b>9</b>
2.1 Ön Tasarım.....	10
2.2 Tasarım 1.....	11
2.3 Tasarım 2.....	13
<b>3. İLERİ KİNEMATİK VE TERS KİNEMATİK ÇÖZÜMÜ</b> .....	<b>19</b>
3.1 İleri Kinematik Çözümü.....	19
3.2 Ters Kinematik Çözümü .....	21
<b>4. YÖRÜNGE PLANLANMASI</b> .....	<b>33</b>
4.1 Trapez Hız Profili İle Yörünge Planlanması .....	33
4.2 Kübik Hız Profili İle Yörünge Planlanması .....	38
<b>5. YAKLAŞIMLI HESAPLANMIŞ TORK İLE PD KONTROLÜ</b> .....	<b>41</b>
5.1 Jacobian Matrisinin Bulunuşu .....	44
5.2 SimMechanics™ İle Simulink® Modelinin Oluşturulması .....	45
5.3 İlave Yüksüz,Birinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar.....	50
5.4 İlave Yüksüz,Birinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar.....	52
5.5 İlave Yükle, Birinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar .....	55
5.6 İlave Yükle, Birinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar .....	57
5.7 İlave Yüksüz, İkinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar .....	59
5.8 İlave Yüksüz, İkinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar .....	62
5.9 İlave Yükle, İkinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar.....	64
5.10 İlave Yükle, İkinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar.....	67
5.11 Ön Tasarım ile Karşılaştırma Benzetimi.....	69

<b>6. SONUÇ VE ÖNERİLER.....</b>	<b>73</b>
<b>KAYNAKÇA .....</b>	<b>75</b>
<b>EKLER.....</b>	<b>77</b>
<b>ÖZGEÇMİŞ.....</b>	<b>111</b>

## **KISALTMALAR**

**DH** : Denavit- Hartenberg

**mNm** : mili newton metre

**kg** : kilogram

**mm** : milimetre

**V** : voltaj

**A** : amper

**s** : saniye



## ÇİZELGE LİSTESİ

Çizelge 2.1 : Ön tasarım motor ve dişli mekanizmalarının özellikleri.....	11
Çizelge 2.2: Tasarım 1 motor ve dişli mekanizmalarının özellikleri. <sup>[11][12]</sup> .....	12
Çizelge 2.3: Tasarım 2 kullanılan malzeme listesi. <sup>[11][12]</sup> .....	16
Çizelge 2.4 : Tasarım 2 motor ve dişli mekanizmalarının özellikleri. <sup>[11][12]</sup> .....	17
Çizelge 2.5 : Robot kolu eklem ağırlıkları. ....	17
Çizelge 3.1: Robot kolu DH parametreleri. ....	20
Çizelge 4.1: Örnek trapez hız profili için hareket çizelgesi.....	34
Çizelge 4.2 : Eklemlerin maksimum hız ve ivme değerleri.....	35
Çizelge 5.1 : Seçilen 2 örnek hareketin değerleri. ....	48
Çizelge 5.2 : Motorların $K_t(i)$ tork sabitleri <sup>[11]</sup> .....	50



## ŞEKİL LİSTESİ

### Sayfa

Şekil 1.1 : İlk insansı robot çalışması (1868) .....	2
Şekil 1.2 : ASIMO insansı robotu. ....	3
Şekil 1.3 : Rollin' Justin insansı robotu. ....	4
Şekil 1.4 : ATLAS insansı robotu. ....	4
Şekil 1.5 : SCHAFT insansı robotu. ....	5
Şekil 1.6 : PETMAN insansı robotu. ....	6
Şekil 1.7 : ROMEO insansı robotu .....	6
Şekil 1.8 : HRP-4 insansı robotu. ....	7
Şekil 1.9 : MOTOMAN çift kol robotu. ....	8
Şekil 1.10 : FRIDA çift kol robotu. ....	8
Şekil 2.1 : 6 serbestlik dereceli robot kolu motor dizilimi. ....	9
Şekil 2.2 : İlkay Meşeli tarafından yapılan ön tasarım. ....	10
Şekil 2.3 : Tasarım 1 katı model görünümü. ....	12
Şekil 2.4 : Tasarım 2 görünüm, motor, Harmonic Drive <sup>®</sup> , algılayıcı dizilimi. ....	14
Şekil 2.5 : Tasarım 2 perspektif görüntüleri. ....	15
Şekil 2.6 : Tasarım 2 ön ve sağ görünüm. ....	15
Şekil 3.1 : DH kıstasına göre motor merkez ve koordinatları. ....	20
Şekil 3.2 : $O_4$ noktasının $O_2$ noktasına göre hareket kabiliyeti. ....	23
Şekil 3.3 : $O_4$ noktasının $O_5$ noktasına göre hareket kabiliyeti. ....	24
Şekil 3.4 : $r_1$ ve $r_2$ yarıçaplı 2 çemberin kesişimi. ....	25
Şekil 3.5 : $\theta_1$ ve $\theta_2$ açılarının geometrik gösterimi. ....	30
Şekil 4.1 : Trapez hız profilinde eklemlerin konum-zaman grafiği örneği. ....	34
Şekil 4.2 : Trapez hız profilinde eklemlerin hız-zaman grafiği örneği. ....	34
Şekil 4.3 : Trapez hız profilinde eklemlerin ivme-zaman grafiği örneği. ....	35
Şekil 4.4 : Kübik hız profilinde eklemlerin konum-zaman grafiği örneği. ....	39
Şekil 4.5 : Kübik hız profilinde eklemlerin hız-zaman grafiği örneği. ....	39
Şekil 4.6 : Kübik hız profilinde eklemlerin ivme-zaman grafiği örneği. ....	40
Şekil 5.1 : Ters dinamik model çalışma mantığı. <sup>[13]</sup> .....	44
Şekil 5.2 : Simulink <sup>®</sup> modeli çalışma prensibi. ....	46
Şekil 5.3 : Birinci hareketin görüntüleri. ....	48
Şekil 5.4 : İkinci hareketin görüntüleri. ....	49
Şekil 5.5 : Hesaplanmış eklem tork değerleri (1. hareket, kübik, yüksüz). ....	50
Şekil 5.6 : Gerçek - referans eklem açısız konumları (1. hareket, kübik, yüksüz). ...	51
Şekil 5.7 : Gerçek - referans eklem açısız hızları (1. hareket, kübik, yüksüz). ....	51
Şekil 5.8 : Hesaplanmış motor tork değerleri (1. hareket, kübik, yüksüz). ....	51
Şekil 5.9 : Motor akım değerleri (1. hareket, kübik, yüksüz). ....	52
Şekil 5.10 : Gerçek - referans uç konum koordinatları (1. hareket, kübik, yüksüz)..	52
Şekil 5.11 : Hesaplanmış eklem tork değerleri (1. hareket, trapez, yüksüz). ....	53
Şekil 5.12 : Gerçek - referans eklem açısız konumları (1. hareket, trapez, yüksüz). 53	
Şekil 5.13 : Gerçek - referans eklem açısız hızları (1. hareket, trapez, yüksüz). ....	53
Şekil 5.14 : Hesaplanmış motor tork değerleri (1. hareket, trapez, yüksüz). ....	54

<b>Şekil 5.15</b> : Motor akım değerleri (1. hareket, trapez, yüksüz). .....	54
<b>Şekil 5.16</b> : Gerçek - referans uç konum koordinatları (1. hareket, trapez, yüksüz). .....	54
<b>Şekil 5.17</b> : Hesaplanmış eklem tork değerleri (1. hareket, kübik, 3 kg yük). .....	55
<b>Şekil 5.18</b> : Gerçek-referans eklem açısal konumları (1. hareket, kübik, 3 kg yük). .....	55
<b>Şekil 5.19</b> : Gerçek - referans eklem açısal hızları (1. hareket, kübik, 3 kg yük). .....	56
<b>Şekil 5.20</b> : Hesaplanmış motor tork değerleri (1. hareket, kübik, 3 kg yük). .....	56
<b>Şekil 5.21</b> : Motor akım değerleri (1. hareket, kübik, 3 kg yük). .....	56
<b>Şekil 5.22</b> : Gerçek - referans uç konum koordinatları (1. hareket, kübik, 3 kg yük). .....	57
<b>Şekil 5.23</b> : Hesaplanmış eklem tork değerleri (1. hareket, trapez, 3 kg yük). .....	57
<b>Şekil 5.24</b> : Gerçek-referans eklem açısal konumları (1. hareket, trapez, 3 kg yük). .....	58
<b>Şekil 5.25</b> : Gerçek - referans eklem açısal hızları (1. hareket, trapez, 3 kg yük). .....	58
<b>Şekil 5.26</b> : Hesaplanmış motor tork değerleri (1. hareket, trapez, 3 kg yük). .....	58
<b>Şekil 5.27</b> : Motor akım değerleri (1. hareket, trapez, 3 kg yük). .....	59
<b>Şekil 5.28</b> : Gerçek - referans uç konum koordinatları (1. hareket, trapez, 3 kg yük). .....	59
<b>Şekil 5.29</b> : Hesaplanmış eklem tork değerleri (2. hareket, kübik, yüksüz). .....	60
<b>Şekil 5.30</b> : Gerçek-referans eklem açısal konumları (2. hareket, kübik, yüksüz). .....	60
<b>Şekil 5.31</b> : Gerçek - referans eklem açısal hızları (2. hareket, kübik, yüksüz). .....	61
<b>Şekil 5.32</b> : Hesaplanmış motor tork değerleri (2. hareket, kübik, yüksüz). .....	61
<b>Şekil 5.33</b> : Motor akım değerleri (2. hareket, kübik, yüksüz). .....	61
<b>Şekil 5.34</b> : Gerçek - referans uç konum koordinatları (2. hareket, kübik, yüksüz). .....	62
<b>Şekil 5.35</b> : Hesaplanmış eklem tork değerleri (2. hareket, trapez, yüksüz). .....	62
<b>Şekil 5.36</b> : Gerçek-referans eklem açısal konumları (2. hareket, trapez, yüksüz). .....	63
<b>Şekil 5.37</b> : Gerçek - referans eklem açısal hızları (2. hareket, trapez, yüksüz). .....	63
<b>Şekil 5.38</b> : Hesaplanmış motor tork değerleri (2. hareket, trapez, yüksüz). .....	63
<b>Şekil 5.39</b> : Motor akım değerleri (2. hareket, trapez, yüksüz). .....	64
<b>Şekil 5.40</b> : Gerçek - referans uç konum koordinatları (2. hareket, trapez, yüksüz). .....	64
<b>Şekil 5.41</b> : Hesaplanmış eklem tork değerleri (2. hareket, kübik, 3 kg yük). .....	65
<b>Şekil 5.42</b> : Gerçek-referans eklem açısal konumları (2. hareket, kübik, 3 kg yük). .....	65
<b>Şekil 5.43</b> : Gerçek - referans eklem açısal hızları (2. hareket, kübik, 3 kg yük). .....	65
<b>Şekil 5.44</b> : Hesaplanmış motor tork değerleri (2. hareket, kübik, 3 kg yük). .....	66
<b>Şekil 5.45</b> : Motor akım değerleri (2. hareket, kübik, 3 kg yük). .....	66
<b>Şekil 5.46</b> : Gerçek-referans uç konum koordinatları (2. hareket, kübik, 3 kg yük). .....	66
<b>Şekil 5.47</b> : Hesaplanmış eklem tork değerleri (2. hareket, trapez, 3 kg yük). .....	67
<b>Şekil 5.48</b> : Gerçek-referans eklem açısal konumları (2. hareket, trapez, 3 kg yük). .....	67
<b>Şekil 5.49</b> : Gerçek - referans eklem açısal hızları (2. hareket, trapez, 3 kg yük). .....	68
<b>Şekil 5.50</b> : Hesaplanmış motor tork değerleri (2. hareket, trapez, 3 kg yük). .....	68
<b>Şekil 5.51</b> : Motor akım değerleri (2. hareket, trapez, 3 kg yük). .....	68
<b>Şekil 5.52</b> : Gerçek-referans uç konum koordinatları (2. hareket, trapez, 3 kg yük). .....	69
<b>Şekil 5.53</b> : Hesaplanmış eklem tork değerleri. .....	69
<b>Şekil 5.54</b> : Gerçek-referans eklem açısal konumları. .....	70
<b>Şekil 5.55</b> : Gerçek - referans eklem açısal hızları. .....	70
<b>Şekil 5.56</b> : Hesaplanmış motor tork değerleri. .....	70
<b>Şekil 5.57</b> : Motor akım değerleri. .....	71
<b>Şekil 5.58</b> : Gerçek-referans uç konum koordinatları. .....	71

<b>Şekil A.1</b> : EC60 motor katalođu.....	78
<b>Şekil A.2</b> : EC90 motor katalođu.....	79
<b>Şekil B.1</b> : Kullanılan Harmonic Drive® diřli mekanizmalarının katalođu.....	80



# İNSANSI ROBOT KOLU TASARIMI VE YÖRÜNGE KONTROLÜ

## ÖZET

Sürekli gelişmekte olan robot teknolojisi, ülkemizde de bu alandaki çalışmaları hızlandırmıştır. İstanbul Teknik Üniversitesi, Makina Fakültesi tarafından yürütülen insansı robot projesi de bu çalışmalara bir örnektir. Bu tez çalışmasında insansı robot projesinin sadece kol kısmının tasarımı ve yörünge kontrolü çalışılmıştır.

İlk olarak SolidWorks® programı ile robot kolunun tasarımı yapılmıştır. Bu sayede kolun toplam ağırlığı, ölçüleri, eylemsizlik momenti değerleri ve montaj şekli belirlenmiştir. Daha sonra ise MATLAB® ortamında robot kolunun ileri kinematik ve ters kinematik çözümleri analitik olarak yapılmıştır. Böylece, istenilen uç konuma istenilen yönelimde varmak için gerekli eklem açıları bulunmuştur.

Tezin ilerleyen kısmında ise ilk konum ve son konum eklem açısal değerlerine bağlı olarak yörünge planlanması yapılmıştır. Çalışmada 2 ayrı yörünge seçilmiştir. Bunlardan ilki kübik hız profili veren, diğeri ise trapez hız profili veren yörüngelerdir. Yörünge planlaması sonucu elde edilen anlık açısal konum, açısal hız ve açısal ivmeler, ters dinamik kontrolde kullanılmıştır.

Yörünge planlamasından sonra ise SolidWorks® programında elde edilmiş tasarım SimMechanics™ kullanılarak MATLAB® Simulink® ortamına aktarılmıştır. Böylece tasarımın bütün dinamik ve geometrik özellikleri MATLAB® ortamına aktarılmıştır. Bundan sonraki adım hareket sırasında gereken ortalama tork değerlerini hesaplamaktır. Bu hesaplamadan sonra, yaklaşımlı hesaplanmış tork değerleri, hesaplanmış referans yörüngeler kullanılarak kapalı çevrim PD kontrol yapılmıştır. Benzetim sonuçları 2 ayrı hareket için kübik ve trapez olmak üzere 2 ayrı yörünge ile elde edilmiştir. Aynı benzetimler kol ucuna yük eklenerek tekrar yapılmıştır. Böylece, benzetimler sonucu elde edilmiş gerekli tork miktarlarını, tasarımda seçilen motor ve dişli mekanizmalarının karşılayıp karşılayamadıkları belirlenmiştir. Son olarak sonuçlar tartışılmıştır.



## **DESIGN OF A HUMANOID ROBOT ARM AND TRAJECTORY CONTROL**

### **SUMMARY**

The robot technology has been continuously developing. Hence, the research in robotics also increases in our country. Since, they can be used in many kinds of industries such as mass production, defense industry, humanoid robots gain wide popularity. Besides, humanoid robots can also be used in social environment for different reasons. The improving technology also contributes the rapid development of robots in general.

As a contribution to the field of robotics, a humanoid robot project has been started by Istanbul Technical University, Mechanical Engineering Department. The purpose of the robot is to work in a manufacturing environment such as pick and place operation, simple assembly or using hand tools.

In this theses, only the humanoid robot arm design and its trajectory control have been studied. As a first step of the design, number of joints, placement of motors, dimensions, selection of motors and power transmission elements must be considered. Since the motors bear themselves in robot arm, their weights must be low and torque values must be high enough. Therefore, it is investigated to find motors which have smallest ratio between the weight and torque values. Besides, in order to obtain a robot arm that has a good appearance, the flat type brushless motors are investigated. Furthermore, it is desired to select motors which have own encoder because, some of the links are not suitable for mounting an additional encoder, so it causes another elimination. Under all these restrictions, EC90 and EC60 motors from Maxon<sup>®</sup> company have been selected.

After determining the motors, it is desired to select a appropriate power transmission which must have low weight and high reduction ratio. Besides, the selected reducers should not cause backlash in order to guarantee the position accuracy. Then, it has been decided using the harmonic drive reducers. It is determined that the CPL series harmonic drives are lightest ones with respect to their maximum torque values. Therefore, the CPL 14A and CPL 17A type harmonic drives have been selected.

After making decision about the proper motor and reducer groups, the design of the robot arm is obtained in the SolidWorks<sup>®</sup> program. The lengths of the links are designed according to the human arm. The necessary bearings, nuts, screws and fittings are designed. Therefore, the total weight, lengths, inertia values of each link and the mate properties are determined.

Then, the forward kinematic and inverse kinematic problems are solved analytically in MATLAB<sup>®</sup> environment. While determining the forward kinematic the Denavit-

Hartenberg parameters are used. Since the inverse kinematic problem cannot be found by using the transformation matrices, a different geometric approach is used. In this method, firstly, the possible elbow positions which can put the end effector into the desired position are determined. Then, the necessary elbow orientation which guarantee the desired orientation of the end effector is selected between possible elbow positions. It means that according to the desired total transformation matrix the desired transformation matrix until elbow is known. Then, by using symbolic transformation matrices, the inverse kinematic problem can be solved. Therefore, the necessary angular positions of the links are found in order to put the end effector into the desired position with desired orientation.

In the next part of the theses, the trajectory planning is made with respect to initial and final angular positions of the links. The aim in the trajectory planning is to determine the instantaneous angular positions, velocities and accelerations during the movement. These values are also used as reference values in inverse dynamic control.

This study contains 2 different trajectory types. The first one has cubic velocity profile while second has trapezoidal. The cubic velocity profile provides a continuous acceleration and a smooth movement. Since, the acceleration is continuous, the velocity and position profiles are also continuous. Besides, due to the necessary torque values are directly proportional to the acceleration, they are also be continuous and there is no sudden change in the torque values. On the other hand, the trapezoidal velocity profile guarantees the minimum travelling time. However, the acceleration profiles contain discontinuity regions and it causes sudden change in the torque values. Then, the necessary torque values become higher in trapezoidal velocity profile.

After trajectory planning, the robot arm design, obtained in SolidWorks<sup>®</sup>, is transferred into MATLAB<sup>®</sup> Simulink<sup>®</sup> environment by using SimMechanics<sup>™</sup> addition of the SolidWorks<sup>®</sup>. Therefore, MATLAB<sup>®</sup> obtains all dynamic and geometric properties of robot arm. Then, the approximate torque values which is necessary to move the robot must be computed. In order to obtain the approximate torque values, the Lagrangian energy method is used. In this method, the torques caused by potential energy, kinetic energy and coriolis or centrifugal effect are computed separately. Therefore, while the closed form control mechanism works, the necessary torque values are computed instantaneously. However, in this study, the centrifugal effect and frictional lost are neglected. In order to minimize errors a PD type compensator has been used.

The closed form PD control mechanism is created by using approximate computed torque and computed reference trajectory. In other words, the approximate torque values, angular position and velocity errors are sent to PD controller.

The simulation results are obtained for 2 different sample motions by using 2 different trajectories which are cubic and trapezoidal. The same simulations are repeated with extra weight which is added to the end effector. In order to determine if the necessary torque values, obtained in simulation, are in the working range of both motors and reducers, one of the sample motions is selected as the most difficult motion. Therefore, maximum necessary torque value can be observed.

After simulations, figures of the computed torque values of links, the actual and reference angular positions of links, the actual and reference angular velocity values of links, computed motor torque values, computed motor current values and actual and reference end effector position are obtained. While determining the motor torque values, the inertia effects of the harmonic drive reducers are taken into account. Besides, the necessary current values are also computed by using the motors torque constant values. In addition, in order to make a comparison between current design and the previous design, an additional simulation under the same conditions has been run. The difference of the performance between the previous and current design has been explained.

Finally, according to the simulation results, it has been observed that selected motors and harmonic drives are properly chosen. The all torque values are in the working range of both motors and reducers. Especially, in the cubic velocity profile, the designed robot can move either heavier loads, grippers or hand tools.



## 1. GİRİŞ

İlk başlarda sadece araştırma ve geliştirme konusu olan robot teknolojisi, günümüzde gelişen teknoloji ve ihtiyaçlar doğrultusunda sanayinin bir parçası olmuş ve gelişimine daha fazla ivme kazandırmıştır. Robot teknolojisinin de bir parçası olan insansı robotlar ise sadece hareket eden, dans eden bir sosyal robottan çıkmış, sanayide çalışabilen işçi robot veya askeri amaçlı kullanılabilen bir insansı robot halini almaya başlamıştır.

Bu tez çalışmasında, İTÜ insansı robot projesi için kol prototipi geliştirme çalışmaları yer almaktadır. Öncelikle tezin amacından bahsedilmekte, sonrasında ise literatürde bulunan insansı robot ve robot kolu çalışmaları incelenmektedir. Tezin devamında SolidWorks® programı aracılığı ile elde edilen robot kolu tasarımı sunulmaktadır. Sonrasında ise robot kolunun ileri kinematik ve ters kinematik çözümünden bahsedilmektedir. Daha sonrasında ise yörünge planlanması anlatılmaktadır. MATLAB® Simulink® ortamında yaklaşımlı hesaplanmış tork ile elde edilen kapalı çevrim PD kontrolü incelendikten sonra tez çalışması, sonuç ve öneriler kısmı ile son bulmaktadır.

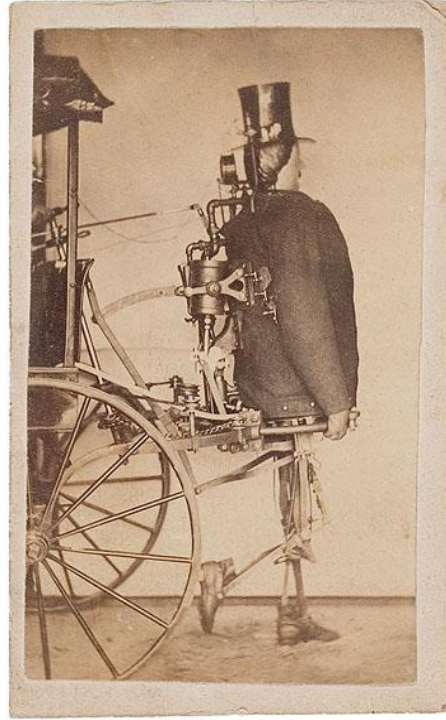
### 1.1 Tezin Amacı

Bu tez kapsamında, üzerindeki çalışmaların gün geçtikçe yoğunlaştığı insansı robotun kol kısmının tasarımı ve yörünge kontrolü çalışılmıştır. Minimum ağırlık ile maksimum verimliliği elde etmek her robot tasarımında olduğu gibi en önemli amaçtır. Bu sebeple, mümkün olduğunca dayanıklı ama hafif malzemeler, kütlesi ve boyutu küçük ancak yüksek tork üretebilen motorlar ve minimum ağırlıkta yüksek tork değerine dayanabilen aynı zamanda yüksek çevrim oranlarına sahip dişli mekanizmaları seçilmeye çalışılmıştır. Bir diğer amaç ise, bütün bunlar yapılırken konum hassasiyetini sağlayabilmek için boşluksuz dişli mekanizmaları ile çalışılmıştır. Daha sonrasında yapılan tasarımın istenilen performansı sağlayıp

sağlamadığı MATLAB® Simulink® ortamında yapılan benzetimler ile incelenmiş ve tasarımın doğruluğu gözden geçirilmiştir.

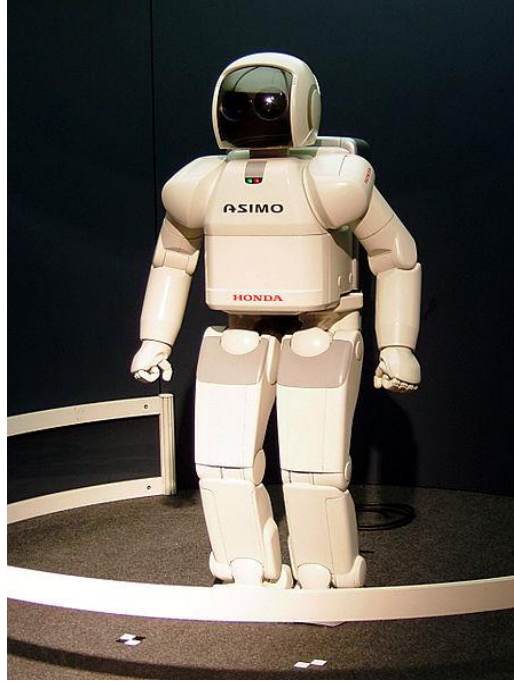
## 1.2 Literatür Araştırması

İnsansı robotlar 100 yılı aşkın bir süredir insanların geliştirmeye çalıştığı ve günümüzde de çok hızlı olarak gelişmeye devam eden bir çalışmadır. Tarihte ilk insansı robot olarak kabul edilebilecek çalışma, Şekil 1.1 'de görülen, Zadoc P. Dederick tarafından 1868 yılında tasarlanmış olan insansı robottur.<sup>[1]</sup> Bu robotun tasarımındaki amaç, at arabasını bir robotun çekmesini sağlamaktır. O zamanki teknolojik durumu düşünürsek, kullanılan eyleyiciler buhar motorlarıdır.



**Şekil 1.1** :İlk insansı robot çalışması (1868)

Günümüzdeki robot çalışmaları incelenirse, karşımıza ilk çıkan model ASIMO olacaktır. ASIMO 1986 yılından bu yana bir Japon firması olan Honda tarafından sürekli geliştirilmektedir. En son 11. tasarımlarını sundukları ASIMO 'nun boyu 130 cm, ağırlığı 48 kg 'dır, bunun yanında 57 adet serbestlik derecesi vardır ve 9 km/saat hızla koşabilmektedir.<sup>[2]</sup> Şekil 1.2 ASIMO 'nun görünümüdür.



**Şekil 1.2 :** ASIMO insansı robotu.

Bir başka insansı robot çalışması ise Şekil 1.3 'de görülen Almanya merkezli DLR firmasına ait olan Rollin' Justin robotudur. Rollin' Justin ortalama 200 kg ağırlığında gövdenin üst kısmında toplamda 43 tane serbestlik derecesi olan bir robottur. Serbestlik derecesinin ASIMO' dan daha az olmasının sebebi bu robotta yürüme aksamı olmaması, onun yerine tekerleklerle hareketin verilmesidir. Ancak bu robot yapabildiği işler ve uygulayabildiği kuvvetler açısından ASIMO 'ya göre çok daha başarılıdır. Ortalama 20 kg yük taşıyabilmekte, ve çalışma alanı 1,7 metre çapında bir kürenin içerisindeydir.<sup>[3]</sup>

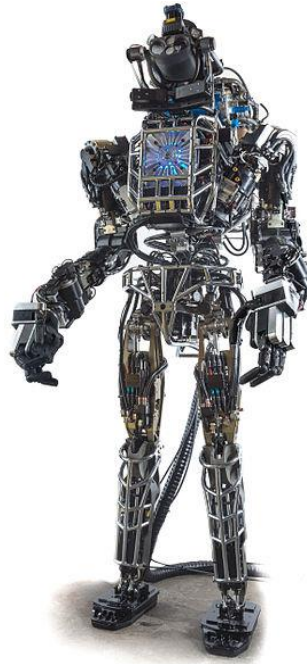
Bir başka insansı robot çalışması ise bir Amerikan firması olan Boston Dynamics tarafından geliştirilen ATLAS robotudur. Robot 1.8 m boyunda, 28 serbestlik dereceli, 150 kg ağırlığındadır. Acil servislerde ve kurtarma timlerinde kullanılması planlanan robotun görünümü Şekil 1.4 'deki gibidir.<sup>[4]</sup>

Literatür araştırması kısmında gösterilecek olan bir başka insansı robot örneği ise, Japonya merkezli SHAFT Inc. firması tarafından tasarlanan SCHAFT isimindeki robottur. 20 Aralık 2013 tarihinde düzenlenen Darpa Rescue Challenge yarışmasında birincilik kazanmış olan bu robot, patentli olan yeni tip motorları sayesinde çok yüksek tork değerlerine çıkabilmektedir. Gerçekten güç konusunda bir fark yaratmış

olan SCHAFT, 1480 mm yüksekliğinde ve 95 kg ağırlığındadır. Şekil 1.5 SCHAFT insansı robotunun dış görünümünü sunmaktadır. SCHAFT 'ın bu kadar büyük bileklerinin olması, aslında iş yapacak robotların bileklerinin daha büyük olması gerektiğini göstermektedir. Bu sebeple, çalışmada bilek kısmının güçlü olmasına özen gösterildi.<sup>[5]</sup>



**Şekil 1.3 :** Rollin' Justin insansı robotu.



**Şekil 1.4 :** ATLAS insansı robotu.



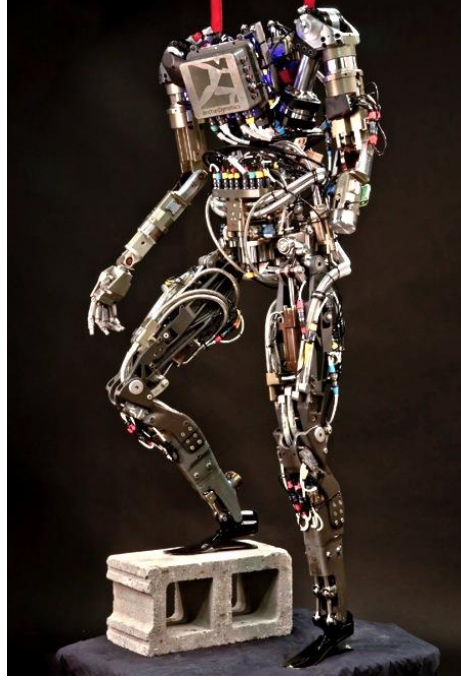
**Şekil 1.5 :** SCHAFT insansı robotu.

Literatürde üzerinde çalışmaların hala devam ettiği bir başka insansı robot ise yine Boston Dynamics firması tarafından Amerika'da geliştirilen PETMAN isimli robottur. Daha çok askeri amaçlı tasarlanmış olan bu insansı robot, boyutları açısından da diğerlerinden biraz daha iridir. Yaklaşık olarak 80 kg ağırlığında olan PETMAN 1.75 m uzunluğundadır. Hız olarak ise maksimum 7 km/saat hıza ulaşabilmektedir.<sup>[6]</sup> PETMAN 'in görünümü Şekil 1.6 'da gösterilmektedir.

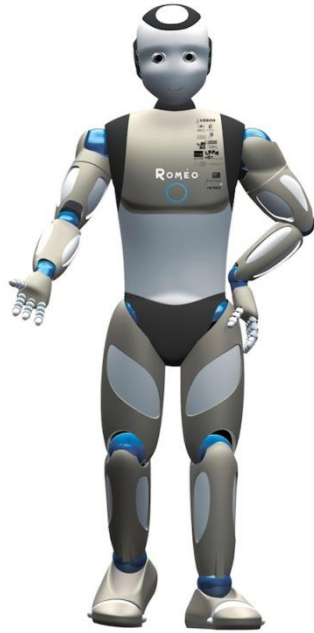
Bir başka insansı robot ise merkezi Fransa'nın Paris şehrinde olan Aldebaran Robotics firmasının geliştirdiği ROMEO isimli insansı robottur. Yaklaşık 40 kg ağırlığında ve 1,40 m boyunda olan ROMEO 'nun toplamda 37 adet serbestlik derecesi mevcuttur. ROMEO 'nun asıl özelliği ise insanlara günlük hayatta yardım edebilecek, asistanlık yapabilecek daha sosyal bir robot olmasıdır. Evcil hayvanları besleyebilir, çöpleri atabilir veya bir tepsi taşıyabilir.<sup>[7]</sup> Şekil 1.7 'de ROMEO 'nun dış görünümü sunulmuştur.

Literatürde insansı robotlar arasında son bahsedilecek olan robot ise HRP-4 robotudur. KAWADA Industries isimli bir Japon firması tarafından geliştirilen bu insansı robot 39 kg ağırlığında ve 151 cm boyundadır. 34 adet serbestlik derecesi

vardır ve tek kol ile taşıyabileceği maksimum yük 0,5 kg kadardır.<sup>[8]</sup> Bu sebeple çok fazla güçlü olduğu söylenemez. Şekil 1.8 'de HRP-4 'ün resmi görülmektedir.



Şekil 1.6 : PETMAN insansı robotu.



Şekil 1.7 : ROMEO insansı robotu



**Şekil 1.8 : HRP-4 insansı robotu.**

Literatürde üzerinde çalışmaların devam ettiği birde çift kol çalışmaları mevcuttur. Yalnızca kol kısımlarından ibaret olan bu robotlar sabit bir yerde iş yapmak, montaj yapmak veya herhangi bir malzemeyi alıp bir başka yere koymak için kullanılırlar. Belki de sanayide kullanımı daha fazla olan bir robottur. Çünkü bazı yerlerde gerçekten yürüme aksamına ihtiyaç duyulmamaktadır.

İlk olarak verilebilecek örnek çift kol çalışması ise bir Japon firması olan Yaskawa Robotics firması tarafından geliştirilen MOTOMAN çift kol robotudur. MOTOMAN çalışma alanı yatayda her kol için 765 mm, dikeyde ise 640 mm olan bir robottur. Toplamda 13 adet serbestlik derecesi vardır. Bunların 12 adeti toplam iki kolda, 1 adette gövdededir. Gerçekten ölçüleri insansı kola göre daha büyük ve çalışma alanı geniştir. Tam anlamıyla iş yapabilecek iki kol olarak düşünülmüştür. Ağırlığı ise ortalama 240 kg dır. Görüldüğü gibi, sadece iki adet koldan ibaret olsa bile çok ağırdır. Fakat bununla orantılı her kol 20 kg yük taşıyabilmektedir.<sup>[9]</sup> Bu değer ise insansı robotlarda ulaşılması zor bir değerdir. Şekil 1.9 MOTOMAN çift kol robotunun görünümünü sunmaktadır.

Bu tez çalışmasında literatür araştırması olarak incelenen son robot ise bir Alman firması olan ABB tarafından geliştirilen FRIDA isimli çift kol çalışan robottur. FRIDA 'nın MOTOMAN 'dan farkı ise 14 adet serbestlik derecesi olmasıdır. Yani her kolda 7 adet serbestlik derecesi mevcuttur ve gövde sabittir, bu ise robot kolunda

fazla serbestlik derecesi anlamına gelmektedir. Fakat bu fazlalık durumu kolun ilave hareket kabiliyeti kazanmasını sağlamış, çalışma alanını arttırmıştır.<sup>[10]</sup> FRIDA çift kol çalışmasının dış görünümü Şekil 1.10 'da görüldüğü gibidir.



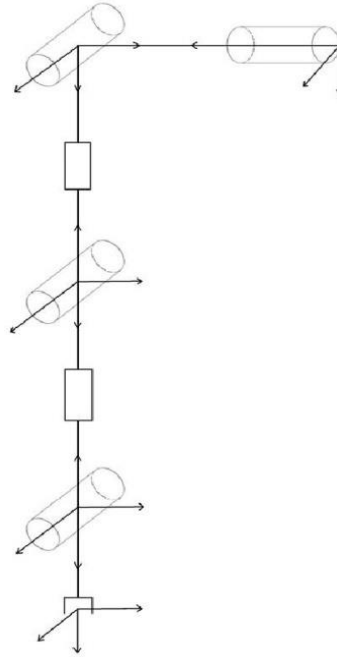
**Şekil 1.9** : MOTOMAN çift kol robotu.



**Şekil 1.10** : FRIDA çift kol robotu.

## 2. ROBOT KOLU TASARIMI

İnsansı robot kolu tasarımındaki en önemli özellik robotun kendi ağırlığı ile beraber ne kadar fazla ağırlığı kaldırabilmesi ve bu işlemi maksimum ne kadar ivmede gerçekleştirebildiğidir. Bu sebeple, motorların uygulayabilecekleri torklar ile kol ağırlığı orantısının (tork / kg) maksimum yapmak, tasarımdaki en önemli hedeftir. 6 serbestlik derecesi olması istenen robot kolunun, motor dizilim şekli de hareket kabiliyeti ve insansı olması konusunda büyük önem taşımaktadır. Motorların dizilim şeması Şekil 2.1 'deki gibidir. Robot tasarımında bir başka düşünülmesi gereken kısım ise dişli mekanizmasıdır. Kullanılacak dişli mekanizmasının hafif olması, boşluksuz olması ve yüksek çevrim oranı sağlayabilmesi gerekmektedir. Özellikle boşluklu çalışan bir tip dişli mekanizmasının robot tasarımında kullanılması düşünülemez. Bu sebeple, Harmonic Drive<sup>®</sup>, yüzey hacmine ve ağırlığına göre çok fazla çevrim oranını boşluksuz olarak sağlayabildiği için robot kolu tasarımında kullanılmıştır.



Şekil 2.1 : 6 serbestlik dereceli robot kolu motor dizilimi.

## 2.1 Ön Tasarım

6 serbestlik dereceli robot kolu tasarımı, ilk olarak İlkay Meşeli tarafından yapılmıştır. İlk tasarımda sürücüsünün daha basit, kontrolünün daha kolay olması nedeniyle RE serisi fırçalı motorlar tercih edilmiştir. Ancak, motorların uzun olması nedeniyle EC serisi motorların kullanılmasına karar verilmiştir. İlkay Meşeli tarafından yapılan tasarım Şekil 2.2 'de görülmektedir.



**Şekil 2.2 :** İlkay Meşeli tarafından yapılan ön tasarım.

Ön tasarımda kullanılan motor ve Harmonic Drive® dişli mekanizmalarının listesi ve özellikleri Çizelge 2.1 'de gösterilmiştir.

Çizelge 2.1' deki listenin dışında ön tasarımda ATI Mini 45 F/T (Kuvvet ve Tork) algılayıcısı son ekleme mevcuttur. Bu algılayıcı ise 0.0917 kg ağırlığında ve maksimum 20 Nm tork ölçümü yapabilmektedir. Robot kolunun toplam ağırlığı yaklaşık 6,5 kg dır.

**Çizelge 2.1 : Ön tasarım motor ve dişli mekanizmalarının özellikleri.**

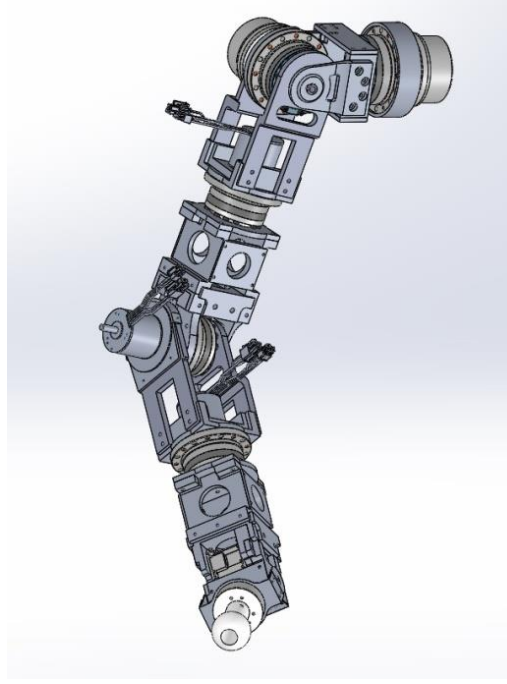
Eklem	Motor	Harmonic Drive®	Motor Maksimum Sürekli Torku (mNm)	Harmonic Drive® Maksimum Giriş Torku (mNm)	Motor Ağırlığı (kg)	Harmonic Drive® Ağırlığı (kg)
1	411678-EC60	CSG-17-2UH	264	583 (1:120)	0,47	0,68
2	411678-EC60	CSG-17-2UH	264	583(1:120)	0,47	0,68
3	251301-EC45	CSG-14-2UH	81,4	360(1:100)	0,11	0,52
4	251301-EC45	CSG-14-2UH	81,4	360(1:100)	0,11	0,52
5	251301-EC45	CSG-14-2UH	81,4	360(1:100)	0,11	0,52
6	Dynamixel MT106	--	8,2	--	--	--

## 2.2 Tasarım 1

Ön tasarımın 3. ve 5. eklemlerinde kullanılan EC45 tipi motorların çıkışlarına herhangi bir mutlak kodlayıcı monte edilememektedir. Bu tip flat motorların arkasında herhangi bir mil de mevcut olmadığı için bu motorların arka kısımlarına da bir artımsal kodlayıcı bağlanamamaktadır. Ayrıca EC45 motorların kendi üzerinde hazırda olan bir artımsal kodlayıcı da mevcut değildir. Bu sebeple, tasarım 1 'deki ilk amaç, kodlayıcı bağlanabilecek aynı zamanda EC45 ile aynı torku üretebilen bir başka motoru mevcut ön tasarımdaki EC 45 motorları ile değiştirmek olmuştur. Bu doğrultuda, 70 Watt gücünde ECi40 tipi motor seçilmiş ve tasarım bu doğrultuda değiştirilmiştir. Tasarım 1'in katı model şekli Şekil 2.3 'de görünmektedir.

Tasarım 1 'de kullanılan motor ve Harmonic Drive® listesi Çizelge 2.2 'deki gibidir.

Ön tasarımda kullanılan ATI mini 45 kuvvet / tork algılayıcısı Tasarım 1'de de kullanılmıştır. Robot kolunun toplam ağırlığı ortalama 6,8 kg olmuştur.



Şekil 2.3 : Tasarım 1 katı model görünümü.

Çizelge 2.2: Tasarım 1 motor ve dişli mekanizmalarının özellikleri.<sup>[11][12]</sup>

Eklem	Motor	Harmonic Drive®	Motor Maksimum Sürekli Torku (mNm)	Harmonic Drive® Maksimum Giriş Torku (mNm)	Motor Ağırlığı (kg)	Harmonic Drive® Ağırlığı (kg)
1	411678-EC60	CSG-17-2UH	264	583(1:120)	0,47	0,68
2	411678-EC60	CSG-17-2UH	264	583(1:120)	0,47	0,68
3	449470-ECi40	CSG-14-2UH(1:100)	82,9	360(1:100)	0,24	0,52
4	449470-ECi40	CSG-14-2UH	82,9	360(1:100)	0,24	0,52
5	449470-ECi40	CSG-14-2UH	82,9	360(1:100)	0,24	0,52
6	Dynamixel MT106	--	8,2	--	--	--

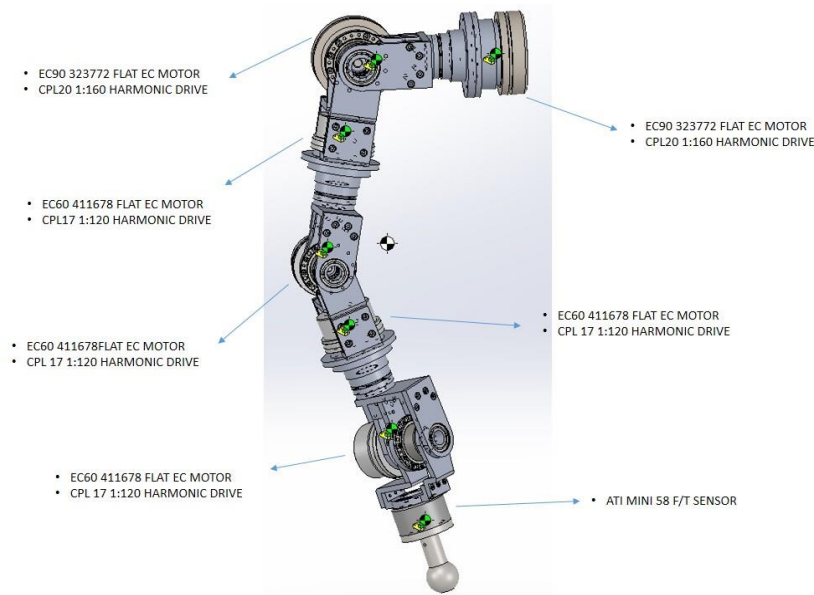
### 2.3 Tasarım 2

Harmonik Drive® üretici firmasının, dayanabildiği maksimum tork miktarına göre çok daha hafif olan yeni tip CPL 2A tipi dişli mekanizmalarının önceki tasarımlarda var olan dişli mekanizmaların yerine kullanılmasına karar verilmiştir. Bununla beraber, kolun toplam ağırlığının azalması sebebiyle daha yüksek tork üretebilen motorlar seçilmiş ve kolun uygulayabileceği tork miktarının artırılması hedeflenmiştir. Özellikle dynamixel yerine tork değeri daha fazla olan bir motor kullanılarak aslında robot kolunun önemli bir kısmını oluşturan bilek bölümünün daha yüksek torklar üretmesi amaçlanmıştır. Gerçekten de iş yapmak için tasarlanan robotların bilek kısımları oldukça güçlü tasarlanmaktadır. Böylece, robot kolunun toplam ağırlığı çok az artmış fakat bununla beraber robot eklemlerinin uygulayabilecekleri tork miktarları çok fazla arttırılmıştır. Bu da, ağırlığına göre çok daha kuvvetli bir robot tasarlanmasını sağlamıştır. Ayrıca, ECi40 gibi arkasında mili olan uzun ve kaba bir motor yerine kendi üzerinde artımsal kodlayıcı bulunan flat tip motorlar seçilmiş böylece tasarımın görselliği iyileştirilmiştir. Tasarım1'de EC60 motorların yerine EC90 tipi daha yüksek tork üreten motorlar, ECi40 ve dynamixel yerine de EC60 tipi motorlar kullanılmıştır. EC90 ve EC60 motorlarının üzerinde kendi kodlayıcıları olması harici bir kodlayıcı takma ihtiyacını da ortadan kaldırmıştır. Artık robot kolunun daha yüksek torklara çıkabilmesi sebebiyle önceki tasarımlarda kullanılan ATI mini 45 tip kuvvet / tork algılayıcısı yerine daha yüksek torklara ve kuvvetlere dayanan ATI mini 58 tip kuvvet / tork algılayıcısı seçilmiştir. ATI mini 45 en fazla 20 Nm torku ölçebilirken, ATI mini58 120 Nm tork değerine kadar ölçüm yapabilmektedir. Şekil 2.4, Şekil 2.5 ve Şekil 2.6, robot kolunun nihai tasarımını olan tasarım 2'yi göstermektedir.

Ayrıca tasarım 2'de birinci, üçüncü ve beşinci eklemlerdeki burkma hareketini yapan 3 motorun çıkışlarına ikişer adet rulman konulmuş ve bir önceki eklem tarafından rulmanlar sabitlenmiştir. Böylece, Harmonic Drive® dişli mekanizmalarının doğrudan aksenal yükü taşıması engellenmiş ve Harmonic Drive® bir yapı elemanı olmaktan çıkmış sadece devir düşürme ve tork arttırma işlemini yapan daha verimli bir dişli mekanizması haline gelmiştir.

Şekil 2.5 'de görüldüğü gibi, tasarımın toplam uzunluğunda çok fazla bir değişiklik söz konusu değildir. Toplam boyu 605.31 mm ve bileğe kadar olan toplam uzunluk

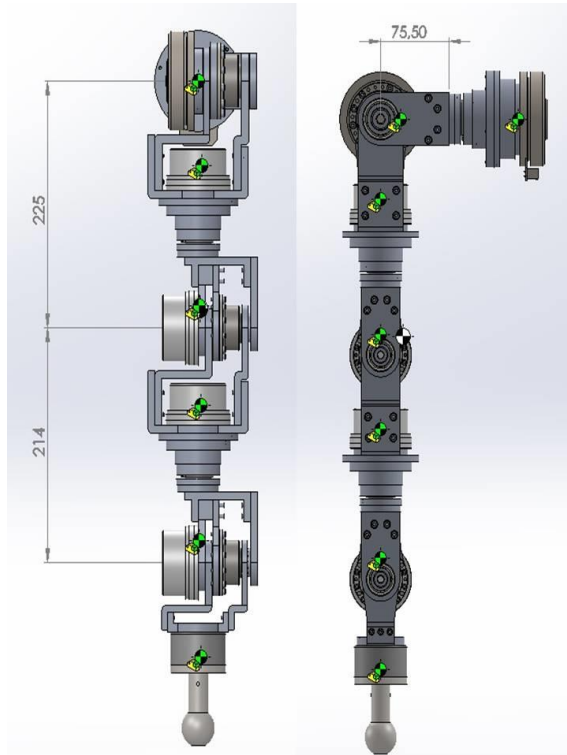
439 mm olmuştur. Bu ölçüler de insansı olması açısından uygun olan ölçüler aralığındadır. Ayrıca Şekil 2.5'de görüldüğü gibi, eklemlerin ağırlık merkezleri hep aynı eksende kesiştirilmiş ve merkezde toplanması sağlanmıştır, bu sayede motorların uygulaması gereken tork miktarlarının düşmesi hedeflenmiştir. Tasarımda çelik olarak 12 adet rulman ve son eklemdaki topuz kısmı vardır. Topuz ise sadece ağırlık yapsın ve belirli bir yük kaldırıldığını varsaymak için çelik malzemeden seçilmiştir. Bunun dışındaki diğer bütün ara bağlantı elemanları ve civatalar dahil alüminyumdur. Tasarım 395,47 gram olan topuz ile birlikte toplam 7620,54 gramdır. Tasarımda kullanılan motor ve Harmonic Drive® listesi ve özellikleri Çizelge 2.4' de görünmektedir. Çizelge 2.4 'de görüldüğü gibi, motorların uygulayabileceği maksimum tork değerleri Harmonic Drive® dişli mekanizmalarının dayanabileceği değerlerden daha düşüktür. Dişli mekanizmalarının ağırlıkları ise önceki tip dişli mekanizmalarına kıyasla çok düşmüş ve yeni motorlar da eski tasarımdakilere nispeten çok daha yüksek tork değerlerini görebilmektedir. Özellikle bilek kısmı bu yeni tasarımda çok daha kuvvetli bir hal almıştır. Çizelge 2.3, nihai tasarımda kullanılan bütün ürünlerin listesini sunmaktadır. Çizelge 2.5, robot kolunda kullanılan eklemlerin ağırlıklarını vermektedir. Robot kolu 7 alt montaj olarak elde edilmiştir ve hepsinin ağırlık merkezi yerleri Şekil 2.5'de görüldüğü gibidir. Seçilen motorların katalogları EK A 'da, Harmonic Drive® dişli mekanizmalarının katalogu ise EK B 'de verilmiştir.



**Şekil 2.4 :** Tasarım 2 görünüm, motor, Harmonic Drive®, algılayıcı dizilimi.



Şekil 2.5 : Tasarım 2 perspektif görünümüleri.



Şekil 2.6 : Tasarım 2 ön ve sağ görünüm.

**Çizelge 2.3:** Tasarım 2 kullanılan malzeme listesi.<sup>[11][12]</sup>

Malzeme	Özelliği	Adet	Ağırlığı (kg)
EC90 323772 Motor	90 Watt, 24 V, Max. tork: 444 mNm	2	0,60
EC60 411678 Motor	100 Watt, 24 V, Max. Tork : 264 mNm	4	0,47
EC90 encoder mile453233	1024 Pulse/N Artımsal	2	0,01
EC60 encoder mile 421986	1024 Pulse/N Artımsal	4	0,01
Harmonic Drive® CPL-20A	1:160 çevrim oranı Max çıkış tork :92 Nm Max giriş tork :0,575 Nm	2	0,14
Harmonic Drive® CPL-17A	1:120 çevrim oranı Max çıkış tork : 54 Nm Max. giriş tork: 0,450 Nm	4	0,1
ATI mini58 Kuvvet/Tork Algılayıcısı	Max. eksenel kuvvet : 6800 N Max. kesme kuvveti : 2800 N Max. Tork: 120 Nm	1	0,345

Kodlayıcı olarak bir turda 1024 adıma sahip olan bir algılayıcı seçilmesinin sebebi motorların yüksek devirde dönmesidir. Daha fazla adıma sahip bir kodlayıcı seçilse sürücü açısından hız problemi teşkil edebilirdi. Bu koşullar altında kolun son konumunda oluşabilecek en fazla hata miktarı 0.017 mm 'dir.

**Çizelge 2.4 :** Tasarım 2 motor ve dişli mekanizmalarının özellikleri.<sup>[11][12]</sup>

Ekleme	Motor	Harmonic Drive®	Motor Maksimum Sürekli Torku (mNm)	Harmonic Drive® Maksimum Giriş Torku (mNm)	Motor Ağırlığı (kg)	Harmonic Drive® Ağırlığı (kg)
1	323772-EC90	CPL-20A-(1:160)	444	575(1:160)	0,6	0,14
2	323772-EC90	CPL-20A-(1:160)	444	575(1:160)	0,6	0,14
3	411678-EC60	CPL-17A(1:120)	264	450(1:120)	0,47	0,1
4	411678-EC60	CPL-17A(1:120)	264	450(1:120)	0,47	0,1
5	411678-EC60	CPL-17A(1:120)	264	450(1:120)	0,47	0,1
6	411678-EC60	CPL-17A(1:120)	264	450(1:120)	0,47	0,1

**Çizelge 2.5 :** Robot kolu ekleme ağırlıkları.

Ekleme	Ağırlık (kg)
1	1,08
2	1,33
3	1,23
4	0,99
5	1,02
6	0,99
7	0,97
Toplam robot kolu ağırlığı	7,62



### 3. İLERİ KİNEMATİK VE TERS KİNEMATİK ÇÖZÜMÜ

Robot tasarımı yapıldıktan sonra çözülmesi gereken ilk problem, eklemlerin açılmal hareketi sonucu oluşan son noktanın konum ve yönelimini belirlemektir. Bu problemin çözülebilmesi için ise robot sisteminin ileri kinematiğı çözümlenmesi gerekmektedir. Fakat asıl çözülmesi gereken ise istenilen konum ve yönelimi sağlamak için eklemlerin ne kadar açılmal hareket yapmasını belirlemektir ki bu da ters kinematik çözümlü ile elde edilir. Eklemlerin birbirleriyle olan ilişkisini ifade eden geçiş matrisleri aracılığı ile elde edilen ileri kinematik çözümlü hem ters kinematik çözümlü bulunurken kullanılır hem de ters kinematik çözümlü bir sağlamasıdır.

Bu doğrultuda robot kolu tasarımıımızın ileri kinematik ve ters kinematik çözümlü tez çalışmasının devam eden kısmında sırasıyla anlatılacaktır.

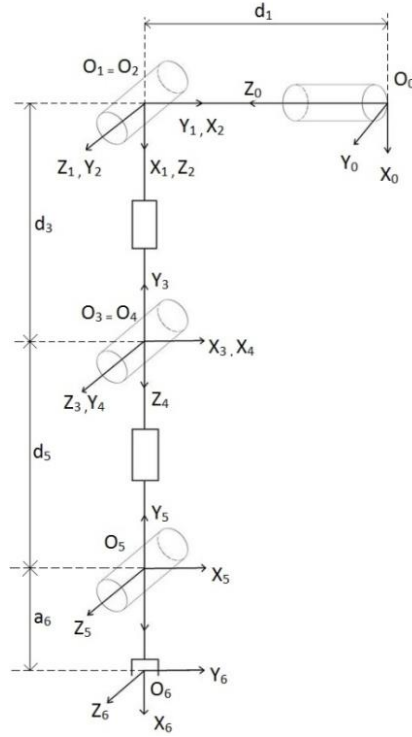
#### 3.1 İleri Kinematik Çözümlü

Robot kolunun ileri kinematik çözümlünü bulmak için Denavit-Hartenberg (DH) parametreleri kullanılmıştır ve tüm çözümlü DH kıstaslarına göre ayarlanmıştır. Şekil 3.1, DH parametrelerine göre bütün motor merkezlerini ve koordinat düzlemlerini ifade etmektedir.

Motorların merkezleri ve koordinat düzlemleri belirlendikten sonra, DH parametreleri belirlenmiştir. DH parametrelerini belirlemek için aşağıda bulunan listedeki tanımlamalar kullanılmıştır ve Çizelge 3.1, nihai tasarımın DH parametrelerini göstermektedir.

DH parametreleri belirlendikten sonra denklem 1'de formülasyonu verilen ardışık eklemler arası homojen geçiş matrisleri ( A matrisleri ) yazılmıştır.<sup>[13]</sup> Ardışık eklemler arası geçiş matrisleri belirlendikten sonra, artık robot kolunun sabit olan

birinci eklem ile son uç noktamız arasındaki toplam homojen geçiş matrisi (T matrisi) denklem 'deki gibi belirlenir.



Şekil 3.1 : DH kıstasına göre motor merkez ve koordinatları.

- $\theta_i(^{\circ})$  :  $z_{i-1}$  yönünde  $x_{i-1}$  ile  $x_i$  arasındaki açı.<sup>[13]</sup>
- $d_i(\text{mm})$  :  $z_{i-1}$  yönünde  $z_{i-1}$  ile  $x_i$  kesişiminin  $O'_{i-1}$  e olan uzaklığı.<sup>[13]</sup>
- $\alpha_i(^{\circ}) = x_i$  yönünde  $z_{i-1}$  ile  $z_i$  arasındaki açı.<sup>[13]</sup>
- $a_i(\text{mm}) = x_i$  yönünde  $z_{i-1}$  ile  $x_i$  kesişiminin  $O'_i$  ye olan uzaklığı.<sup>[13]</sup>

Çizelge 3.1: Robot kolu DH parametreleri.

Eklem(i)	$a_i(\text{mm})$	$\alpha_i(^{\circ})$	$d_i(\text{mm})$	$\theta_i(^{\circ})$	Aralık ( $^{\circ}$ )
1	0	-90	75,5	$\theta_1^*$	-180...+180
2	0	90	0	$\theta_2^* + 90$	-15...+195
3	0	-90	225	$\theta_3^*$	-180...+180
4	0	90	0	$\theta_4^*$	-115...+115
5	0	-90	214	$\theta_5^*$	-180...+180
6	166,31	0	0	$\theta_6^* - 90$	-115...+115

$$A_i = \begin{vmatrix} c\theta_i & -s\theta_i * c\alpha_i & s\theta_i * s\alpha_i & a_i * c\theta_i \\ s\theta_i & c\theta_i * c\alpha_i & -c\theta_i * s\alpha_i & a_i * s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (1)$$

$$T = \prod_{i=1}^6 A_i \quad (2)$$

İleri kinematik çözümünün elde edildiği MATLAB® kodu Ek C kısmında mevcuttur.

### 3.2 Ters Kinematik Çözümü

Yörünge planlaması kısmına geçmeden önce çözümlenmesi gereken ters kinematik problemi bize, istenilen uç nokta konum ve yönelimine karşılık gelen eklem açılal dönme miktarlarını bulmamızı sağlamaktadır. Robotlar ters kinematik problemi açısından genel olarak 3 farklı formda karşımıza çıkarlar. Bunlardan ilki daima tekil çözüm veren robotlardır. İkincisi ise, istenilen konum ve yönelime maksimum belirli sayıda çözüm verebilen robotlardır. Bu robotlarda mutlaka tekil çözüm verdikleri konum ve yönelimler mevcuttur. Örnek vermek gerekirse, her robotun çalışma alanının sınırları daima tekil sonuç vermek zorundadır. Üçüncü ve sonuncu tip ise, belirli bir aralık içerisinde sınırlı sayıda çözüm sunmayan robotlardır. Tasarımı yapılan insansı robot kolu ise üçüncü tip robot olarak kabul görebilir. Robot eklemlerinin çalışma aralığı da oluşabilecek çözüm sayısını azaltan bir etken olarak karşımıza çıkmaktadır. Üçüncü tip olarak tanımladığımız robotları ikinci tip robotlardan ayıran en önemli özellik, son 3 eklem koordinat düzlemlerindeki eksen yönlerinin çakışmıyor olması diyebiliriz. Bu eksenlerin çakışmıyor olması kinematik ayırım yöntemi uygulanamamasına sebep olmaktadır. Bir başka deyişle, bilek hareketi son eklem olduğu için kinematik ayırım ile çözüm bulunamamakta, bu ise tasarımı yapılan robotu üçüncü tip robot sınıfına sokmaktadır.

Bu sebeple ters kinematik çözümünü bulurken tamamı ile geometrik bir yaklaşım ile çözüm yoluna gidilmeye çalışılmıştır. İlk olarak, robot uç noktasının istenilen konumuna bakılarak çözüme başlanmış ve istenilen son konuma gelebilmek için

dirsek kısmının yani dördüncü merkezimizin  $x$  ekseninde hangi aralıkta olması gerektiği saptanmıştır. Bir başka deyişle, son noktaya varmak için dirseğimizin hangi  $x$  değerlerinde olması gerektiği bulunmuştur.

Şekil 3.2 incelendiği zaman,  $(x_e, y_e, z_e')$  noktası olarak adlandırdığımız dirseğin bulunduğu konumun veya  $O_4$  merkezinin,  $O_2$  merkezi etrafında  $d_3$  mesafesi kadar yarıçaplı bir kürenin yüzeyinde kalmak zorunda olduğu görülmektedir. Şekil 3.3 ise  $O_4$  merkezinin aynı zamanda konumunu  $(x_w, y_w, z_w')$  olarak adlandırdığımız  $O_5$  merkezi etrafında  $d_5$  mesafesi kadar yarıçaplı bir kürenin yüzeyinde kalmak zorunda olduğunu göstermektedir. Uç konumunun istenilen yönelimi ve konumu  $(x_6, y_6, z_6')$  bilindiği takdirde ise  $O_5$  konumunu ifade eden  $(x_w, y_w, z_w')$  değerleri belirlenebilmektedir. Bunun için ilk önce son konum yönelimi euler açısı formasyonunda ZYZ dönüşümü kullanılarak bulunur. Euler ZYZ dönüşümü aşağıdaki gibidir.<sup>[13]</sup>

$\mu$  : Z yönünde ilk dönüş.

$\beta$  : Y yönünde dönüş.

$\gamma$  : Z yönünde ikinci dönüş

$$R = \begin{matrix} c\mu * c\beta * c\gamma - s\mu * s\gamma & -c\beta * s\gamma - s\mu * c\gamma & c\mu * s\beta \\ s\mu * c\beta * c\gamma + c\mu * s\gamma & -s\mu * c\beta * s\gamma + c\mu * c\gamma & s\mu * s\beta \\ -s\beta * c\gamma & s\beta * s\gamma & c\beta \end{matrix} \quad (3)$$

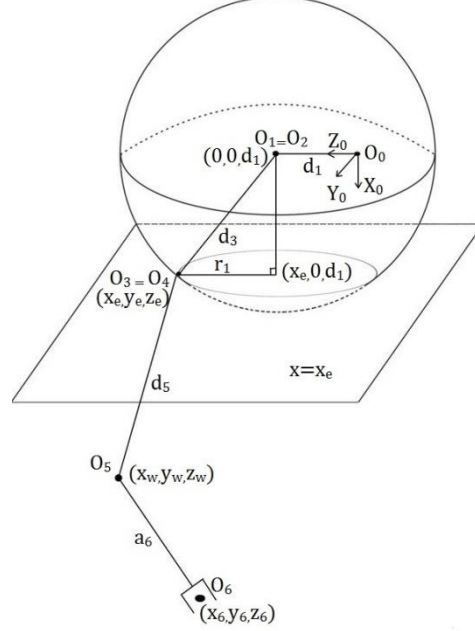
Artık istenilen uç konumu yönelimi ve konumuna karşılık gelen  $(x_w, y_w, z_w')$  noktası aşağıdaki gibi bulunabilir.<sup>[13]</sup>

$$x_w = x_6 - a_6 * R(1,1) \quad (4)$$

$$y_w = y_6 - a_6 * R(2,1) \quad (5)$$

$$z_w = z_6 - a_6 * R(3,1) \quad (6)$$

Şekil 3.2 aynı zamanda,  $d_3$  yarıçaplı küreyi herhangi bir  $x = x_e$  mesafesinde kestiğimiz ve  $O_4$  merkezini aynı  $x$  değerinde tutmak istediğimiz zaman, dirsek konumunun o  $x$  değerini korumak için  $r_1$  yarıçaplı bir çember üzerinde kalmak zorunda olduğunu göstermektedir.



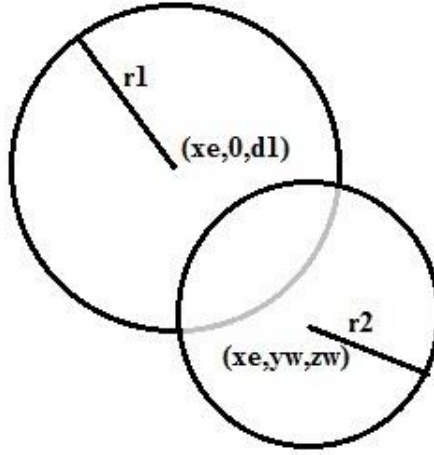
Şekil 3.2 :  $O_4$  noktasının  $O_2$  noktasına göre hareket kabiliyeti.

Şekil 3.3 ise aynı  $x = x_e$  mesafesinde alt kısımda oluşan küreyi kestiğimiz ve  $O_4$  merkezini aynı  $x$  değerinde tutmak istediğimiz zaman, dirsek konumunun o  $x$  değerini korumak için  $r_2$  yarıçaplı bir diğer çember üzerinde kalmak zorunda olduğunu göstermektedir. 7. ve 8. denklemler  $r_1$  ve  $r_2$  yarıçaplarının herhangi bir  $x = x_e$  değerindeki değerlerini bulmamızı sağlamaktadır.<sup>[14]</sup>

Artık elimizde yarıçaplarını herhangi bir  $x = x_e$  noktası için yarıçaplarını bildiğimiz 2 tane çember bulunmaktadır. Şekil 3.4 'de görüldüğü gibi bu 2 çemberin kesişim noktaları ise dirsek bölgemizin  $y$  ve  $z$  eksenlerinin koordinatlarını belirlememize yarayacaktır. Bunun için ise denklem 9 ve 10 'daki  $r_1$  ve  $r_2$  yarıçaplı iki çemberin denklemleri birbirine eşitlenmiştir.<sup>[14]</sup>

$$r_1 = \sqrt{d_3^2 - x_e^2} \quad (7)$$





Şekil 3.4 :  $r_1$  ve  $r_2$  yarıçaplı 2 çemberin kesişimi.

$$A = \frac{(y_w^e + r_1^2 + z_w^2 - r_2^2 - d_1^2)}{2 * y_w} \quad (13)$$

$$B = \frac{(z_w - d_1)}{y_w} \quad (14)$$

Denklem 11 ifadesini tanımsız yapan durumlardan biri ( $z_w - d_1 = y_w$ ) , diğeri ( $y_w = 0$ ) durumlardır. Bu durumlar altındaki çözümlerde aşağıda sıralanmıştır.

Eğer ( $z_w - d_1 = y_w$ ) ;

$$z_{e 1,2} = \frac{2*(A+d_1) \mp \sqrt{4*A^2 + 8*A*d_1 + 4*d_1^2 - 8*(A^2 + d_1^2 - r_1^2)}}{4} \quad (15)$$

$$y_{e 1,2} = A - z_{e 1,2} \quad (16)$$

Eğer ( $y_w = 0$ );

$$z_{e 1,2} = \frac{(r_2^2 - r_1^2 - z_w^2 + d_1^2)}{2 * (d_1 - z_w)} \quad (17)$$

$$y_{e\ 1,2} = \mp \sqrt{r_2^2 - z_{e\ 1,2}^2 - z_w^2 + 2 * z_w * z_{e\ 1,2}} \quad (18)$$

Şekil 3.2 'de görüldüğü gibi,  $x_e$  değeri  $-d_3$  ile  $d_3$  aralığında bir değer alabilmektedir. Bir for döngüsü içerisinde son konumu sağlayabilen bütün  $x_e$  değerleri ve buna bağlı olarak  $y_e$  ve  $z_e$  değerleri bulunmuş olur. Fakat son konumu sağlayabilen bütün olası dirsek koordinatları istenilen son konum yönelimini sağlayamamaktadır. Bu sebeple, bunların arasından istenilen yönelimi sağlayan ( $'x_e, y_e, z_e'$ ) koordinatları belirlenir. Bunun bulunması için ise 4. merkezin rotasyon matrisinin belirlenmesi gerekir. Aşağıdaki denklemler ise 4. merkeze kadar olan geçiş matrisinin bulunuşunu anlatmaktadır.

$$T_4 = \prod_{i=1}^4 A_i \quad (19)$$

$$T_4 = \begin{bmatrix} \vec{X}_4 & \vec{Y}_4 & \vec{Z}_4 \vec{E} \\ 0 & 0 & 0 \ 1 \end{bmatrix} \quad (20)$$

$$\vec{E} = [x_e; y_e; z_e] \quad (21)$$

$\vec{Z}_4$  vektörü  $O_4$  merkezinden  $O_5$  merkezine doğru uzanan bir vektör olduğuna göre, aşağıdaki formda yazılabilir.

$$\vec{Z}_4 = \frac{(\vec{W} - \vec{E})}{\|\vec{W} - \vec{E}\|} \quad (22)$$

$$\vec{W} = [x_w; y_w; z_w] \quad (23)$$

Eğer  $O_2$  konumunu ifade eden  $\vec{R}_5$  vektörünü ve  $O_4$  merkezinden  $O_2$  merkezine doğru olan  $\vec{R}_0$  adını verdiğimiz vektörü yazmamız gerekirse aşağıdaki gibi olacaktır.

$$\vec{R}_5 = [0; 0; d_1] \quad (24)$$

$$\vec{R}_0 = \frac{(\vec{R}_s - \vec{E})}{\|\vec{R}_s - \vec{E}\|} \quad (22)$$

Son olarak  $\vec{Y}_4$  ve  $\vec{X}_4$  vektörleri denklem 23 ve 24 'deki ifadeler ile bulunabilir.

$$\vec{Y}_4 = \vec{R}_0 \times \vec{Z}_4 \quad (23)$$

$$\vec{X}_4 = \vec{Y}_4 \times \vec{Z}_4 \quad (24)$$

İstenilen uç konum konumunu sağlayan ( $x_e, y_e, z_e$ ) konumuna göre 4. ekleme kadar olan geçiş matrisi belirlendikten sonra 3. ve 5. ekleme kadar olan geçiş matrisleri artık belirlenebilir. Bu geçiş matrislerini bulmamızdaki amaç hem ters kinematik işlemi daha kolay yapabilmek hem de istenilen son konum yönelimini sağlayabilmektir. Aynı yöntem ile aşağıdaki ifadeler aracılığı ile 3. ve 5. ekleme kadar olan geçiş matrisleri bulunmuştur.

$$T_3 = \prod_{i=1}^3 A_i \quad (25)$$

$$T_3 = \begin{matrix} \vec{X}_3 & \vec{Y}_3 & \vec{Z}_3 \vec{E} \\ 0 & 0 & 0 \ 1 \end{matrix} \quad (26)$$

İlk olarak  $\vec{Y}_3$  vektörü, motor merkez ve koordinatlarını gösteren Şekil 3.1 'de görüldüğü gibi  $\vec{Z}_4$  vektörünün tersi yönündedir ve aşağıdaki ifade aracılığı ile bulunur.

$$\vec{Y}_3 = \frac{(\vec{E} - \vec{W})}{\|\vec{E} - \vec{W}\|} \quad (27)$$

$\vec{Z}_3$  vektörü ise  $O_4$  merkezinden  $O_2$  merkezine doğru olan  $\vec{R}_0$  vektörü ile  $\vec{Y}_3$  vektörünün çarpımı ile denklem 23'de görüldüğü gibi bulunur.

$$\vec{Z}_3 = \vec{R}_0 \times \vec{Y}_3 \quad (28)$$

Son olarak  $\vec{Z}_3$  ve  $\vec{Y}_3$  belirlendiğine göre  $\vec{X}_3$  vektörü denklem 29'da olduğu gibi tespit edilir.

$$\vec{X}_3 = \vec{Y}_3 \times \vec{Z}_3 \quad (29)$$

Aynı yöntem 5. ekleme kadar olan geçiş matrisinin bulunmasında da kullanılmıştır. 5. ekleme kadar olan geçiş matrisinin bulunuşu aşağıdaki gibidir.

$$T_5 = \prod_{i=1}^5 A_i \quad (30)$$

$$T_5 = \begin{pmatrix} \vec{X}_5 & \vec{Y}_5 & \vec{Z}_5 \vec{W} \\ 0 & 0 & 0 \ 1 \end{pmatrix} \quad (31)$$

$\vec{Y}_5$  vektörü, Şekil 3.1 'de görüldüğü gibi,  $O_5$  konumundan  $O_6$  konumuna olan vektörün ters işaretlisi olacaktır. Son konum konumu olan  $O_6$  konumu yani  $(x, y, z)$  konumunu ifade eden vektöre  $W_s$  ismini verirse, denklem 32' deki ifade  $\vec{Y}_5$  vektörünün bulunuşunu göstermektedir.

$$\vec{Y}_5 = \frac{(\vec{W} - \vec{W}_s)}{\|\vec{W} - \vec{W}_s\|} \quad (32)$$

$O_5$  konumundan  $O_4$  konumuna olan vektöre  $\vec{R}_w$  ismini verilmiştir ve denklem 33 'de olan ifade ile bulunmaktadır. Dolayısıyla, denklem 34 'de görüldüğü gibi  $\vec{Z}_5$  vektörü  $\vec{Y}_5$  vektörü ile  $\vec{R}_w$  vektörünün çarpımının sonucudur.

$$\vec{R}_w = \frac{(\vec{E} - \vec{W})}{\|\vec{E} - \vec{W}\|} \quad (34)$$

$$\vec{Z}_5 = \vec{Y}_5 \times \vec{R}_w \quad (35)$$

Sonuç olarak  $\vec{X}_5$  vektörü ise denklem 36' da görüldüğü gibi,  $\vec{Y}_5$  vektörü ile  $\vec{Z}_5$  vektörünün çarpımıdır.

$$\vec{X}_5 = \vec{Y}_5 x \vec{Z}_5 \quad (36)$$

Yukarda bulunan sonuçlar istenilen son uç konumunu yani  $(x, y, z)$  konumunu sağlamamız için olabilecek dirsek konumuna yani  $(x_e, y_e, z_e)$  konumuna göre belirlenmiştir. Fakat daha önce de bahsedildiği gibi, bütün olası dirsek konumları istenilen son konum yönelimini sağlayamaz. Bu sebeple, son konum yönelimini doğru veren dirsek konumlarını ve buna bağlı oluşan 3. , 4. ve 5. eklem geçiş matrislerinin belirlenmesi gerekir. İlk olarak 6. ekleme yani son ekleme kadar olan toplam geçiş matrisi, denklem 37' deki gibi tanımlanır.

$$T = \begin{bmatrix} \vec{X}_6 & \vec{Y}_6 & \vec{Z}_6 \vec{W}_s \\ 0 & 0 & 0 \quad 1 \end{bmatrix} \quad (37)$$

Şekil 3.1'de görüldüğü gibi,  $\vec{Z}_6$  vektörü hangi doğrultuda olursa olsun  $\vec{Z}_5$  ve  $\vec{Y}_4$  vektörleri ile aynı doğrultuda olmak zorundadır.  $\vec{Z}_6$  vektörü aynı zamanda euler formasyonu tarafından bulunmuş son eklem rotasyon matrisinin 3. sütununun ilk üç değeridir ve aşağıdaki denklem 38,  $\vec{Z}_6$  vektörünü ifade etmektedir.

$$\vec{Z}_6 = [R(1,3); R(2,3); R(3,3)] \quad (38)$$

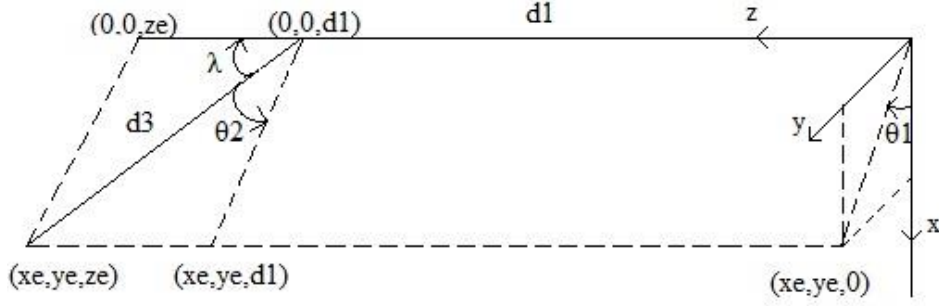
Sonuç olarak, olası  $(x_e, y_e, z_e)$  dirsek konumlarına göre bulunan  $\vec{Z}_5$  ve  $\vec{Y}_4$  vektörleri arasından,  $\vec{Z}_6$  vektörüne belirli bir hata miktarından daha küçük bir değerle yaklaşan sonuçlar, bizim ters kinematik çözümünde kullanacağımız 3., 4. ve 5. ekleme kadar olan geçiş matrislerini oluşturacaklardır. Bir başka deyişle,  $(x_e, y_e, z_e)$  dirsek konumlarına göre bulunan  $\vec{Z}_5$  ve  $\vec{Y}_4$  vektörleri arasından 39 ve 40 numaralı denklemdeki koşulları sağlayanlar bizim çözümümüzü oluşturacaklardır.

$$\vec{Y}_4 = \vec{R}_0 x \vec{Z}_4 \cong [R(1,3); R(2,3); R(3,3)] \quad (39)$$

$$\vec{Z}_5 = \vec{Y}_5 x \vec{R}_w \cong [R(1,3); R(2,3); R(3,3)] \quad (40)$$

Artık sayısal değerleri belirlenmiş olan 3., 4. ve 5. ekleme kadar olan geçiş matrislerine  $(T_3, T_4, T_5)$  göre gereken eklem açıları aşağıdaki sırada bulunur. İlk

olarak  $\theta_1$  ve  $\theta_2$  değerlerini geometrik yaklaşım ile Şekil 3.5 'de görüldüğü şekilde bulabiliriz.



Şekil 3.5 :  $\theta_1$  ve  $\theta_2$  açılarının geometrik gösterimi.

$$\theta_1 = \text{atan}\left(\frac{y_e}{x_e}\right) = \text{atan}\left(\frac{T_4(2,4)}{T_4(1,4)}\right) \quad (41)$$

$$\theta_2 = \text{atan}\left(\frac{\sqrt{x_e^2 + y_e^2}}{z_e - d_1}\right) - \frac{\pi}{2} \quad (42)$$

$\theta_1$  ve  $\theta_2$  açı değerleri bulunduğuna göre artık  $\theta_3, \theta_4, \theta_5$  ve  $\theta_6$  değerleri geçiş matrisleri kullanılarak aşağıdaki gibi sırasıyla bulunabilir.

$$L = A_1^{-1} * T_3 = A_2 * A_3 \quad (43)$$

$$\theta_3 = \text{atan}\left(\frac{L(1,3)}{L(3,3)}\right) \quad (44)$$

$$K = A_1^{-1} * T_4 = A_2 * A_3 * A_4 \quad (45)$$

$$\theta_4 = \text{atan}\left(\frac{K(3,3)}{K(1,3)}\right) \quad (46)$$

$$A_5 = \begin{vmatrix} c\theta_5 & 0 & -s\theta_5 & 0 \\ s\theta_5 & 0 & c\theta_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (47)$$

$$M = A_4^{-1} * A_3^{-1} * A_2^{-1} * A_1^{-1} * T_5 = A_5 \quad (48)$$

$$\theta_5 = \text{atan}\left(\frac{M(1,2)}{M(1,1)}\right) \quad (49)$$

$$A_6 = \begin{vmatrix} s\theta_6 & c\theta_6 & 0 & a_6 * s\theta_6 \\ -c\theta_6 & s\theta_6 & 0 & -a_6 * c\theta_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (50)$$

$$S = A_5^{-1} * A_4^{-1} * A_3^{-1} * A_2^{-1} * A_1^{-1} * T = A_6 \quad (51)$$

$$\theta_6 = \text{atan}\left(\frac{S(1,1)}{S(1,2)}\right) \quad (52)$$

Ters kinematik çözümünün elde edildiği MATLAB® kodu Ek D kısmında mevcuttur. .



## 4. YÖRÜNGE PLANLANMASI

İnsansı robot kolunun ters kinematik çözümü yapıldıktan sonra sıradaki işlem, yörünge planlanması adını verdiğimiz ilk konumdan son konuma giderken robot eklemlerinin, açısal hızlarının ve açısal ivmelerinin zamana göre değişiminin belirlenmesidir. Bu işlem sonucu elde ettiğimiz anlık açısal konum, açısal hız ve açısal ivme değerleri benzetim sırasında referans değerler olarak kullanılacaktır.

Yörünge planlaması yapılırken dikkat edilmesi gereken en önemli şeylerden biri bütün eklemlerin son konumlarına doğru giderlerken hareketlerine aynı anda başlayıp aynı anda son vermiş olmalarıdır. Yörünge planlaması sırasında çok çeşit hız profilleri kullanılabilir. Bu çalışmada trapez ve kübik ismini verdiğimiz 2 farklı hız profili kullanılacaktır.

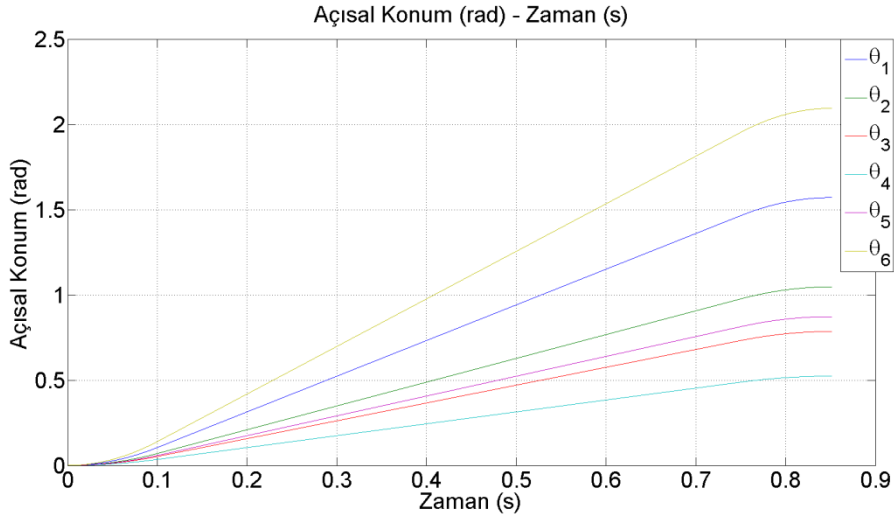
### 4.1 Trapez Hız Profili İle Yörünge Planlanması

Trapez hız profili toplam hareket süresini en düşük tutan hız profilidir. Fakat bu özelliğinin yanında, trapez hız profilini oluşturan ivme grafiğinde süreksiz bölgeler yer almaktadır. Bu da ani hız değişimlerine sebep olmakta ve kontrolü zorlaştırmaktadır. Bu sebeple, bazı çalışmalarda ani ivme değişiminin görüldüğü bu süreksiz ivme bölgelerine daha yumuşak bir geçiş verilmekte, böylece süreksiz ivme profilinden kurtulmaya çalışılmaktadır. Fakat bu çalışmada sadece benzetim yapıldığı için standart trapez hız profili kullanılmıştır.

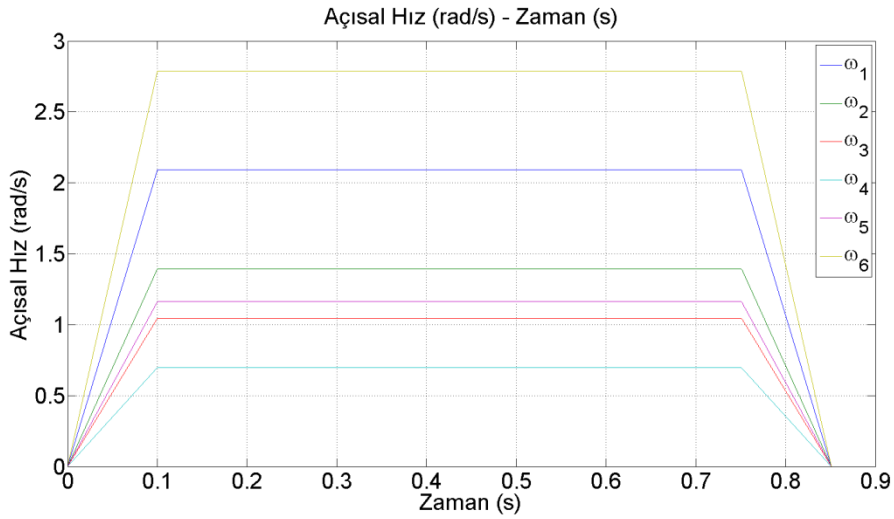
Trapez hız profili, doğrusal bir ivmelenme bölgesi, sabit hız bölgesi ve tekrar doğrusal eksi yönde ivmelenme bölgelerinden ibarettir. Şekil 4.1, Şekil 4.2 ve Şekil 4.3 yapılan hareketin Çizelge 4.1'de tanımlandığı örnek bir trapez hız profili grafiği ve bu profile bağlı olan konum ve ivme grafiklerini göstermektedir. Grafikler motorlara değil doğrudan eklemlere aittir.

**Çizelge4.1:** Örnek trapez hız profili için hareket çizelgesi.

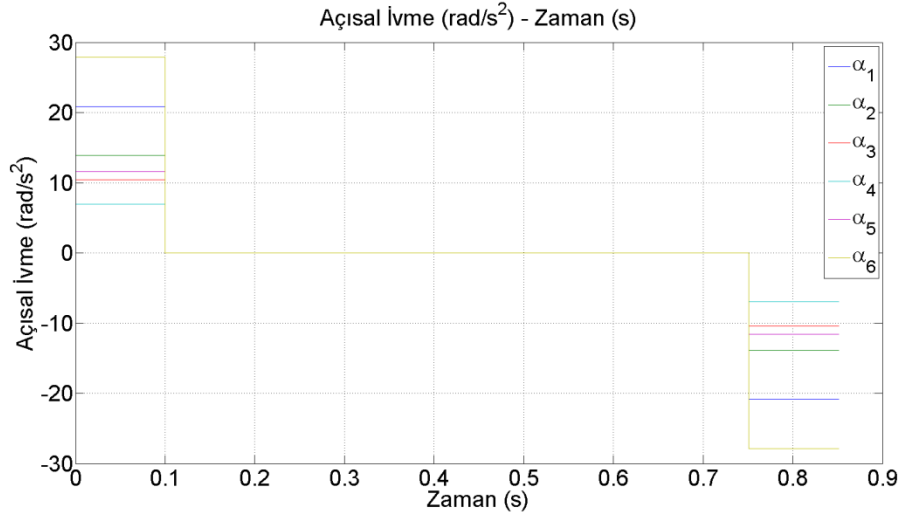
Eklem(i)	İlk Açısal Konum (rad)	Son Açısal Konum (rad)
1	0	1,5708
2	0	1.0472
3	0	0.7854
4	0	0.5236
5	0	0.8727
6	0	2.0944



**Şekil 4.1 :** Trapez hız profilinde eklemlerin konum-zaman grafiği örneği.



**Şekil 4.2 :** Trapez hız profilinde eklemlerin hız-zaman grafiği örneği.



**Şekil 4.3 :** Trapez hız profilinde eklemlerin ivme-zaman grafiği örneği.

Şekil 4.3 'de görüldüğü gibi trapez hız profili süreksiz ivme grafiği sağlamakta bunun sonucunda ani hız değişimleri yaşanmaktadır.

Çizelge 4.2 'de motorların uygulayabilecekleri maksimum hız ve ivmelerin çevrim oranına bölünmesiyle bulunan eklemlerin maksimum hız ve ivmelerini belirtilmektedir. Bu tablodaki değerler doğrultusunda tez çalışmasının devamında trapez hız profilinin nasıl belirlendiği adım adım anlatılmaktadır. Trapez hız profili ile yörünge planlanmasının MATLAB® kodu Ek E kısmında mevcuttur.

**Çizelge 4.2 :** Eklemlerin maksimum hız ve ivme değerleri.

Eklemler(i)	Eklemler 1	Eklemler 2	Eklemler 3	Eklemler 4	Eklemler 5	Eklemler 6
Max. Eklemler İvmesi (rad/s <sup>2</sup> )	20,9	20,9	37,1	37,1	37,1	37,1
Max. Eklemler Hızı (rad/s)	2,09	2,09	3,26	3,26	3,26	3,26
Max. Motor Hızı (RPM)	2590	2590	3740	3740	3740	3740

- Adım 1: Ters geometrik modelden elde edilen ilk ve son açısal konum bilgileri trapez yörünge planlanmasının girdileri olarak ayarlanır.

- Adım 2: En fazla açısal hareket yapan eklem belirlenir.
- Adım 3: Bu değer sistemdeki maksimum açısal hareket olarak atanır ve o ekleme Çizelge 4.2 'de karşılık gelen maksimum açısal hız değeri de sistemdeki maksimum açısal hız değeri olarak atanır.
- Adım 4: Sistemdeki maksimum ivme değeri olarak da  $37,1 \text{ rad/s}^2$  seçilir. Bu değer Çizelge 4.2 'de görüldüğü gibi eklemler arasındaki maksimum ivme değeridir.
- Adım 5: Sonrasında da aşağıdaki ifadeler aracılığı ile ivme süresi ve toplam hareket zamanı bulunur.

$$\text{ivme süresi} = \frac{\text{max açısal hız}}{\text{max açısal ivme}} \quad (53)$$

$$\text{toplam zaman} = \text{ivme süresi} + \frac{|\text{max. açısal hareket}|}{\text{max açısal hız}} \quad (54)$$

- Adım 6: Toplam zaman ve ivme süresi belirlendiğine göre artık tüm eklemlerin maksimum açısal hız ve ivmeleri denklem 55 ve 56 'da olduğu gibi bulunur.

$$\text{maksimum eklem hızı}(i) = \frac{\theta(i)}{\text{toplam zaman} - \text{ivme süresi}} \quad (55)$$

$$\text{Eklem ivmesi}(i) = \frac{\text{maksimum eklem hızı}(i)}{\text{ivme süresi}} \quad (56)$$

- Adım 7: Denklem 55 ve 56'da bulunan maksimum eklem hızları ve eklem ivmelerinin Çizelge 4.2 'deki eklem maksimum hız ve ivmelerinden büyük olup olmadığı kıyaslanır.
- Adım 8: Eğer adım 7 'nin cevabı evet ise, adım 4 'e geri dön ve maksimum ivme değeri olan  $37,1 \text{ rad/s}^2$  değerini belirli bir miktar azalt. Bu işleme denklem 56'da bulunan ivme değerlerinin Çizelge 4.2'deki eklem maksimum ivme değerlerinden daha küçük bulunmasına kadar devam et.

- Adım 9: Eğer hesaplanan ivme değerleri maksimum ivme değerlerinden küçük olduğu halde hesaplanan hız değerlerinden herhangi biri Çizelge 4.2'deki maksimum eklem açısal hızlarından daha büyük ise adım 3'e geri dön. Bu işlemi bütün hesaplanan ivme ve hız değerlerinin Çizelge 4.2'deki maksimum eklem hız ve ivmelerinden küçük bulunmasına kadar devam et.

Yukarıdaki adımlar 2 adet for döngüsü içerisinde MATLAB® programı tarafından yapılmıştır. Artık bütün eklemlerin maksimum açısal hızları ve ivme değerleri tespit edilmiştir. Sıradaki işlem ise elde ettiğimiz değerler doğrultusunda eklemlerin anlık açısal konum, açısal hız ve açısal ivme değerlerini aşağıdaki denklemler ile saptamaktır. Anlık değerler her  $\Delta zaman = 0.04s$  aralığında belirlenmiştir.

- Eğer (zaman  $\leq$  ivme süresi)

$$anlık\ açısal\ ivme(i) = Eklem\ ivmesi(i)(sabit) \quad (57)$$

$$anlık\ açısal\ hız(i) = anlık\ ivme(i) * zaman \quad (58)$$

$$anlık\ açısal\ pozisyon(i) = ilk\ açısal\ konum + \frac{anlık\ açısal\ ivme(i) * zaman^2}{2} \quad (59)$$

- Eğer (ivme süresi < zaman  $\leq$  toplam zaman - ivme süresi)

$$anlık\ açısal\ ivme(i) = 0 \quad (60)$$

$$anlık\ açısal\ hız(i) = maximum\ eklem\ hızı(i) \quad (61)$$

$$\begin{aligned} & anlık\ açısal\ pozisyon(i) \\ & = açısal\ pozisyon(zaman - \Delta zaman) + anlık\ açısal\ hız(i) \\ & * \Delta zaman \end{aligned} \quad (62)$$

- Eğer (zaman  $\geq$  toplam zaman - ivme süresi)

$$anlık\ açısal\ ivme(i) = -Eklem\ ivmesi(i)(sabit) \quad (62)$$

$$\begin{aligned}
& \text{Anlık açısal hız}(i) \\
& = \text{maximum eklem hızı}(i) - \text{anlık açısal ivme}(i) \\
& * (\text{zaman} + \text{ivme süresi} - \text{toplam zaman})
\end{aligned} \tag{63}$$

$$\begin{aligned}
& \text{anlık açısal pozisyon}(i) \\
& = \text{açısal pozisyon}(\text{zaman} - \Delta\text{zaman}) + \text{anlık açısal hız}(i) \\
& * \Delta\text{zaman} - \text{anlık açısal ivme}(i) \\
& * \frac{\Delta\text{zaman}^2}{2}
\end{aligned} \tag{64}$$

## 4.2 Kübik Hız Profili İle Yörünge Planlanması

Daha önce bahsettiğimiz gibi kübik hız profili uygulamanın avantajı sürekli ivme değerinin olmasıdır. Bunun yanı sıra ise toplam hareket süresi azalmaktadır. Trapez hız profili kısmında Çizelge 4.1 'de verilen örnek hareketin aynısı kübik hız profili içinde uygulanmıştır. Şekil 4.4, Şekil 4.5 ve Şekil 4.6 bu örnek hareketin eklem açısal konum, açısal hız ve açısal ivme değerlerinin zamana göre değişimini göstermektedir.

Kübik hız profillerini ve hız profillerine bağlı olarak oluşacak konum ve ivme profillerini belirlemek için aşağıda sıralanmış adımlar uygulanmıştır.

- Adım 1: Ters geometrik modelden elde edilen ilk ve son açısal konum bilgileri trapez yörünge planlamasının girdileri olarak ayarlanır.
- Adım 2: Denklem 65<sup>[15]</sup> 'de hesaplanan olası 12 adet toplam zaman seçeneklerinden maksimum değere sahip olan seçilir ve o değer toplam zaman değeri olarak atanır.

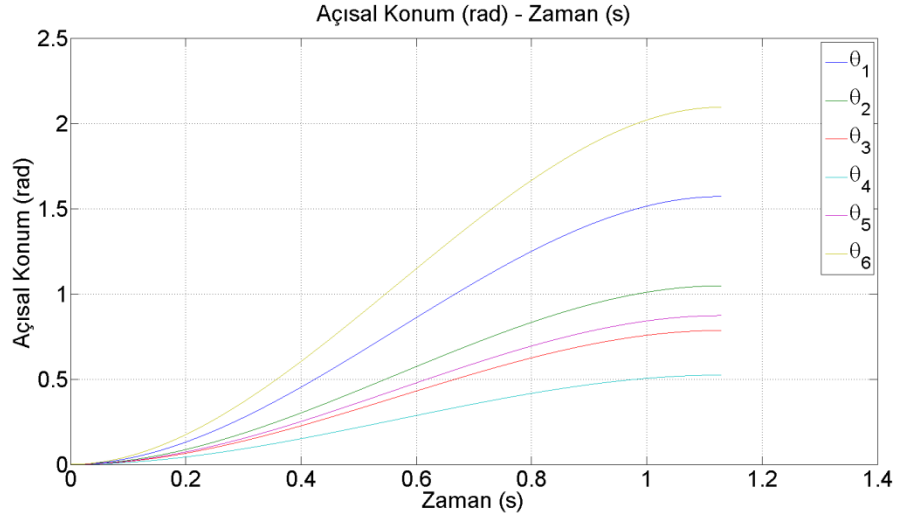
$$\text{toplam zaman} = \max \left( 3 * \frac{|\theta(i)|}{2 * \text{eklem max. açısal hız}(i)}, \frac{\sqrt{6 * |\theta(i)|}}{\text{ektme max. açısal ivme}(i)} \right) \tag{65}$$

- Adım 3: Toplam hareket süresi belirlendiğine göre artık eklemlere uygulanacak maksimum ivme ve hız değerleri denklem 66 ve 67<sup>[15]</sup> 'de görüldüğü gibi hesaplanabilir.

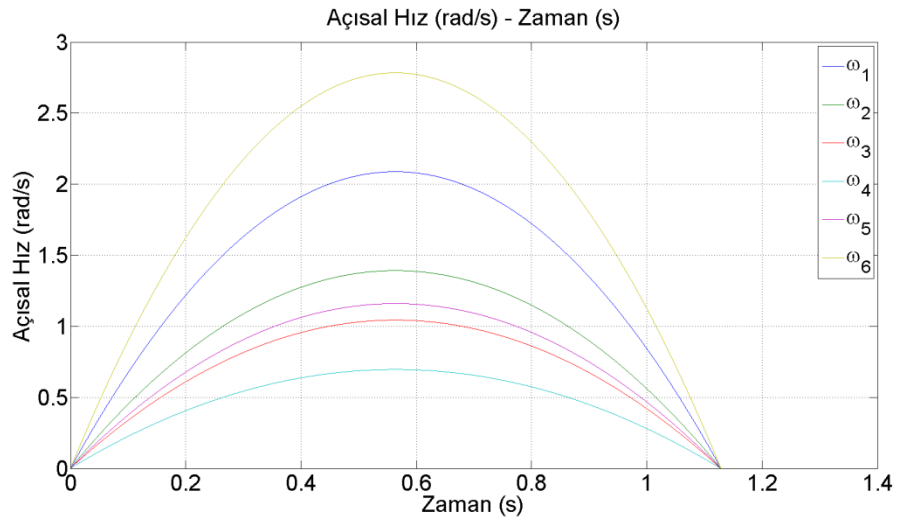
$$max. \text{a} \text{ç} \text{ı} \text{ş} \text{a} \text{l} \text{ i} \text{v} \text{m} \text{e} (i) = \frac{6 * |\theta(i)|}{\text{topl} \text{a} \text{m} \text{ z} \text{a} \text{m} \text{a} \text{n}^2} \quad (66)$$

$$max. \text{a} \text{ç} \text{ı} \text{ş} \text{a} \text{l} \text{ h} \text{i} \text{z} (i) = \frac{3 * |\theta(i)|}{2 * \text{topl} \text{a} \text{m} \text{ z} \text{a} \text{m} \text{a} \text{n}} \quad (67)$$

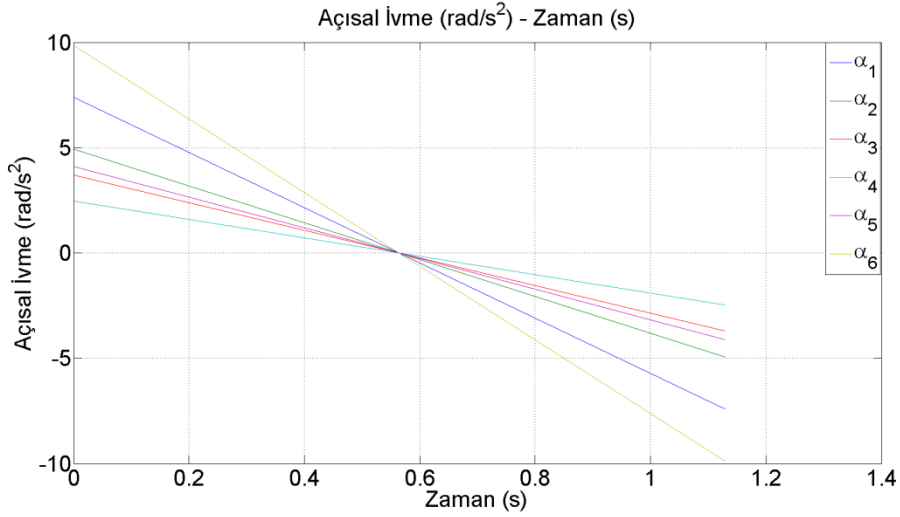
- Adım 4 : Artık bütün eklemlerin maksimum açısal hızları ve ivme değerleri tespit edilmiştir.



Şekil 4.4 : Kübik hız profilinde eklemlerin konum-zaman grafiği örneği.



Şekil 4.5 : Kübik hız profilinde eklemlerin hız-zaman grafiği örneği



Şekil 4.6 : Kübik hız profilinde eklemlerin ivme-zaman grafiği örneği.

- Adım 5 : Sıradaki işlem ise elde ettiğimiz değerler doğrultusunda eklemlerin anlık açısal konum, açısal hız ve açısal ivme değerlerini denklem 68,69 ve 70<sup>[15]</sup> 'de görüldüğü gibi belirlemektir. Anlık değerler her  $\Delta zaman = 0.04s$  aralığında belirlenmiştir.

*anlık açısal pozisyon(i)*

$$= \text{ilk açısal konum} + \left( 3 * \left( \frac{\text{zaman}}{\text{toplam zaman}} \right)^2 - 2 * \left( \frac{\text{zaman}}{\text{toplam zaman}} \right)^3 \right) * \theta(i) \quad (68)$$

$$\text{anlık açısal hız}(i) = \left( 6 * \frac{\text{zaman}}{\text{toplam zaman}^2} - \frac{6 * \text{zaman}^2}{\text{toplam zaman}^3} \right) * \theta(i) \quad (69)$$

$$\text{anlık açısal ivme}(i) = \left( \frac{6}{\text{toplam zaman}^2} - 12 * \frac{\text{zaman}}{\text{toplam zaman}^3} \right) * \theta(i) \quad (70)$$

Kübik hız profili ile yörünge planlanmasının MATLAB<sup>®</sup> kodu Ek F kısmında mevcuttur.

## 5. YAKLAŞIMLI HESAPLANMIŞ TORK İLE PD KONTROLÜ

Tez çalışmasının bu kısmında MATLAB® Simulink® ortamında benzetimi yapılmış olan yaklaşimli hesaplanmış tork kontrolünden bahsedilecektir. Bu benzetimin yapılmasındaki amaç, istediğimiz yörüngeyi, istediğimiz hız ve ivmeyle hareket ettirmemiz için gereken anlık tork değerlerini gözlemlemek ve kapalı sistem kontrol parametrelerini belirlemektir. Ayrıca benzetim sonucunda elde ettiğimiz tork değerlerinin motorlar tarafından gerçekten uygulanabilip uygulanamayacağı da gözlemlenecektir. Bunun sonucunda tasarlanan robot kolunun ve seçilen motor ve dişli mekanizmalarının istenilen performansı sağlayıp sağlamadığı da belirlenir.

Hareket süresince eklemlere uygulanması gereken tork değerlerini tespit etmek için Euler-Lagrange denklemlerinden faydalanılmıştır. Kısaca L-E denklemleri dediğimiz yöntem enerji denklemlerinden faydalanarak tork değerlerinin belirlenmesini sağlamaktadır. Denklem 71, Lagrangian ismini verdiğimiz denklemin kinetik enerji ile potansiyel enerjinin farkını ifade ettiğini göstermektedir.<sup>[13]</sup>

$$L = K - P \quad (71)$$

$K$  : Kinetik Enerji.

$P$  : Potansiyel Enerji.

Kinetik enerjinin hıza göre kısmi türevi momentum eşitliğini verdiğine göre kinetik enerjinin açısal hıza göre kısmi türevi bize açısal momentum değerini verecektir. Momentumun zamana göre değişimi yani türevi nasıl kuvvet eşitliğini veriyorsa, açısal momentumun da zamana göre türevi bize tork bilgisini verecektir. Ayrıca potansiyel enerjinin konuma göre türevi kuvvet bilgisini veriyorsa potansiyel enerjinin açısal konuma göre türevi yine tork bilgisini elde etmemizi sağlayacaktır. Denklem 72, bir cismin var olan kinetik enerjisi ve potansiyel enerjisinden o cismin

o konumunu korumak veya hareketine devam etmesi için ihtiyacı olan tork değerinin nasıl elde edildiğini göstermektedir.<sup>[13]</sup>

İnsansı robot kolunun hareketi esnasında ötelemeden ve dönmeden dolayı oluşan kinetik enerji ifadesi ise denklem 73<sup>[13]</sup> 'de görüldüğü gibidir.

$$\tau = \frac{d}{dt} * \frac{\partial K}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} \quad (72)$$

$$K = \frac{1}{2} * m(i) * V(i)^T * V(i) + \frac{1}{2} * \dot{\theta}^T(i) * I(i) * \dot{\theta}(i) \quad (73)$$

$\dot{\theta}(i)$  : i. eklem açısal hızı.

$I(i)$  : i. eklem eylemsizlik momenti tensörü.

$$I(i) = R(i) * EM(i) * R(i)^T \quad (3x3) \quad (74)$$

$EM(i)$  : i. eklem dönme eksenindeki eylemsizlik momenti (skalar)

$R(i)$  : İlk eklem ile i. eklem arası rotasyon matrisi.

$$V(i) = J_{vi}(\theta) * \dot{\theta} \quad (75)$$

$$\omega = J_{wi}(\theta) * \dot{\theta} \quad (76)$$

$J_{vi}$  : Jakobin matrisinin i. kolonunun ötelemeyi ifade eden ilk 3 değeri.

$J_{wi}$  : Jakobin matrisinin i. kolonunun ötelemeyi ifade eden son 3 değeri.

Eklemlerin açısal hızları ile doğrusal hızları arasındaki ilişkiyi ifade eden jakobian matrisinin bulunuşu tez çalışmasının ilerleyen kısımlarında anlatılacaktır. Yukarıdaki ifadeleri kinetik enerji denkleminde yerine koyarsak denklem 77<sup>[13]</sup> 'deki ifade elde edilir. Potansiyel enerji ifadesi ise denklem 80' de görüldüğü gibidir.

$$K = \frac{1}{2} * \dot{\theta}^T * [\sum_{i=1}^6 (m_i * J_{vi}^T(\theta) * J_{vi}(\theta) + J_{wi}^T(\theta) * R_i(\theta) * I_i * R_i^T(\theta) * J_{wi})] * \dot{\theta} \quad (77)$$

$$D(\theta) = \left[ \sum_{i=1}^6 (m_i * J_{vi}^T(\theta) * J_{vi}(\theta) + J_{wi}^T(\theta) * R_i(\theta) * I_i * R_i^T(\theta) * J_{wi}) \right] \quad (78)$$

$$K = \frac{1}{2} * \dot{\theta}^T * D(\theta) * \dot{\theta} \quad (79)$$

$D(\theta)$  : Eylemsizlik momenti matrisi.

$D(\theta)$  matrisinin belirlenmesi için yazılan MATLAB<sup>®</sup> kodu Ek H kısmında mevcuttur.

$$P_i(\theta) = m_i * g^T * r_{ci}(\theta) \quad (80)$$

$$P(\theta) = \sum_{i=1}^6 P_i(\theta) \quad (81)$$

$$L = K - P = \frac{1}{2} * \sum_{i=1}^6 \left( \sum_{j=1}^6 (d_{i,j}(\theta) * \dot{\theta}_i * \dot{\theta}_j) \right) - P(\theta) \quad (82)$$

Eğer denklem 72 hesaplanırsa toplam tork değerimiz denklem 83<sup>[13]</sup> 'deki gibi ifade edilebilir.

$$\tau = D(\theta)_{6x6} * \ddot{\theta}_{6x1} + C(\theta, \dot{\theta})_{6x6} * \dot{\theta}_{6x1} + g(\theta)_{6x1} \quad (83)$$

$$D(\theta)_{6x6} * \ddot{\theta}_{6x1} = \tau_{a\check{c}isalivme} \quad (84)$$

$\tau_{a\check{c}isalivme}$  : Eylemsizlik momentinden dolayı oluşan tork.

$$C(\theta, \dot{\theta})_{6x6} * \dot{\theta}_{6x1} = \tau_{merkezcil} \quad (85)$$

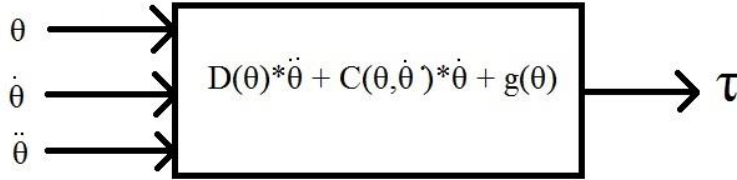
$\tau_{merkezcil}$  : Merkezci ivmeden dolayı oluşan tork.

$$g(\theta)_{6x1} = \frac{\partial P}{\partial \theta} \quad (86)$$

$$g(\theta)_{6x1} = \tau_{yer çekimi} \quad (87)$$

$\tau_{yer çekimi}$  : Yer çekimi ivmesinden dolayı oluşan tork.

Denklem 83 ile denklem 87 arasındaki ifadeler robot kollarında oluşan tork değerlerinin açısal ivmeden yani eylemsizlik momentinden dolayı, merkezci ivmeden dolayı ve yer çekimi ivmesinden yani potansiyel enerji farkından dolayı oluştuğunu göstermektedir. Bu tez çalışmasında merkezci ivme etkisi ihmal edilmiştir. O etkiden dolayı oluşacak hataların kontrol mekanizması tarafından minimize edilmesi hedeflenmiştir. Şekil 5.1 ters dinamik modelin çalışma mantığını göstermektedir.  $g(\theta)$  matrisinin belirlenmesi için yazılan MATLAB<sup>®</sup> kodu Ek I kısmında mevcuttur.



Şekil 5.1 : Ters dinamik model çalışma mantığı.<sup>[13]</sup>

## 5.1 Jacobian Matrisinin Bulunuşu

Jakobian matrisi daha önce de bahsedildiği gibi eklem doğrusal hızları ile son konum doğrusal ve açısal hızları arasındaki ilişkiyi göstermektedir. 6 serbestlik dereceli bir robot tasarımı yaptığımız için jakobian matrisinin boyutu 6x6 olacaktır. Denklem 88 jakobian matrisinin tanımını yapmaktadır.

$$\frac{V_{3x6}}{W_{3*6}} = J_{6x6} * \dot{\theta}_{6x1} \quad (88)$$

$V_{3 \times 6}$  : Uç noktanın doğrusal hız vektörü.

$W_{3 \times 6}$  : Uç noktanın açısal hız vektörü.

$$J_{6 \times 6} = \frac{Jv_{3 \times 6}}{Jw_{3 \times 6}} \quad (89)$$

Bütün eklemlerimiz arasındaki bağlantı şekli sadece dönme hareketi yaptığına göre denklem 90 jacobian matrisinin nasıl belirlendiğini ifade etmektedir. Jacobian matrisinin sembolik sonucu Ek G kısmında mevcuttur.

$$J_i = \begin{matrix} z_{i-1} * (O_n - O_{i-1}) \\ z_{i-1} \end{matrix} \quad (90)$$

$z_{i-1}$  : i. rotasyon matrisinin 3. kolonu ( yani dönme eksenini).

$O_n$  : Uç noktanın konum vektörü.

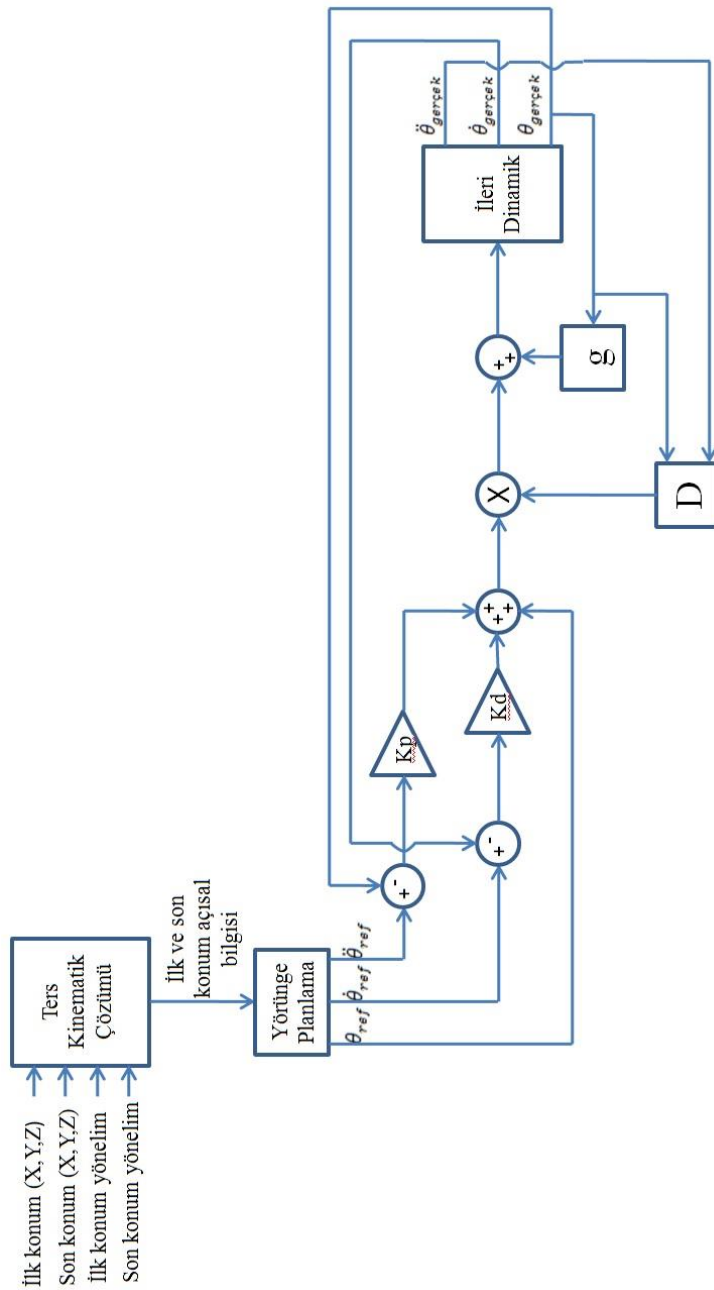
$O_{i-1}$  : (i - 1). eklemin konum vektörü.

## 5.2 SimMechanics™ İle Simulink® Modelinin Oluşturulması

Jacobian matrisinin ve dolayısıyla  $D$  ve  $g$  matrislerinin belirlenmesinden sonra Simulink® ortamında kapalı sistem kontrol yapılabilir. Bunun için ilk olarak SolidWorks® programı ile MATLAB® Simulink® arasındaki ilişkiyi kuran SimMechanics™ eklentisi kullanılmış ve SolidWorks® programında bütün kütle ve ağırlık merkezleri ve montaj ilişkileri belirlenmiş olan tasarımımızın Simulink® modeli elde edilmiştir. Bu sayede artık robot kolunun bütün kütle bilgileri, eylemsizlik bilgileri, montaj ilişkileri, katı model görünümü ile birlikte Simulink® ortamına aktarılmış olur. Simulink® modelinin çalışma prensibi Şekil 5.2 'de görüldüğü gibidir. Simulink® modelinde de görüldüğü gibi kapalı sistem PD kontrolü yapılmaktadır. İşlem sırası aşağıdaki gibidir.

- Adım 1: İlk olarak MATLAB® ana fonksiyonu çalıştırılır. Bu fonksiyonun girdileri ilk ve son konum yönelim bilgileridir.

- Adım 2: Ana fonksiyon ilk olarak ters kinematik fonksiyonuna gider. İlk ve son konum ve yönelimleri için gereken eklem açıları hesaplanır.
- Adım 3: Artık ilk ve son eklem açıları bilindiğine göre seçime bağlı olarak kübik veya trapez hız profili elde eden fonksiyona gidilir. Böylece Simulink® ortamına referans olarak verilecek olan anlık açısal konum, açısal hız ve açısal ivme değerleri belirlenmiş olur.
- Adım 4: Artık Simulink® modeli çalıştırılabilir.



Şekil 5.2 : Simulink® modeli çalışma prensibi.

- Adım 5: Simulink® modelinde ise bütün eklemlere hesaplanılmış tork uygulanmakta ve sonucunda oluşan gerçek açısal konum, açısal hız ve açısal ivme değerleri ölçülmektedir. Böylece hatamıza bağlı olarak PD kontrolü çalışabilir. Hesaplanılmış tork kontrolü, denklem 91<sup>[16]</sup> 'de görüldüğü gibidir.

$$\tau = D(\theta) * (\ddot{\theta}_{ref} + Kp * e + Kd * \dot{e}) + g(\theta) \quad (91)$$

$\ddot{\theta}_{ref}$  : Referans açısal ivme.

$Kp$  : Oransal kazanç.

$Kd$  : Türevsel kazanç.

$$e = \theta_{ref} - \theta_{gerçek} \quad (92)$$

$$\dot{e} = \dot{\theta}_{ref} - \dot{\theta}_{gerçek} \quad (93)$$

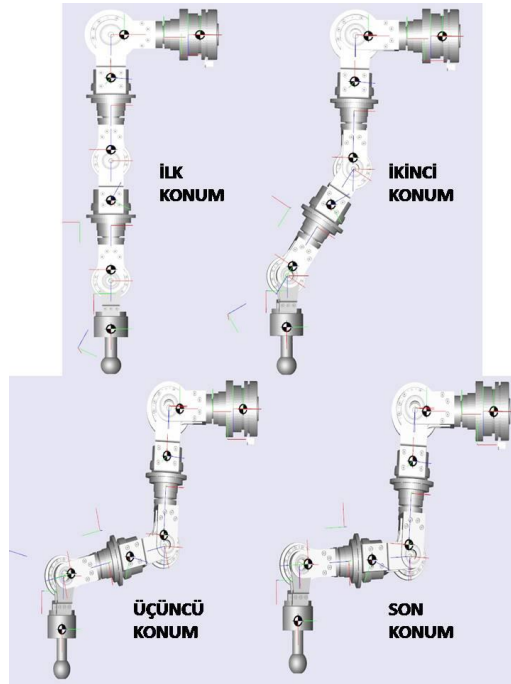
Kontrol sisteminin çalışması prensibi belirlendikten sonra örnek olarak 2 ayrı hareket için oluşan grafikler gösterilebilir. Bu tez çalışmasında 2 ayrı hareket için hem kübik hem trapez hız profilleri denenmiştir. Ayrıca kolun son kısmına ilave yük konularak aynı işlem tekrarlanmıştır. Böylece dinamikleri birbirinden farklı 8 ayrı benzetim örneği sunulacaktır. 2 ayrı hareket Çizelge 5.1 'de tanımlanmıştır.

Birinci ve ikinci hareketin ilk konumlarından son konumlarına doğru giderlerken elde edilen 4 farklı görünümüleri de Şekil 5.3 ve Şekil 5.4 'de görüldüğü gibidir.

Çizelge 5.1 'de belirtilen hareketler için elde ettiğimiz grafikler tez çalışmasının bir sonraki kısmında kendi alt başlıkları altında verilecektir. Her benzetim sonrasında ,ekleme uygulanan yaklaşımlı hesaplanmış tork değerlerinin, gerçek ve referans eklem açısal konum değerlerinin, gerçek ve referans eklem açısal hız değerlerinin, motorların uygulaması gereken tork değerlerinin, motorlara uygulanması gereken akım değerlerinin ve son olarak gerçek ve referans uç konum değerlerinin grafikleri sırasıyla listelenmiştir.

**Çizelge5.1:**Seçilen 2 örnek hareketin değerleri.

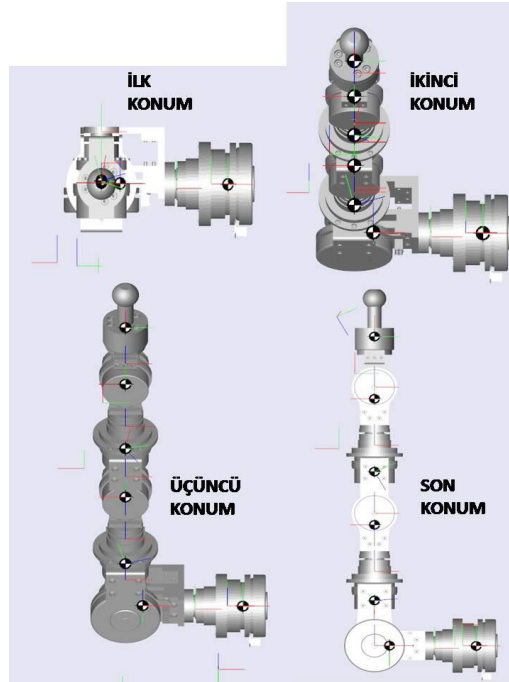
Eklem(i)	Hareket 1 İlk Konum ve Yönelimi	Hareket 1 Son Konum ve Yönelimi	Hareket 2 İlk Konum ve Yönelimi	Hareket 2 Son Konum ve Yönelimi
X (mm)	605,31	400	0	-605,31
Y (mm)	0	50	605,31	0
Z (mm)	75,5	300	75,5	75,5
$\mu$ (rad) z ekseninde ilk dönüş	$-\pi/2$	$-\pi/2$	0	$-\pi/2$
$\beta$ (rad) y ekseninde dönüş	$-\pi/2$	$-\pi/2$	$-\pi/2$	$\pi/2$
$\gamma$ (rad) z ekseninde ikinci dönüş	$\pi/2$	$\pi/2$	$\pi/2$	$-\pi/2$



**Şekil 5.3 :** Birinci hareketin görüntüleri.

Ekleme uygulanan tork değerlerinden motorun uygulaması gereken tork değerlerine geçiş yaparken denklem 94 'deki ifade kullanılmıştır.

Motorlara uygulanması gereken akım değerleri ise denklem 95 'de olduğu gibi bulunmuştur.



Şekil 5.4 : İkinci hareketin görüntüleri.

$$\tau_{motor}(i) = \frac{\tau_{eklem}(i)}{r_{harm.drive}(i)} + I_{harm.drive}(i) * \ddot{\theta}(i) * r_{harm.drive}(i) \quad (94)$$

$r_{harm.drive}(i)$  : i. dişli mekanizmasının çevrim oranı.

$I_{harm.drive}(i)$  : i. dişli mekanizmasının iç kısmının eylemsizlik momenti.

$$A(i) = \frac{\tau_{motor}(i)}{K_t(i)} \quad (95)$$

$A(i)$  : i. motorun akım değeri.

$K_t(i)$  : i. motorun tork sabiti.

Motorların tork sabitleri katalog değerlerinden alınmıştır. Çizelge 5.1 motorların tork sabitlerini göstermektedir.

**Çizelge5.2 : Motorların  $K_t(i)$  tork sabitleri<sup>[11]</sup>**

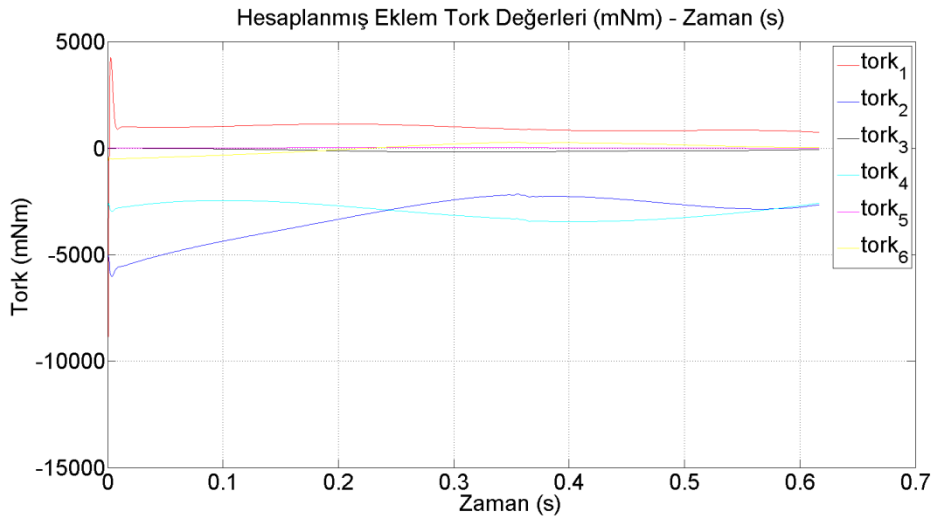
EC 90 tork sabiti (mNm / A)	EC 60 tork sabiti (mNm / A)
70,5	53,4

Kapalı sistem PD kontrolünde kullanılan  $K_p$  ve  $K_d$  diye isimlendirdiğimiz oransal ve türevsel kazanç değerleri benzetim denemeleri sonucunda 550000 ve 1000 olarak belirlenmiştir.

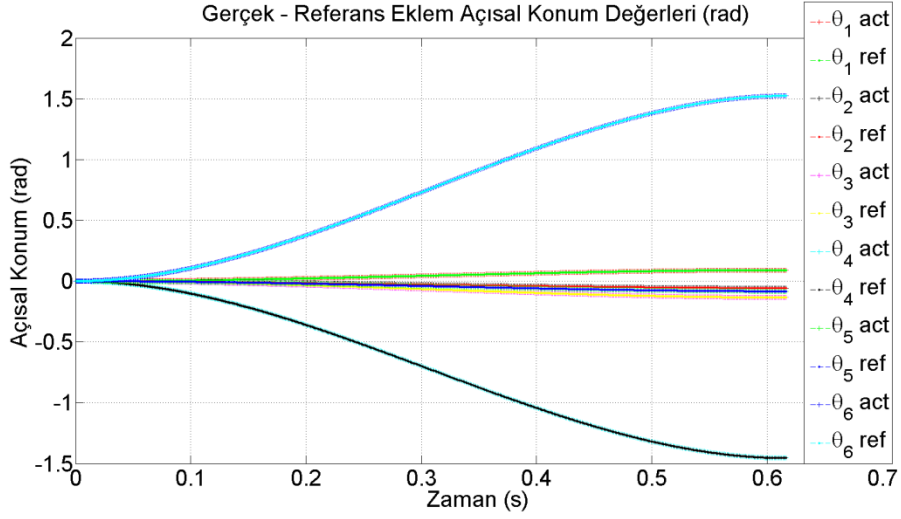
Teorik olarak eğer yaklaşık hesaplanmış tork değerleri içerisinde merkezci ivme etkisi de ilave edilmiş olsa, konum kazancının robot kolunun doğal frekansının karesi, hız kazancının ise onun türevi yani doğal frekansın 2 katı olması gerekir. Fakat, bu çalışmada merkezci ivme göz ardı edildiği için oransal ve türevsel kazançlar teoride olması gerekenden daha farklıdır.

### 5.3 İlave Yüksüz, Birinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar

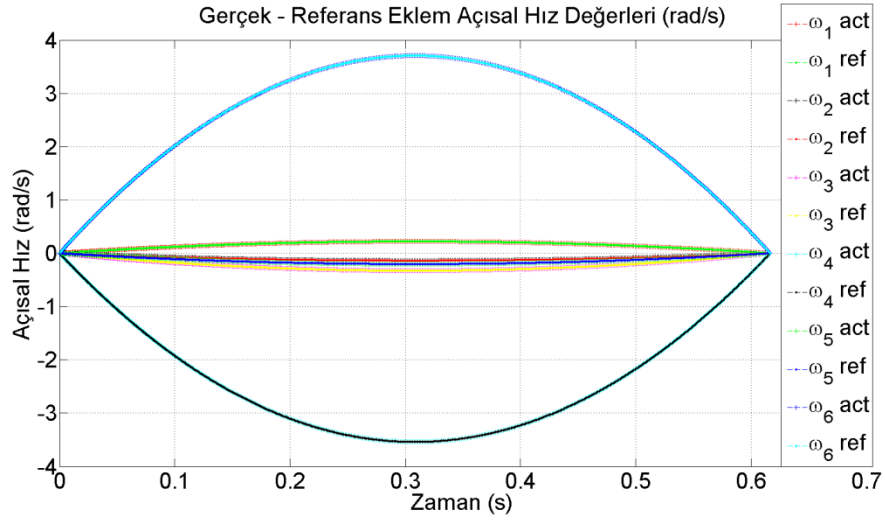
Şekil 5.5 ile Şekil 5.10 arası birinci harekete kübik hız profili uygulandığı zamanki grafikleri göstermektedir. Bu benzetimde kol ucuna herhangi bir ilave yük konulmamıştır. Daha önce de belirtildiği gibi grafikler sırasıyla yaklaşık hesaplanmış eklem tork değerlerini, gerçek ve referans eklem açısız konumlarını, gerçek ve referans eklem açısız hızlarını, yaklaşık hesaplanmış motor torklarını ve akımlarını, gerçek ve referans uç konum koordinatlarını göstermektedir.



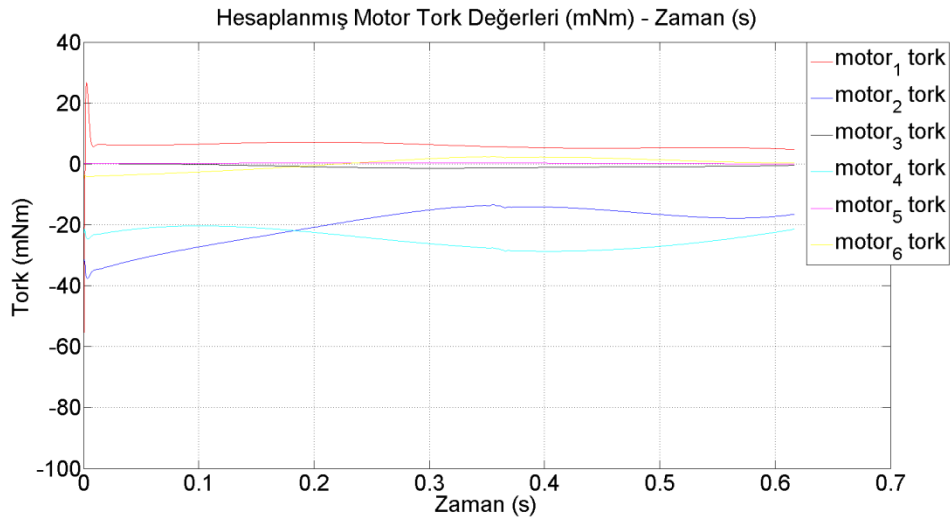
**Şekil 5.5 : Yaklaşımli hesaplanmış eklem tork değerleri (1. hareket, kübik, yüksüz).**



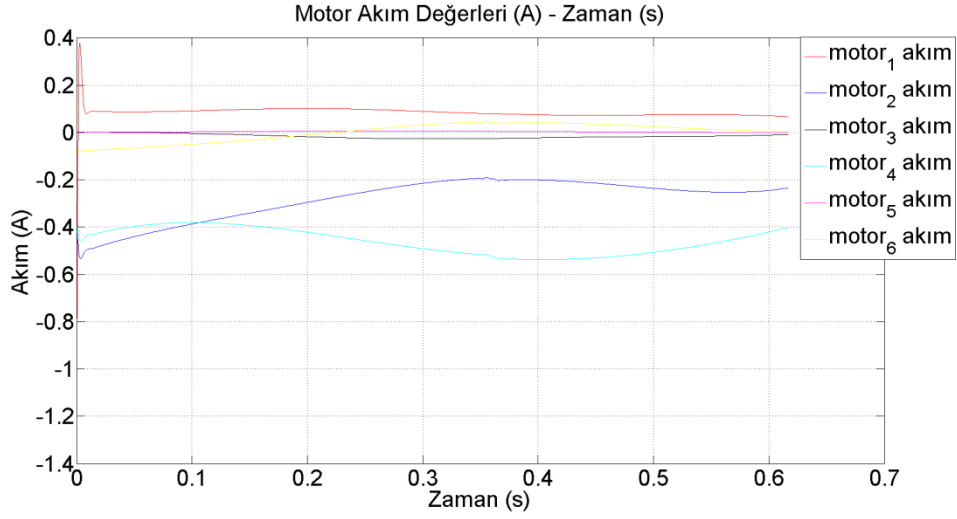
Şekil 5.6 : Gerçek - referans eklem açısal konumları (1. hareket, kübik, yüksüz).



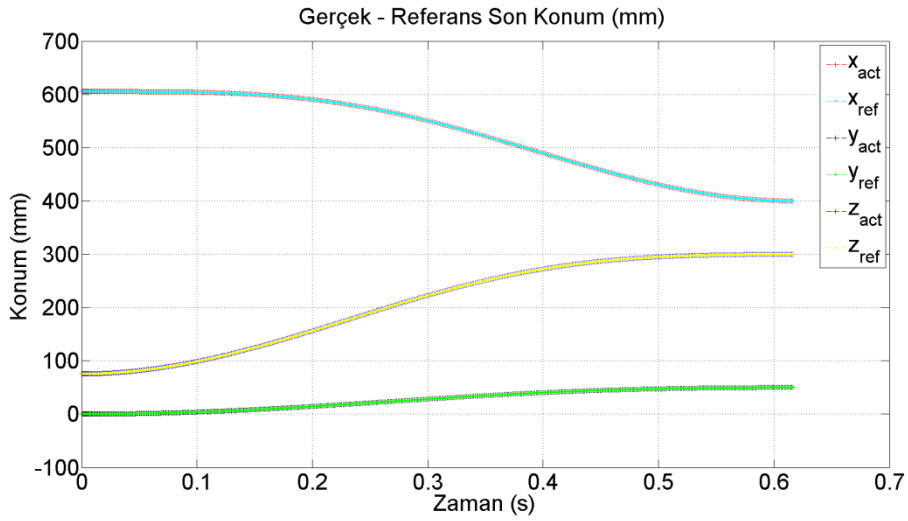
Şekil 5.7 : Gerçek - referans eklem açısal hızları (1. hareket, kübik, yüksüz).



Şekil 5.8 : Yaklaşımli hesaplanmış motor tork değerleri (1. hareket, kübik, yüksüz).



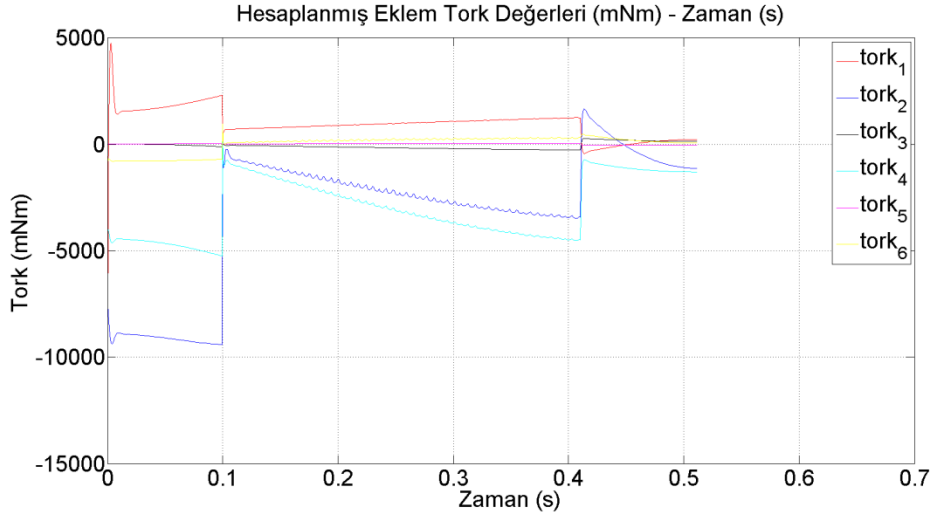
Şekil 5.9 : Motor akım değerleri (1. hareket, kübik, yüksüz).



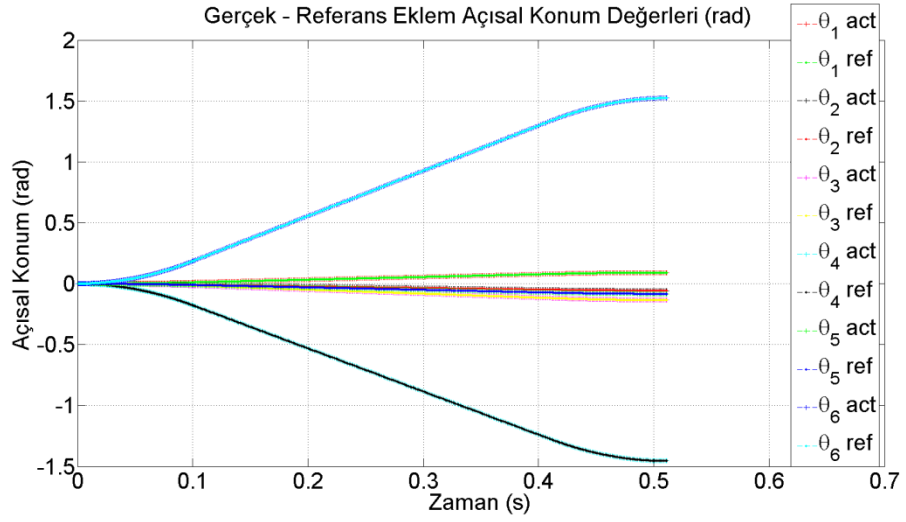
Şekil 5.10 : Gerçek - referans uç konum koordinatları (1. hareket, kübik, yüksüz).

#### 5.4 İlave Yüksüz, Birinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar

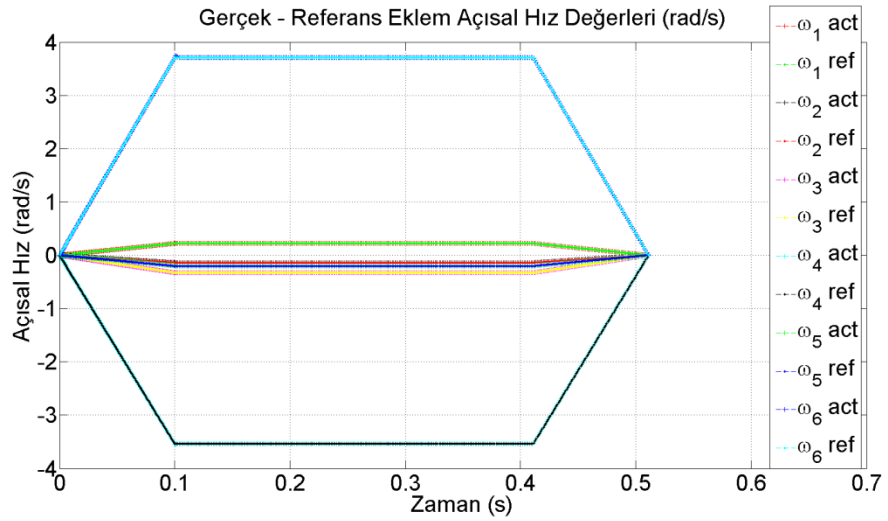
Şekil 5.11 ile Şekil 5.16 arası birinci harekete trapez hız profili uygulandığı zamanki grafikleri göstermektedir. Bu benzetimde kol ucuna herhangi bir ilave yük konulmamıştır.



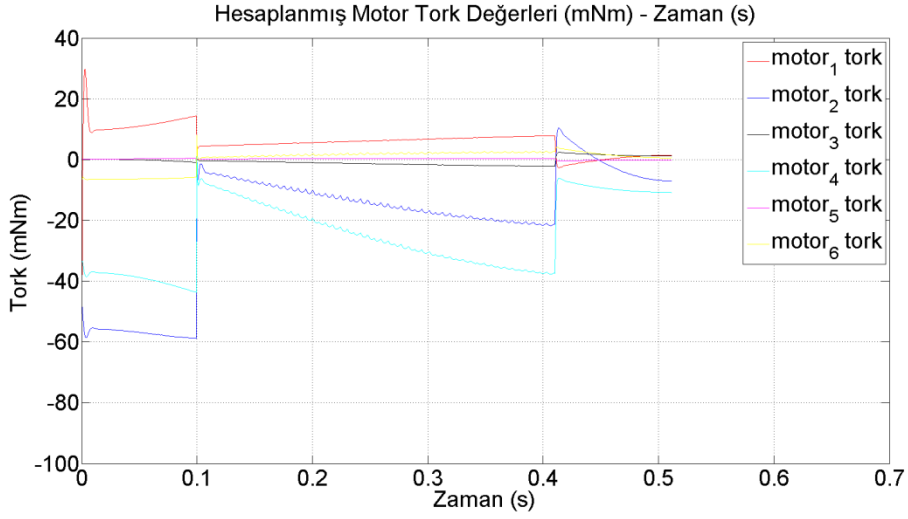
Şekil 5.11 : Yaklaşımli hesaplanmış eklem tork değerleri (1.hareket, trapez, yüksüz).



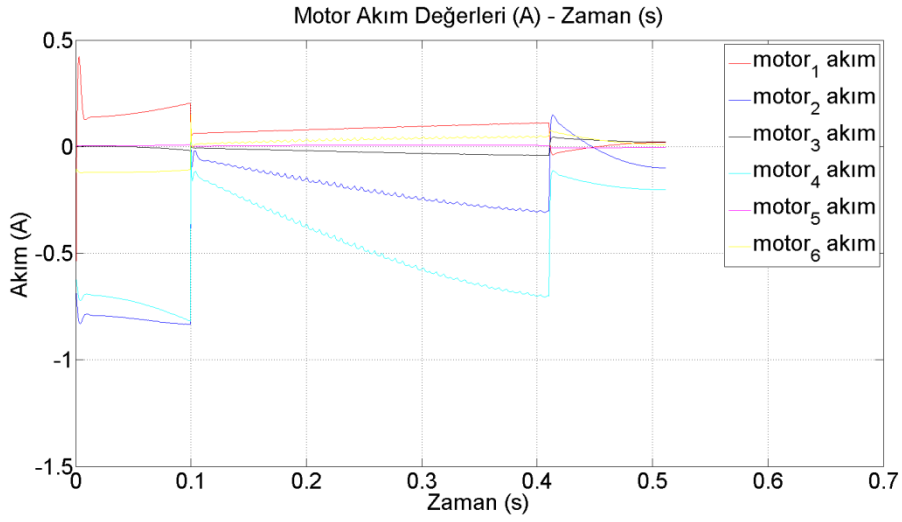
Şekil 5.12 : Gerçek - referans eklem açısıl konumları (1. hareket, trapez, yüksüz).



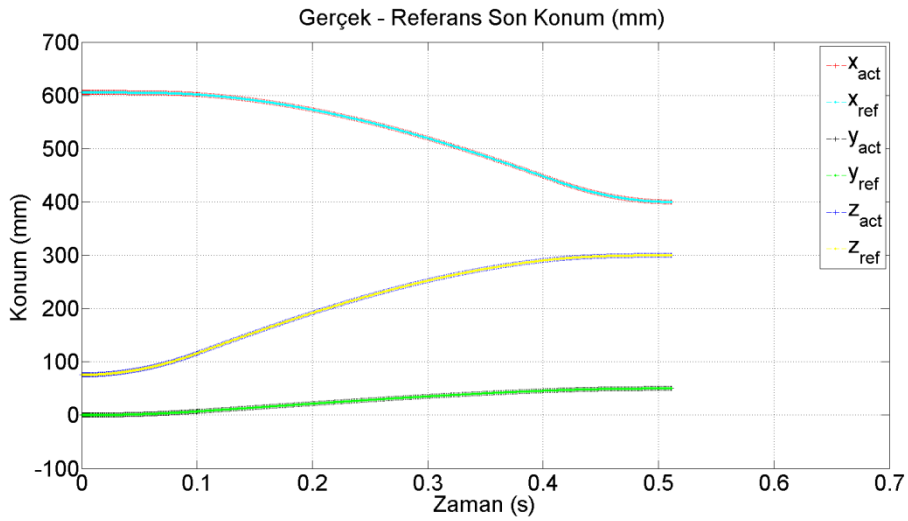
Şekil 5.13 : Gerçek - referans eklem açısıl hızları (1. hareket, trapez, yüksüz).



Şekil 5.14 : Yaklaşımlı hesaplanmış motor tork değerleri (1. hareket, trapez, yüksüz).



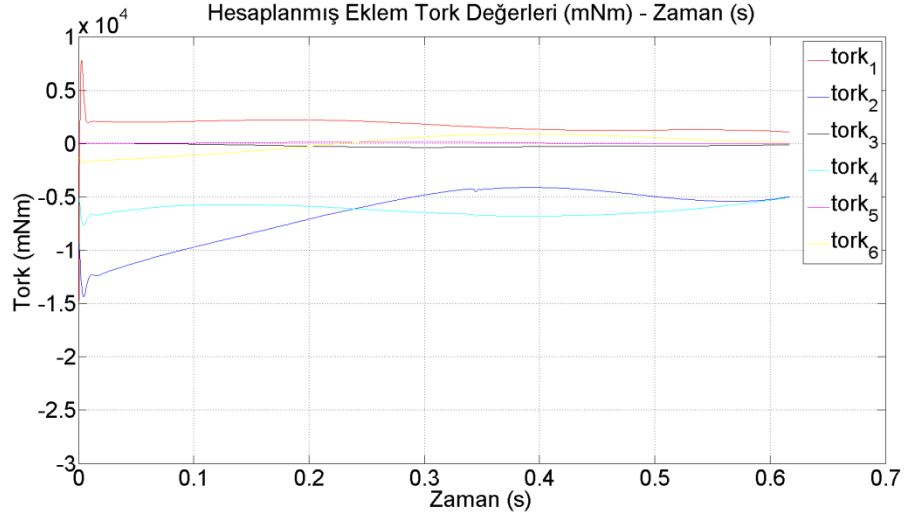
Şekil 5.15 : Motor akım değerleri (1. hareket, trapez, yüksüz).



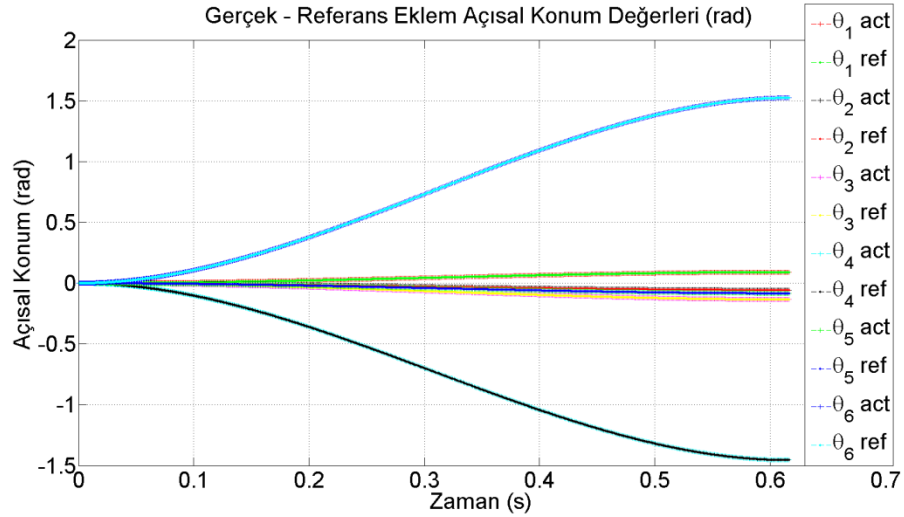
Şekil 5.16 : Gerçek - referans uç konum koordinatları (1. hareket, trapez, yüksüz).

## 5.5 İlave Yükü, Birinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar

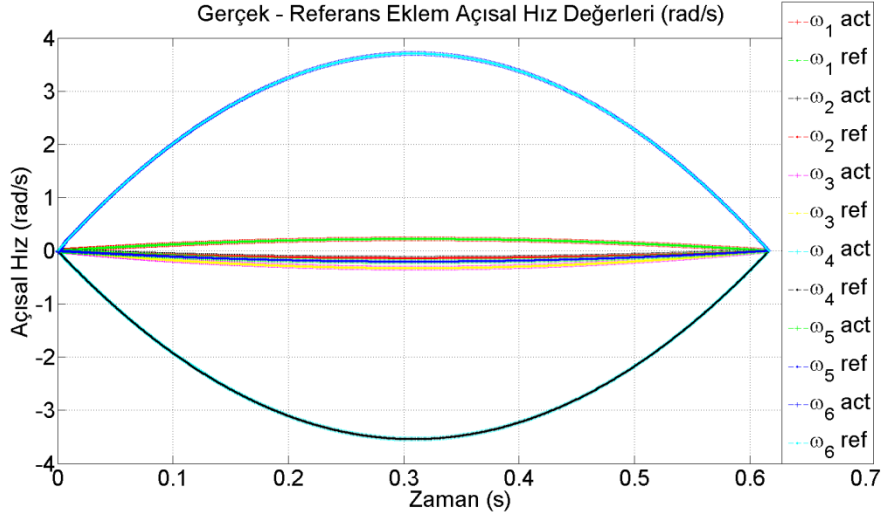
Bu benzetimde kübik hız profili birinci harekete uygulanmıştır. Fakat şimdi kol uç konumuna 3 kg ağırlığında ilave bir yük vardır. Şekil 5.17 ile Şekil 5.22 arasındaki grafikler ilave konulan yükün sonuçları nasıl değiştirdiğini göstermektedir.



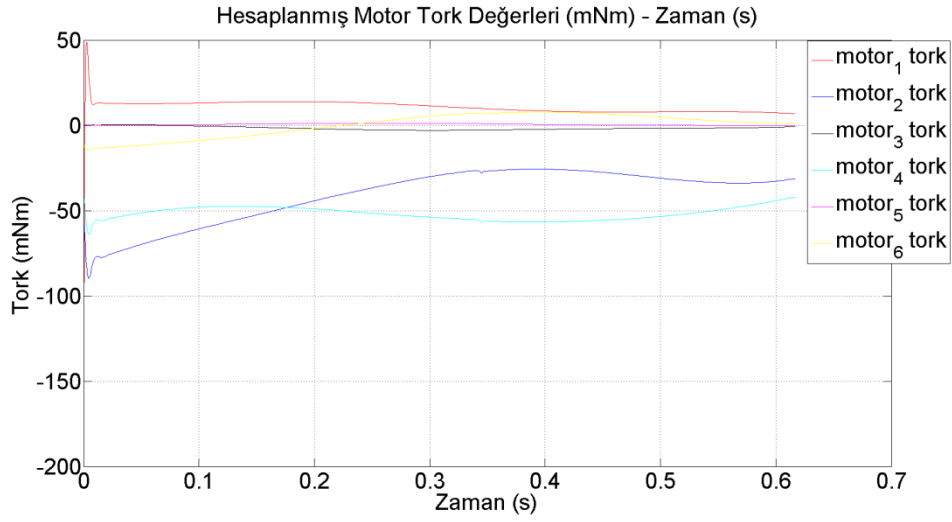
Şekil 5.17: Yaklaşımli hesaplanmış eklem tork değerleri(1.hareket,kübik,3 kg yükü).



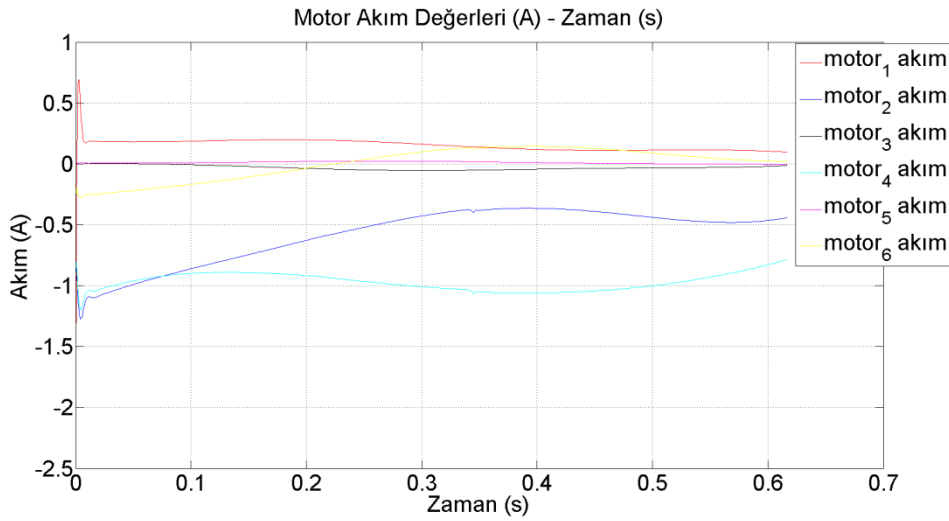
Şekil 5.18 : Gerçek-referans eklem açılal konumları (1.hareket, kübik, 3 kg yükü).



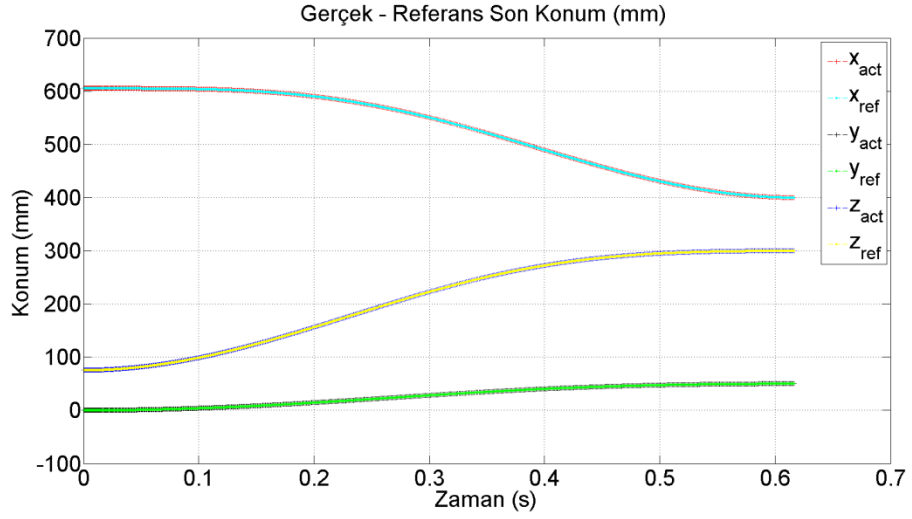
Şekil 5.19 : Gerçek - referans eklem açısal hızları (1. hareket, kübik,3 kg yükü).



Şekil 5.20 :Yaklaşımli hesaplanmış motor tork değerleri(1.hareket,kübik,3 kg yükü).



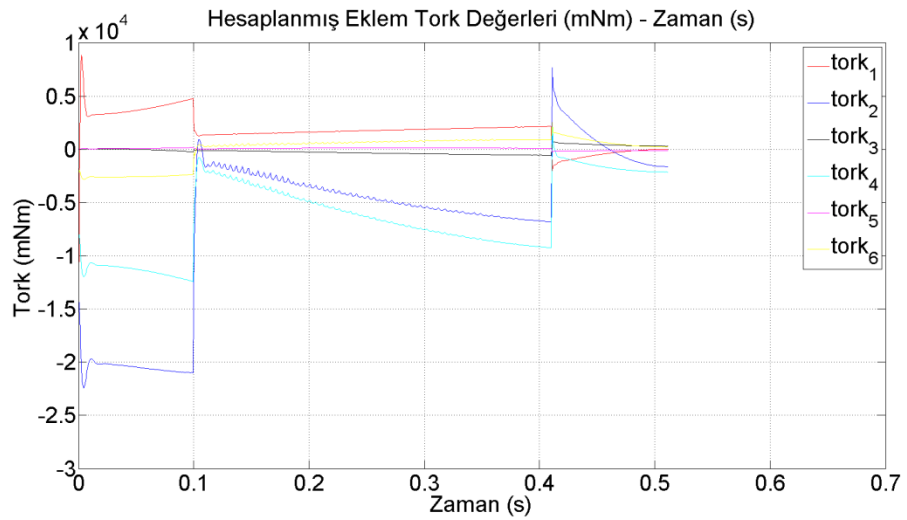
Şekil 5.21 : Motor akım değerleri (1. hareket, kübik, 3 kg yükü).



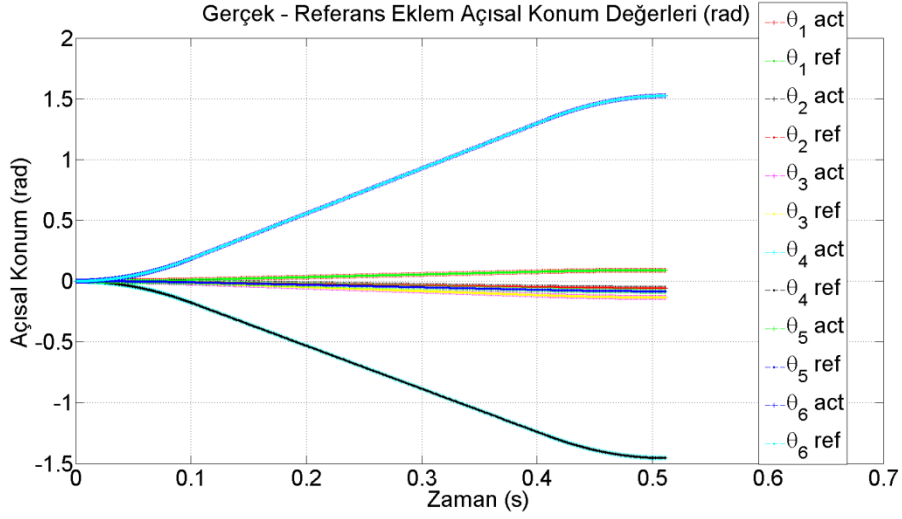
Şekil 5.22 : Gerçek - referans uç konum koordinatları (1. hareket, kübik, 3 kg yük).

## 5.6 İlave Yük, Birinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar

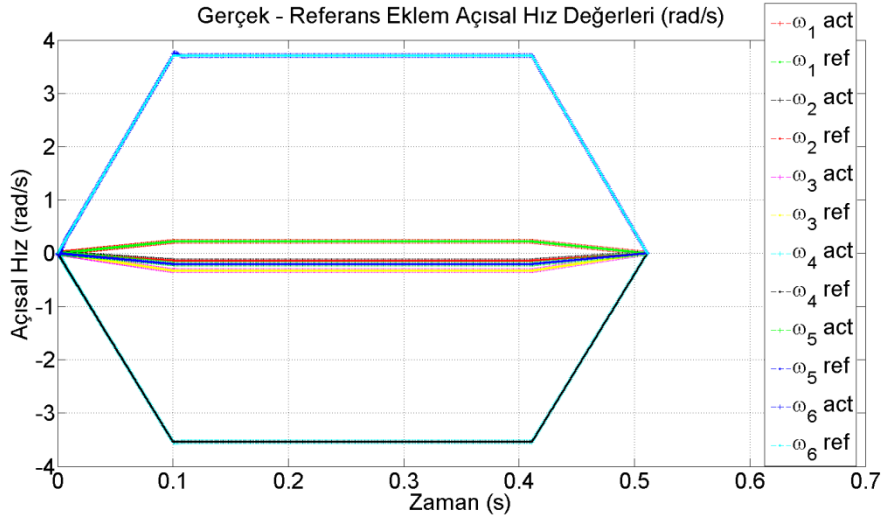
Bu benzetimde ise bir önceki benzetimden farklı olarak aynı yük altında 1. harekete trapez hız profili uygulanmıştır. Şekil 5.23 ile Şekil 5.28 arası grafikler bu benzetimin sonuçlarını göstermektedir.



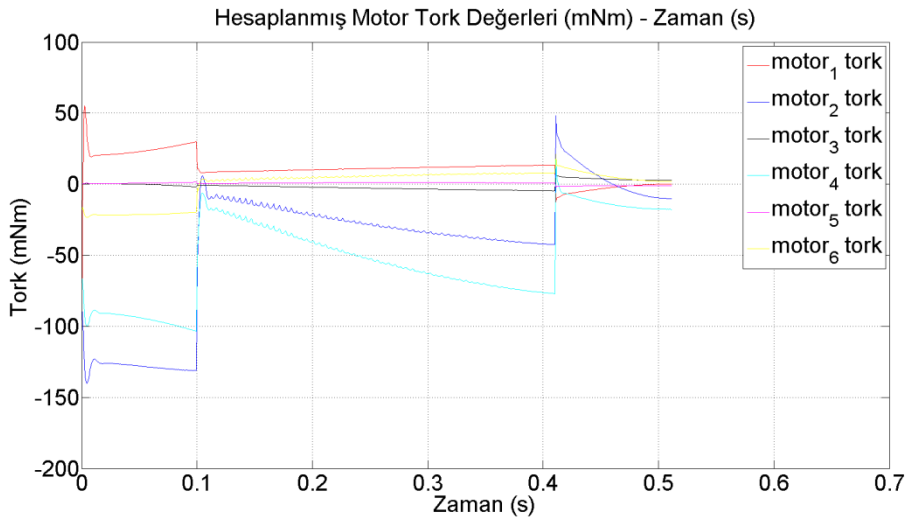
Şekil 5.23: Yaklaşımli hesaplanmış eklem tork değerleri(1.hareket, trapez, 3 kg yük).



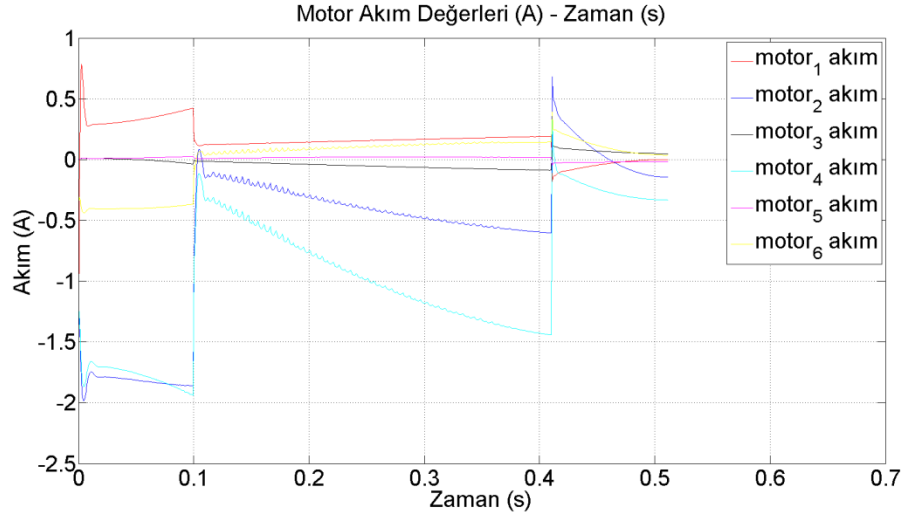
Şekil 5.24 : Gerçek-referans eklem açısal konumları (1.hareket, trapez,3 kg yükü).



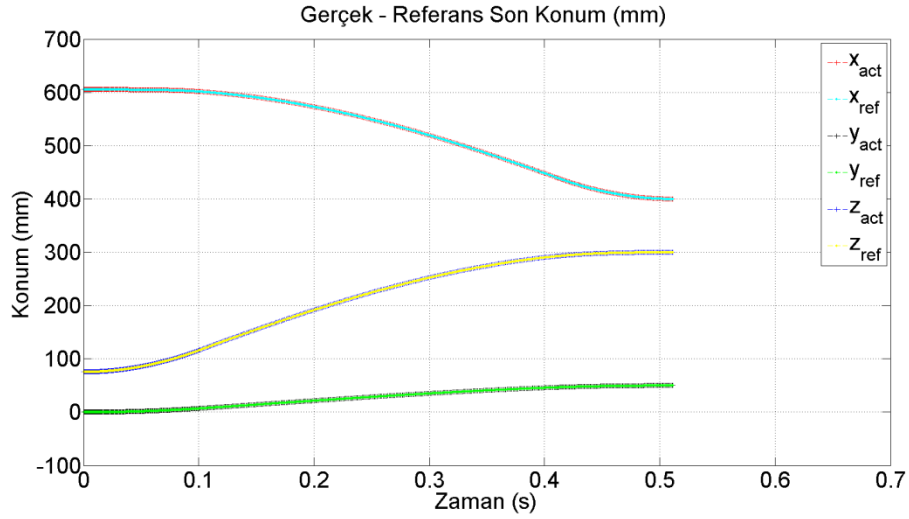
Şekil 5.25 : Gerçek - referans eklem açısal hızları (1. hareket, trapez,3 kg yükü).



Şekil 5.26: Yaklaşımli hesaplanmış motor tork değerleri(1.hareket, trapez,3 kg yükü).



Şekil 5.27 : Motor akım değerleri (1. hareket, trapez, 3 kg yükü).

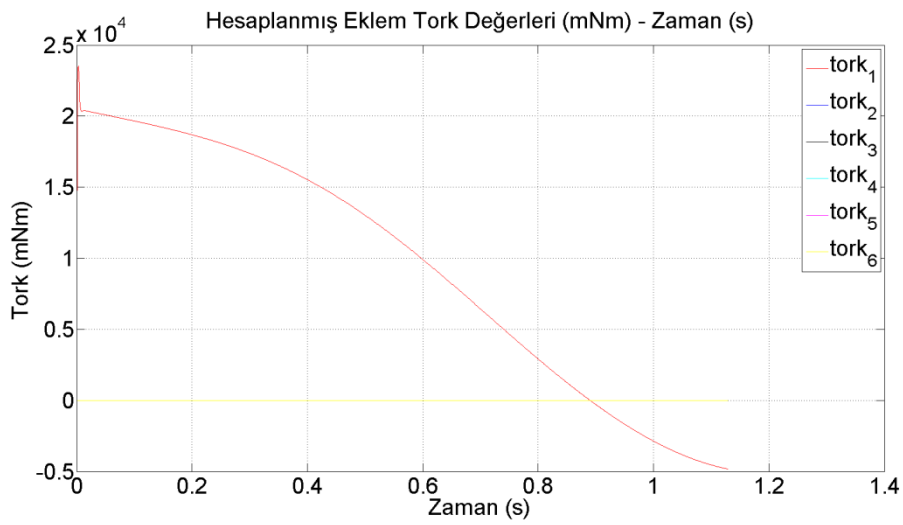


Şekil 5.28 : Gerçek - referans uç konum koordinatları(1. hareket, trapez, 3 kg yükü).

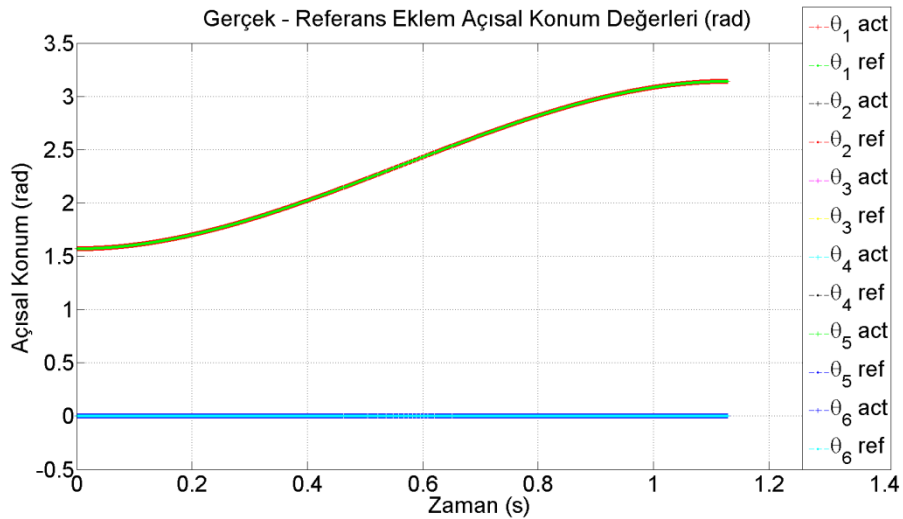
## 5.7 İlave Yüksüz, İkinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar

Tez çalışmasının devam eden kısımlarında sunulacak grafikler ikinci harekete ait grafiklerdir. Bu hareketin benzetimlerini yaparken ilk amaç en fazla yükü taşıyan ilk motora uygulanabilecek maksimum torku uygulayan hareketi gözlemlemektir. Bu sebeple hareket sadece ilk motora verilecektir. Şekil 5.4 ikinci hareketi göstermektedir.

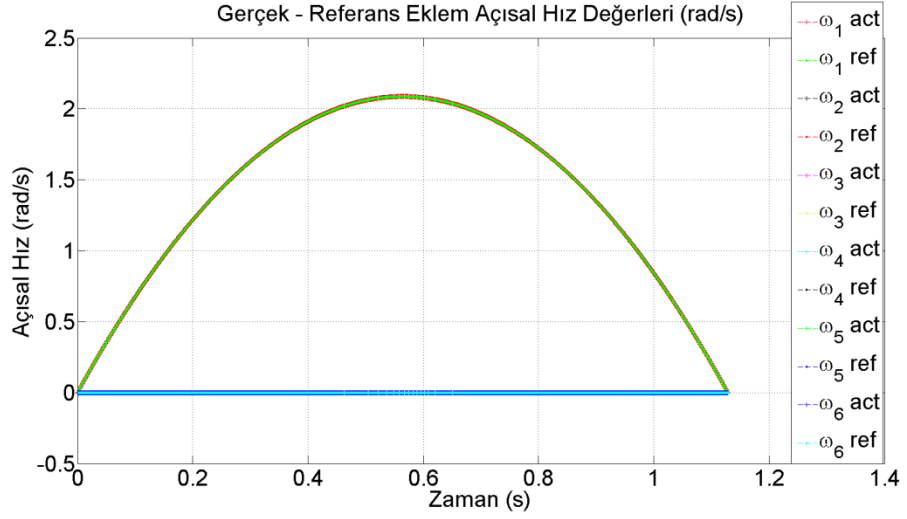
Bu benzetimde robot koluna gelebilecek maksimum torku gözlemek için yapılan ikinci harekete kübik hız profili uygulanmıştır. İlave bir yük konulmamıştır. Şekil 5.29 ile Şekil 5.30 arası grafikler bu benzetime ait olanlardır. Daha önce de bahsedildiği gibi, ikinci hareketteki amaç maruz kalınabilecek maksimum tork değerini gözlemlemeye çalışmaktır. Bu sebeple, en kötü senaryonun benzetimi yapılmaya çalışılmıştır. Bunun için en fazla yüke ve eylemsizlik değerine maruz kalan ilk eklem göz önünde tutulmuştur. Hareket sadece ilk ekleme gerçekleşmektedir. Kol yatay konumdan yukarı doğru dikey konuma doğru ilerlemektedir.



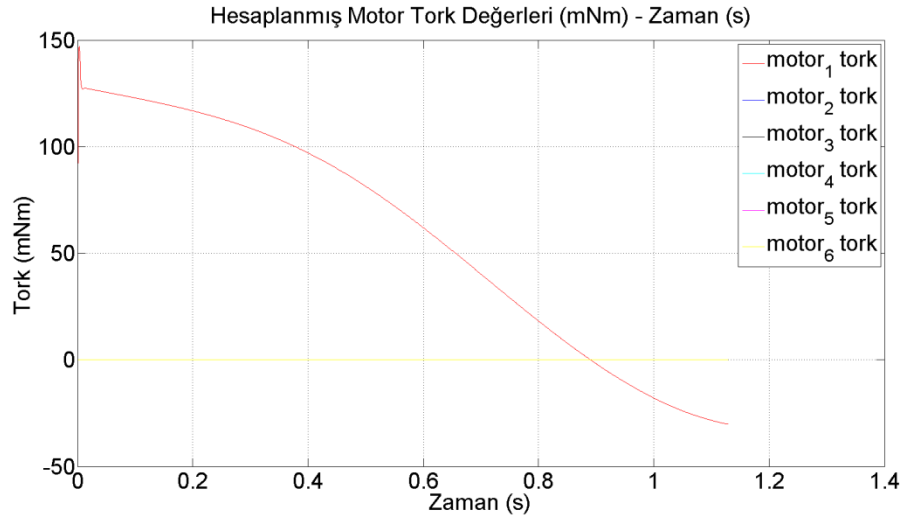
Şekil 5.29 : Yaklaşımlı hesaplanmış eklem tork değerleri (2. hareket, kübik, yüksüz).



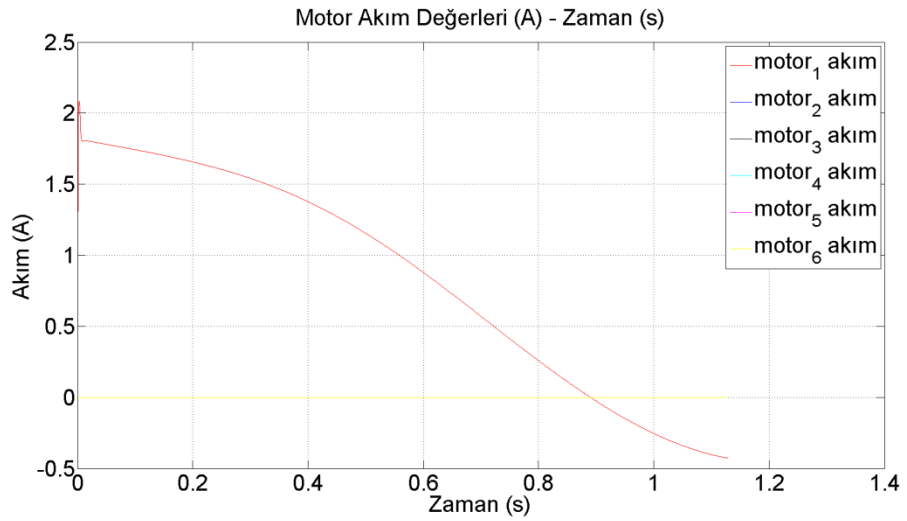
Şekil 5.30 : Gerçek-referans eklem açısal konumları (2.hareket, kübik,yüksüz).



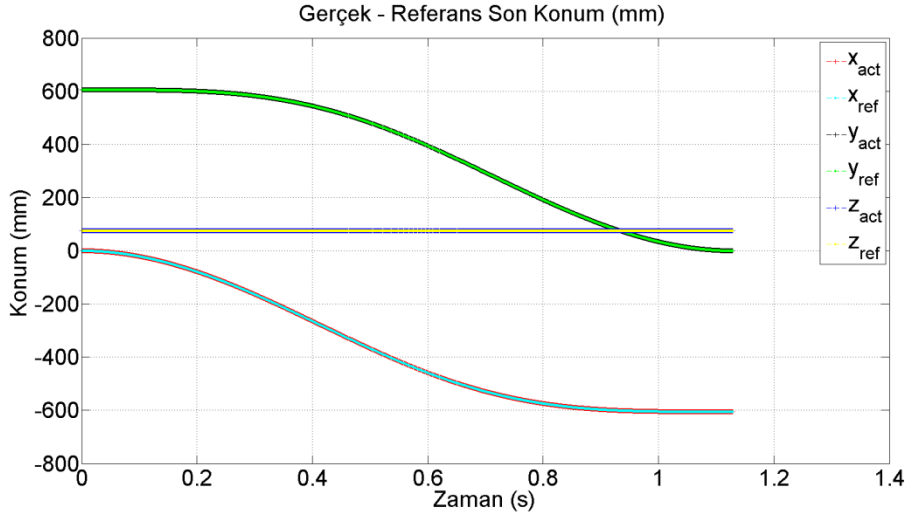
Şekil 5.31 : Gerçek - referans eklem açıl hızları (2. hareket, kübik, yüksüz).



Şekil 5.32 : Yaklaşımlı hesaplanmış motor tork değerleri (2. hareket, kübik, yüksüz).



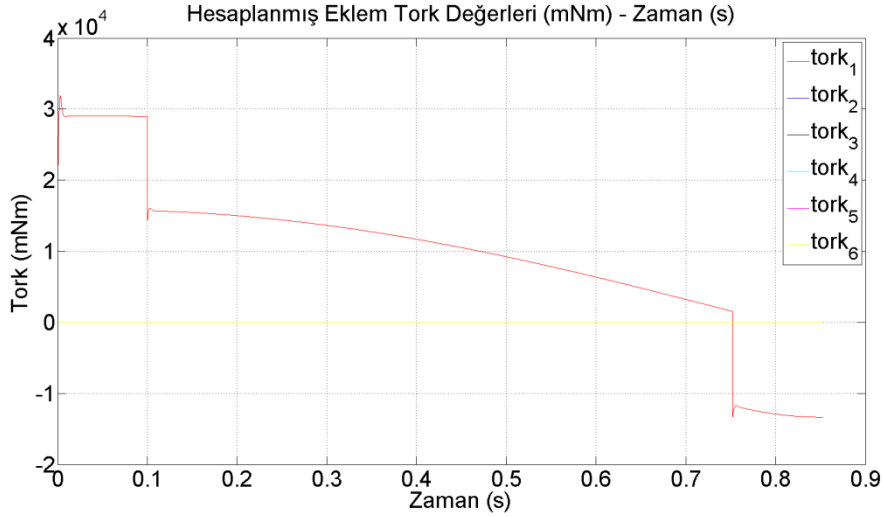
Şekil 5.33 : Motor akım değerleri (2. hareket, kübik, yüksüz).



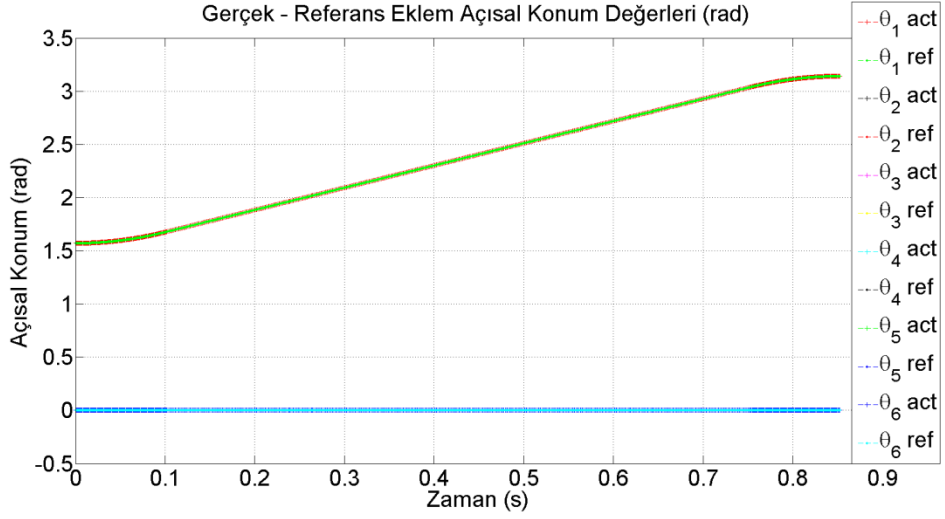
Şekil 5.34 : Gerçek - referans uç konum koordinatları (2. hareket, kübik, yüksüz).

## 5.8 İlave Yüksüz, İkinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar

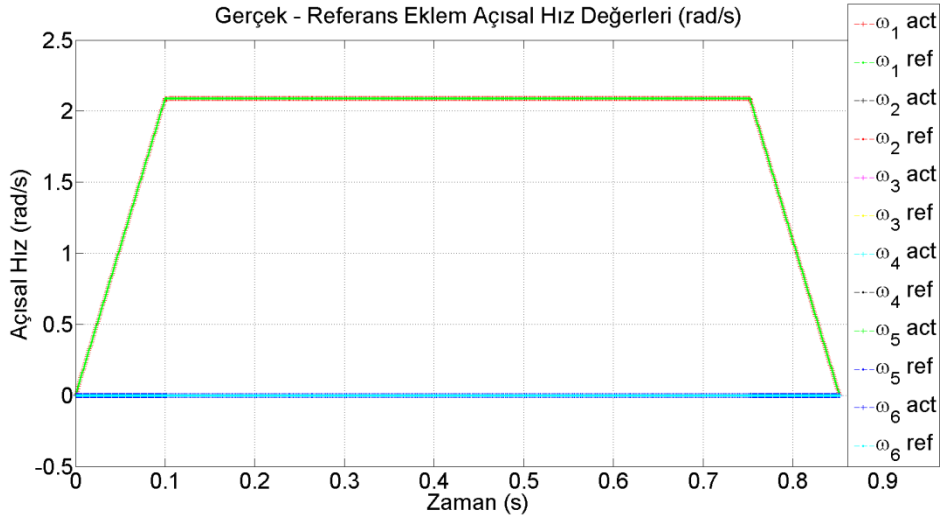
Şekil 5.35 ile Şekil 5.40 arasındaki grafikler ikinci hareketin trapez hız profili uygulandığı zamanki sonuçlarını sunmaktadır. Bu benzetimde kol ucuna ilave yük konulmamıştır.



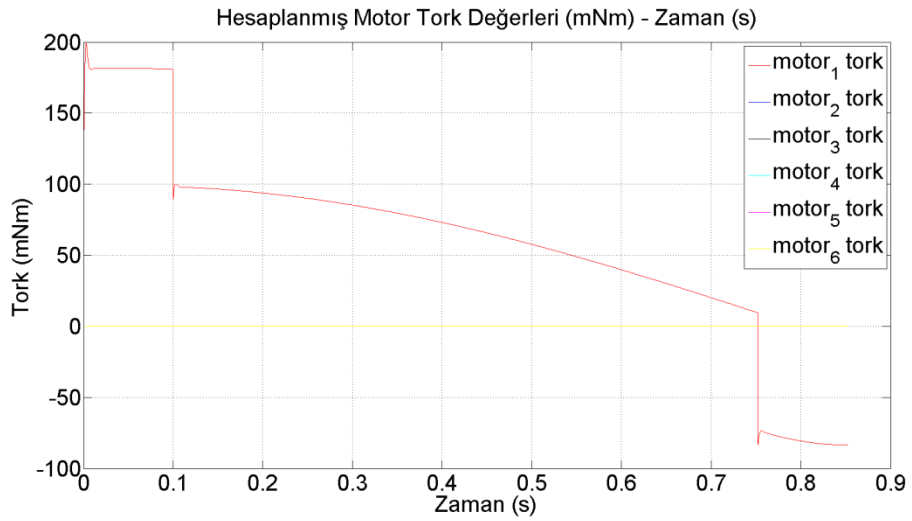
Şekil 5.35 : Yaklaşımlı hesaplanmış eklem tork değerleri (2.hareket, trapez, yüksüz).



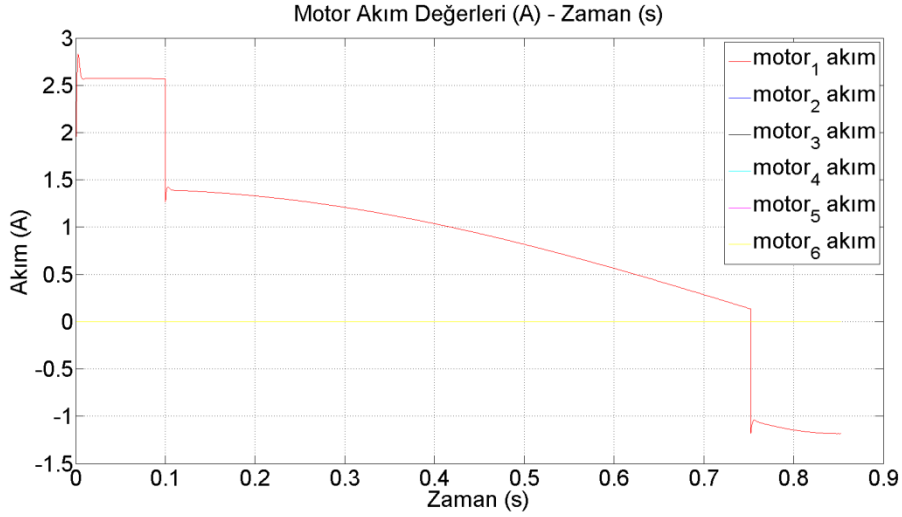
Şekil 5.36 : Gerçek-referans eklem açısal konumları (2.hareket, trapez,yüksüz).



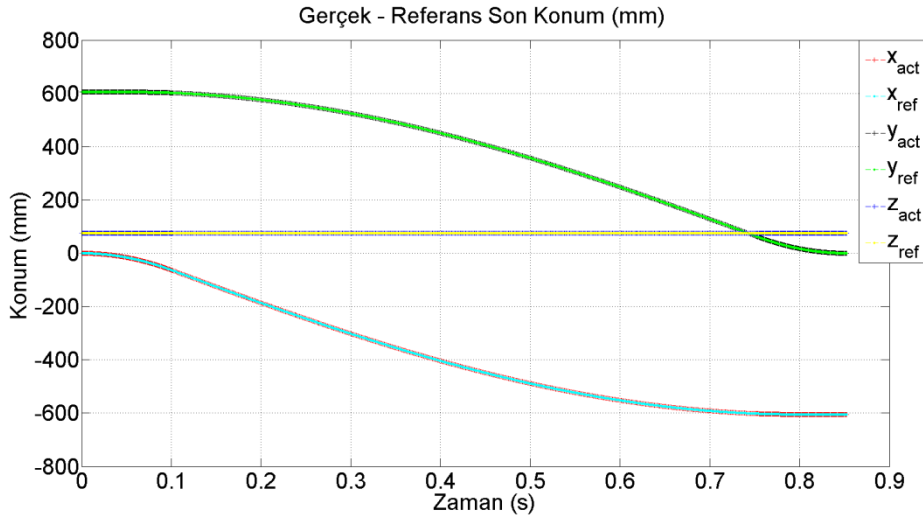
Şekil 5.37 : Gerçek - referans eklem açısal hızları (2. hareket, trapez,yüksüz).



Şekil 5.38 : Yaklaşımli hesaplanmış motor tork değerleri (2. hareket, trapez, yüksüz).



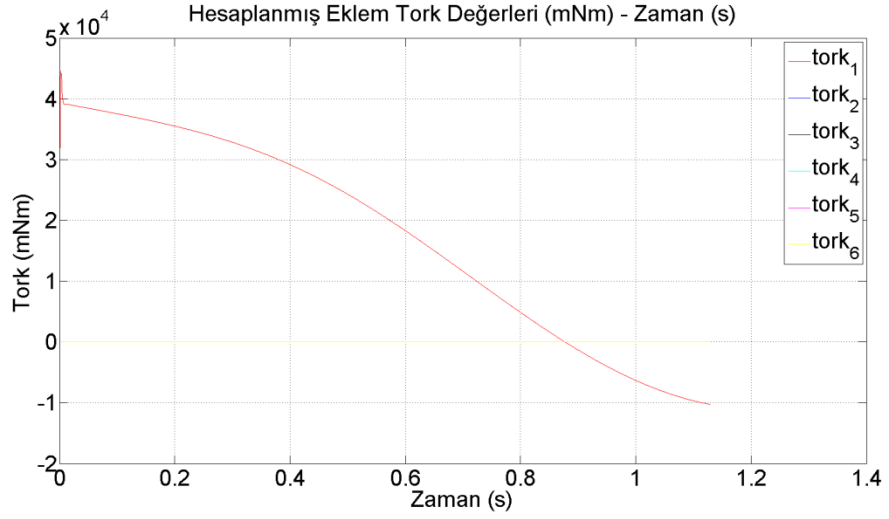
Şekil 5.39 : Motor akım değerleri (2. hareket, trapez, yüksüz).



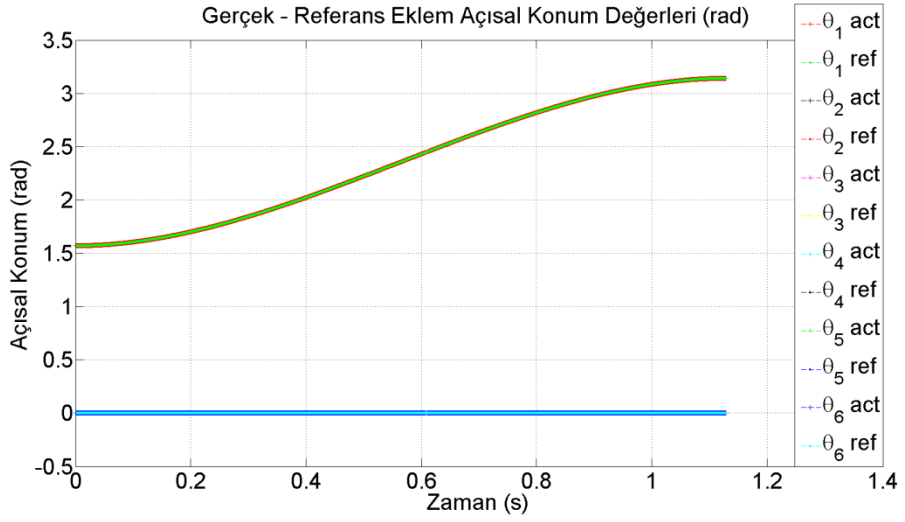
Şekil 5.40 : Gerçek - referans uç konum koordinatları (2. hareket, trapez , yüksüz).

## 5.9 İlave Yükle, İkinci Harekete Kübik Hız Profili Uygulanmış Sonuçlar

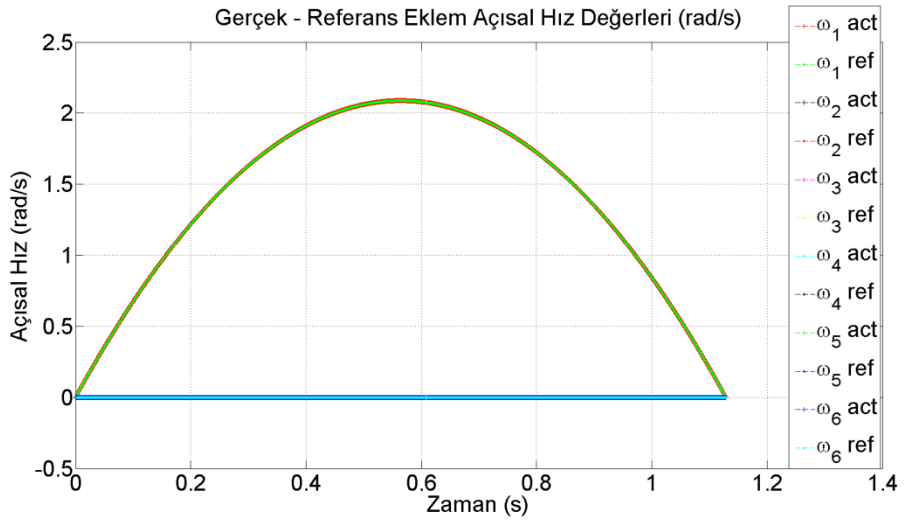
Bu benzetimdeki amaç ise aynı birinci harekette olduğu gibi ikinci hareketinde kol ucuna ilave edilen yükten nasıl etkilendiğini gözlemlemektir. 3 kg ağırlığında ilave bir yük kol ucuna eklenmiştir. İkinci hareket ilk motorun en fazla torka ihtiyaç duyduğu hareketlerden biridir ve bunun yanına ilave yükte konulduğu için bu benzetim en zor koşullardan birini gözlemlememizi sağlamaktadır. Şekil 5.41 ile Şekil 5.46 arası grafikler bu benzetimin sonuçlarıdır.



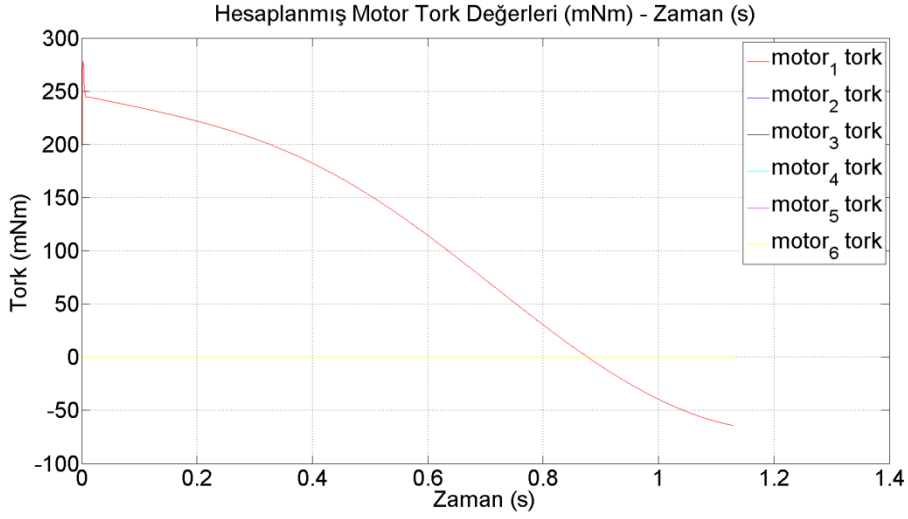
Şekil 5.41 : Yaklaşımlı hesaplanmış eklem tork değerleri(2.hareket,kübik,3 kg yükü).



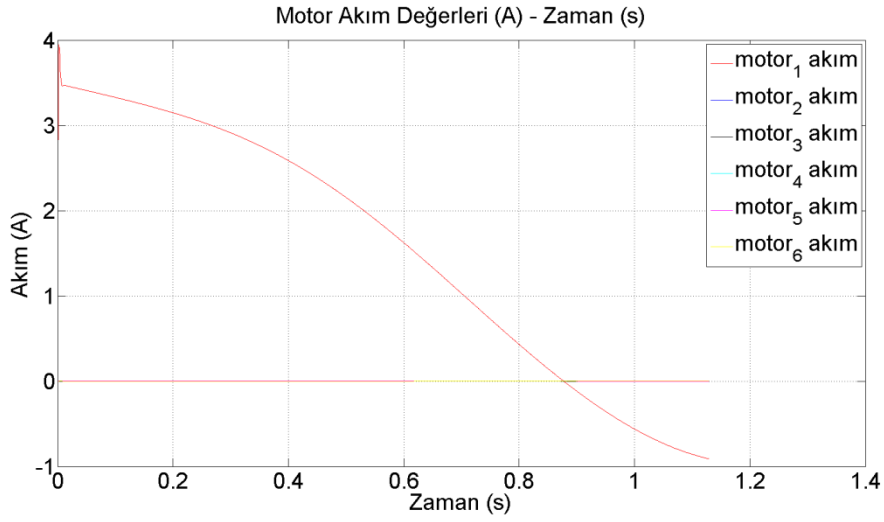
Şekil 5.42 : Gerçek-referans eklem açısai konumları (2.hareket, kübik, 3 kg yükü).



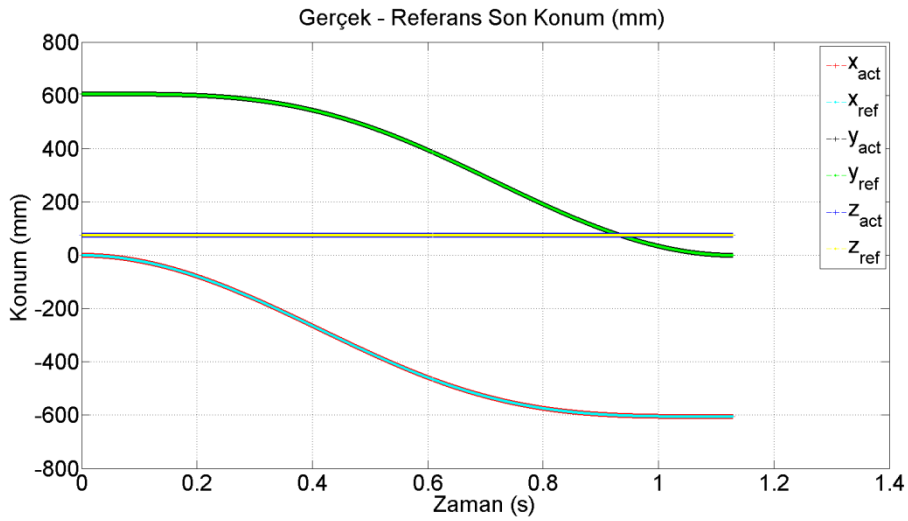
Şekil 5.43 : Gerçek - referans eklem açısai hızları (2. hareket, kübik, 3 kg yükü).



Şekil 5.44 : Yaklaşımli hesaplanmış motor tork değerleri(2.hareket,kübik,3 kg yükü).



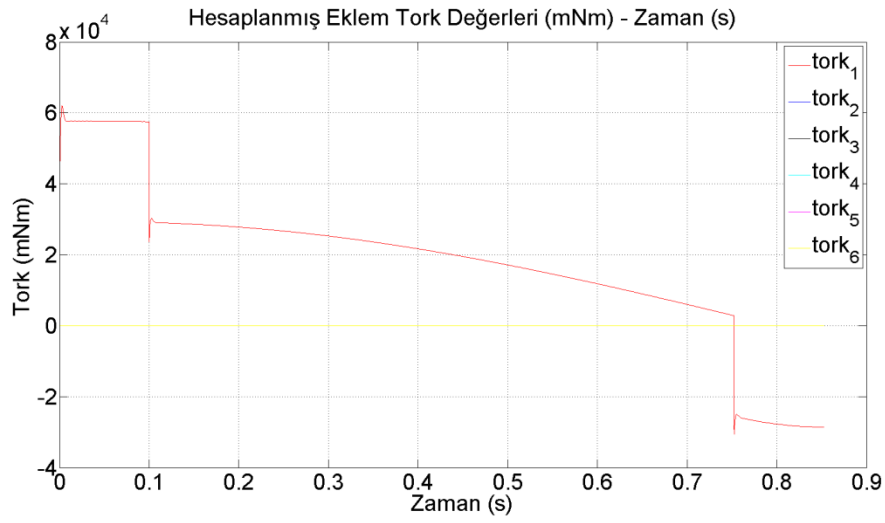
Şekil 5.45 : Motor akım değerleri (2. hareket, kübik, 3 kg yükü).



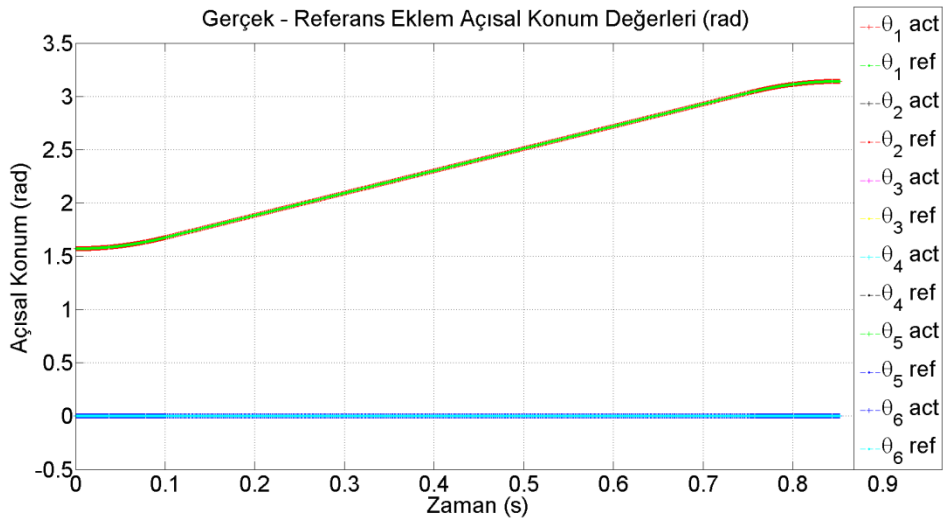
Şekil 5.46 : Gerçek-referans uç konum koordinatları (2. hareket, kübik , 3 kg yükü).

## 5.10 İlave Yükle, İkinci Harekete Trapez Hız Profili Uygulanmış Sonuçlar

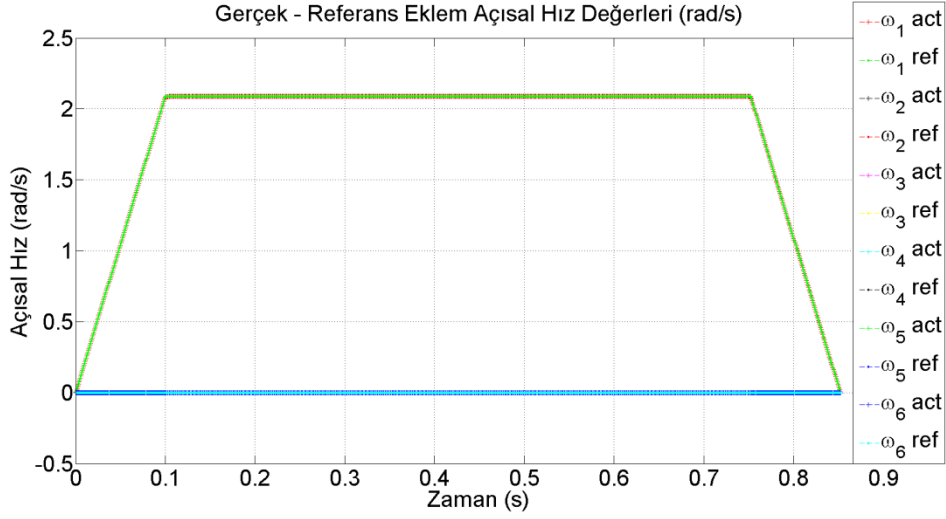
Bu benzetimde artık robotun ilk motoruna uygulanabilecek maksimum tork görülmektedir. İkinci hareket ilk motoru en çok zorlayan hareketlerden biridir. Bunun üzerine kol ucuna yük konulmuş ve trapez hız profili uygulanmıştır. Trapez hız profili de ivme değerinin en yüksek görüldüğü ve süreksiz ivme içerdiği için gereken tork miktarını arttıracaktır. Bu sebeple, bu benzetim bize gereken tork açısından en kötü senaryoyu sunmaktadır. Şekil 5.47 ile Şekil 5.52 arası grafikler bu benzetimin sonuçlarıdır.



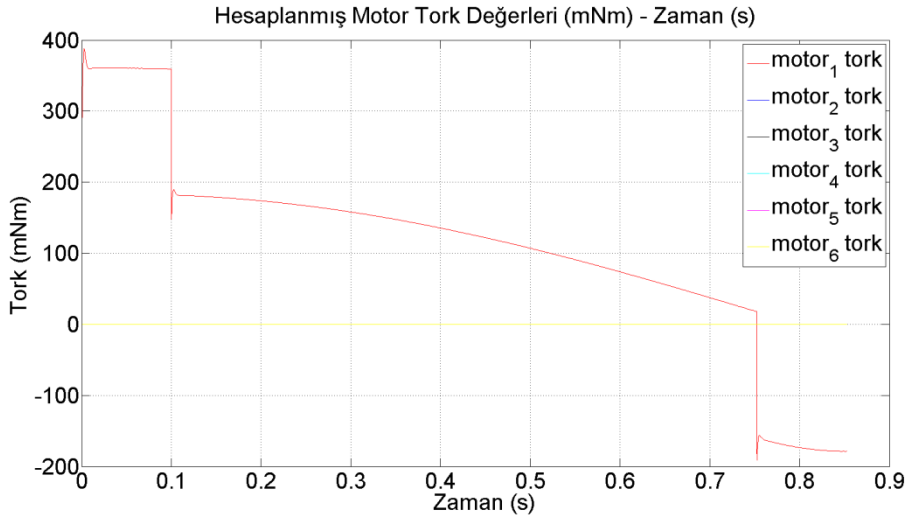
Şekil 5.47: Yaklaşımlı hesaplanmış eklem tork değerleri(2.hareket, trapez, 3 kg yük).



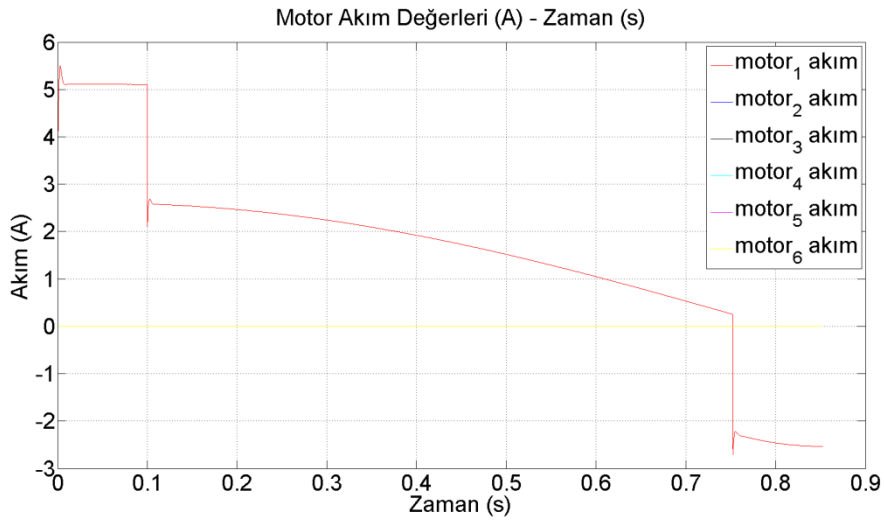
Şekil 5.48 : Gerçek-referans eklem açısal konumları (2.hareket, trapez, 3 kg yük).



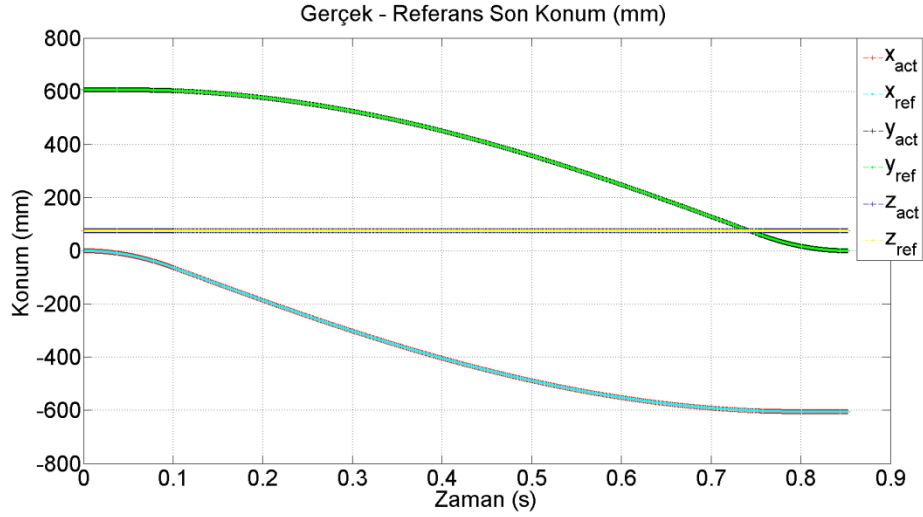
Şekil 5.49 : Gerçek - referans eklem açısal hızları (2. hareket, trapez, 3 kg yükü).



Şekil 5.50: Yaklaşımlı hesaplanmış motor tork değerleri(2.hareket, trapez, 3 kg yükü).



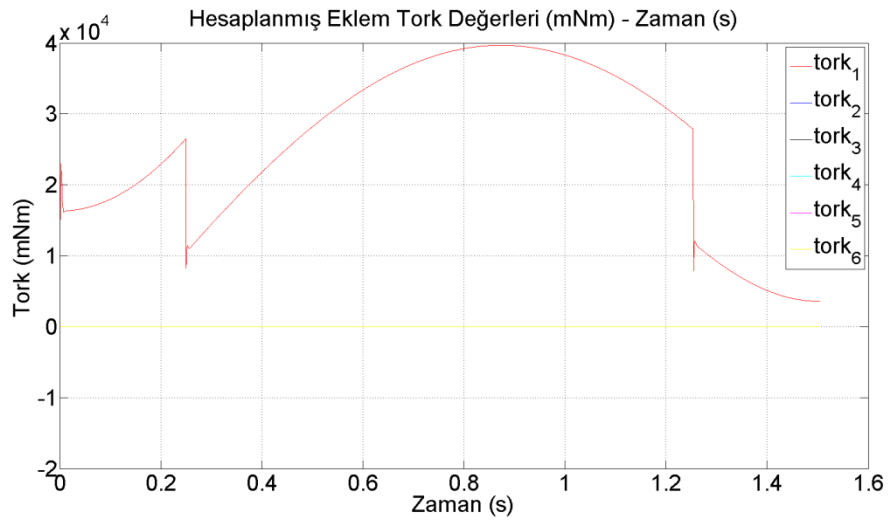
Şekil 5.51 : Motor akım değerleri (2. hareket, trapez, 3 kg yükü).



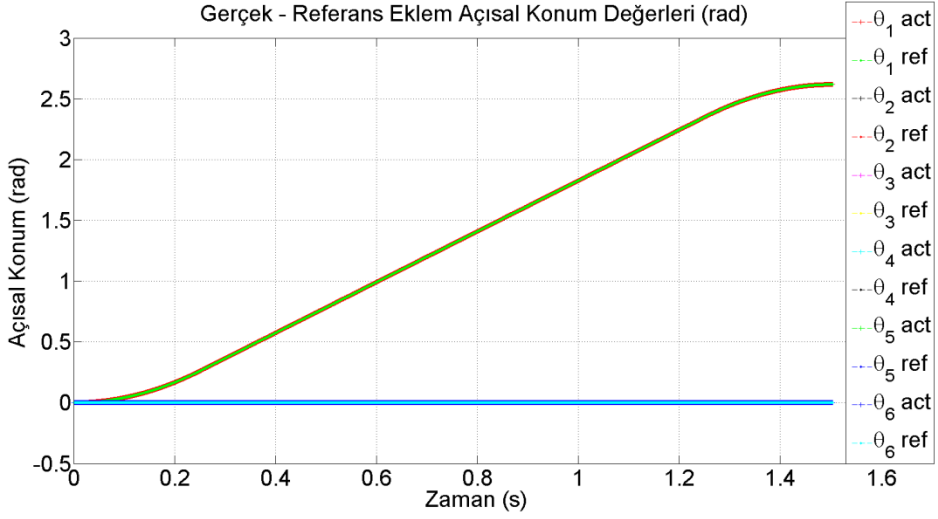
Şekil 5.52 : Gerçek-referans uç konum koordinatları (2. hareket, trapez, 3 kg yük).

### 5.11 Ön Tasarım ile Karşılaştırma Benzetimi

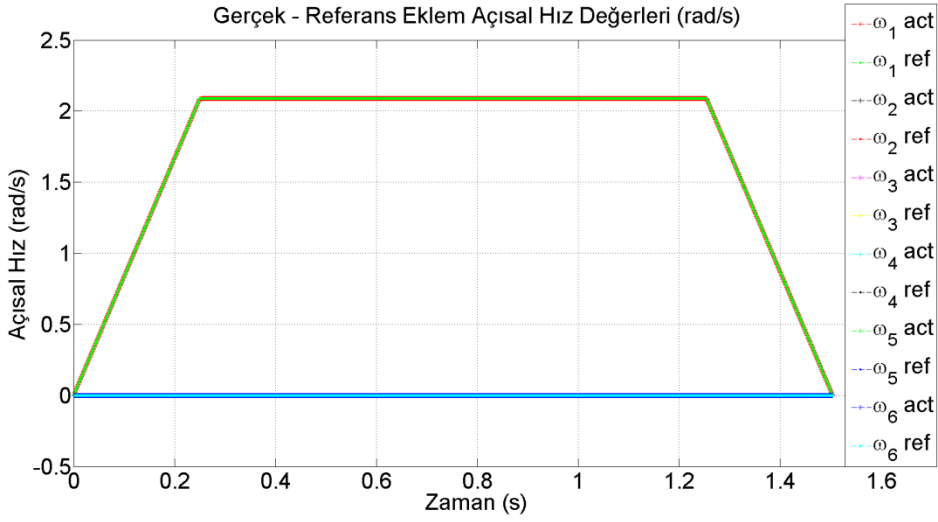
Son olarak İlkay Meşeli tarafından ön tasarımı yapılmış olan robot kolunun dinamik benzetim sonuçları ile karşılaştırma yapmak için aynı koşullarda son bir benzetim daha yapılmıştır. Bu benzetimde ilk eklemler, ilk konumundan  $150^0$  'ye kadar 1.5 saniyede kol ucunda 5 kg ilave ağırlık ile dönmesi istenmiştir. Şekil 5.53 ile Şekil 5.58 arası grafikler bu aynı şartlarda yeni tasarım üzerinde yapılmış benzetimin sonuçlarıdır.



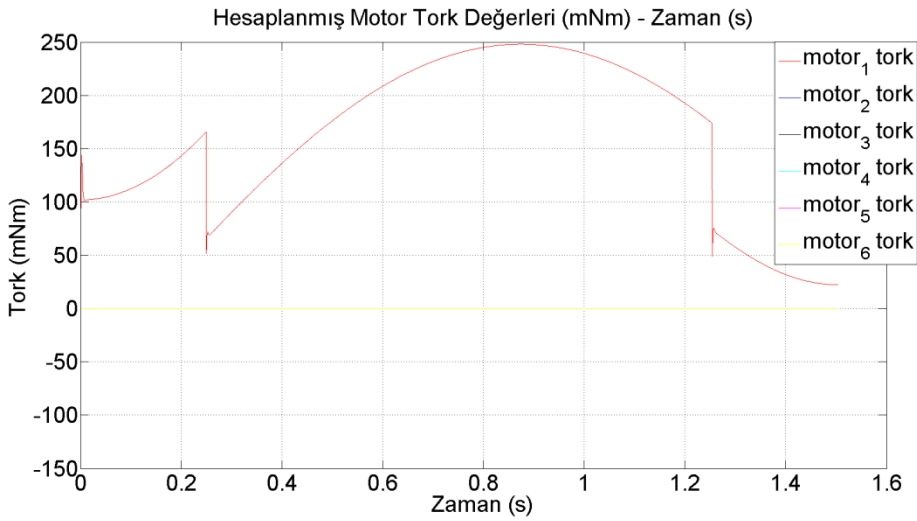
Şekil 5.53 : Yaklaşımlı hesaplanmış eklemler tork değerleri.



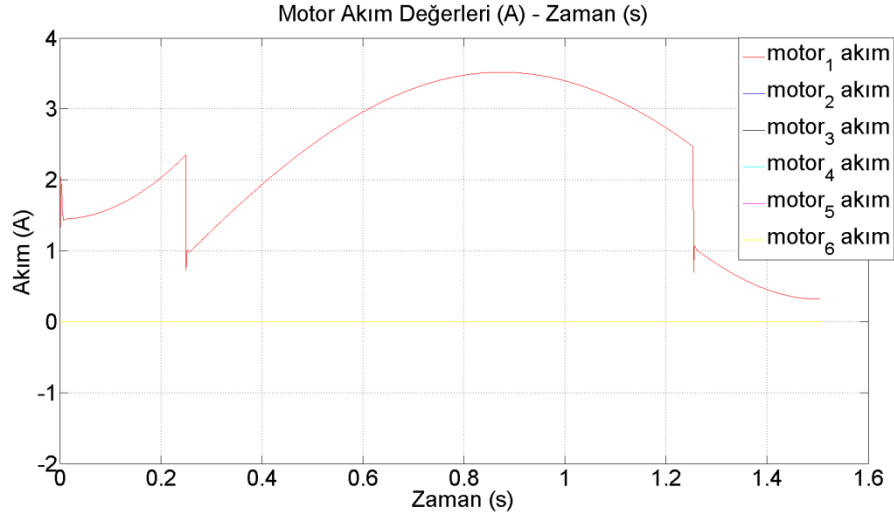
Şekil 5.54 : Gerçek-referans eklem açısal konumları.



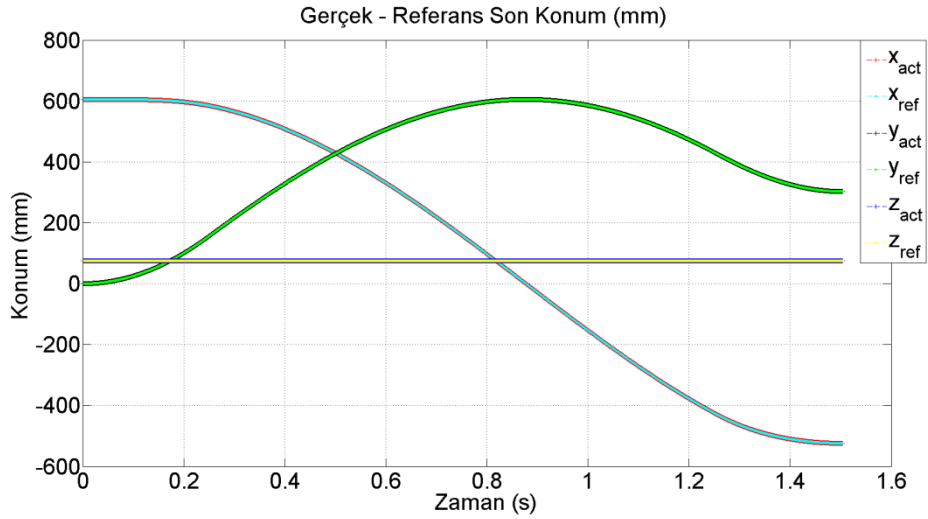
Şekil 5.55 : Gerçek - referans eklem açısal hızları.



Şekil 5.56 : Yaklaşımli hesaplanmış motor tork değerleri.



Şekil 5.57 : Motor akım değerleri.



Şekil 5.58 : Gerçek-referans uç konum koordinatları.

Ön tasarımdaki sonuçlara göre 5 kg yük ile 1,5 saniyede bu hareketi yapmak motor ve dişli mekanizmasının sınır tork değerlerine ulaşmasına sebep olmuştur. Bu tez çalışmasında tasarlanan robot kolunun aynı yük ve aynı sürede kübik hız profili kullanılarak benzetimi yapılmıştır. Şekil 5.53 'de görüldüğü gibi ilk eklemden bulunan EC 90 tipi motorda oluşan maksimum tork değeri 248 mNm kadardır. EC 90 motorunun uygulayabileceği maksimum tork değeri ise 444 mNm olduğuna göre motor açısından sınır değerine henüz ulaşılmamıştır. Şekil 5.56 incelendiğinde ise eklemden oluşan maksimum tork değerinin 39,7 Nm olduğu görülmektedir. İlk eklemden kullanılan CPL 20A tipi Harmonic Drive® dişli mekanizmasının

dayanabildiđi maksimum tork deęeri ise 92 Nm kadardır. Bu deęerler ise diřli mekanizması aısından da bir problem olmadıđını, hatta diřli mekanizmasının limit deęeri ile gereken tork deęeri arasında ciddi bir fark olduđunu gstermektedir. Sonu olarak, tasarımı yapılan insansı robot kolunun daha nce yapılmıř olan n tasarıma kıyasla daha gl olduđu, daha yksek ykler kaldırabileceęi, daha yksek tork deęerlerini saęlayabileceęi veya hareketi daha hızlı gerekleřtirebileceęi aık bir řekilde grlmektedir.

## 6. SONUÇ VE ÖNERİLER

Bu çalışmada, İTÜ Makina Fakültesi, Sistem Dinamiği ve Kontrol Bölümü tarafından yürütülen insansı robot çalışmasının bir parçası olan insansı robot kolunun tasarımı ve yörünge kontrolü yapılmıştır. Sırasıyla insansı robot kolu mekanik tasarımı, ileri kinematik ve ters kinematik çözümleri, yörünge planlanması ve yaklaşımlı hesaplanmış tork ile PD kontrolü üzerinde çalışılmıştır.

Bu çalışmadaki asıl amaç, kol tasarımını yapmak ve tasarım sonucunda elde edilen insansı robot kolunun, dinamik ve kinematik çözümlerini elde etmektir. 5. bölümde anlatıldığı gibi, hem motorların, hem de Harmonic Drive® dişli mekanizmalarının sınır tork değerlerine seçilen yörüngelerin hiçbirinde ulaşılmamıştır. Bu yapılan çalışmanın ve seçilen motor, dişli mekanizması gruplarının uygun olduğunu göstermektedir. Şekil 5.50 'de görüldüğü gibi en yüksek motor tork değeri son benzetimde görülmektedir ve değeri ortalama 387,8 mNm 'dir. Çizelge 2.3'de görüldüğü gibi, ilk eklem için seçtiğimiz EC90 motorunun uygulayabileceği maksimum tork değeri ise 444 mNm 'dir. Bu ise seçilen motorların daha büyük yük ile, istenilen ivme değerinde de çalışabileceğini göstermektedir. Eklem torqları incelendiği zaman ise aynı benzetimde, Şekil 5.47 'de görüldüğü gibi eklemde oluşan maksimum tork değeri 62,1 Nm 'dir. Çizelge 2.3 'deki EC90 motoruna monte edilen Harmonic Drive® dişli mekanizmasının dayanabildiği maksimum tork ise 92 Nm'dir. Dolayısıyla dişli mekanizması açısından da bir problem görülmemektedir.

Bütün bu kıyaslamalar sonucunda denilebilir ki, tasarımı yapılan insansı robot kolu benzetimlerde yapılan denemelerden daha kötü şartlar ve tork değerleri altında da çalışabilir. Bir başka deyişle, ilave daha fazla yük ile veya daha yüksek tork değerleri ile çalışabilir. Robot kolu üzerinde devam edilen çalışmalarda bir el tasarımı yapılması gerekmektedir. Bundan sonra yapılması gereken çalışmalar ise mekanik tasarımın detaylandırılması, sürücü devrelerin tasarlanması, kontrol algoritmalarının geliştirilmesi, haberleşme protokollerinin ve gerçek zamanda kontrolün uygulanmasıdır.



## KAYNAKÇA

- [1]<<http://wonderfulengineering.com/history-of-humanoid-robots-in-pictures-1868-1970/>>, alındığı tarih: 24.08.2013.
- [2]**ASIMO**. *Wikipedia*, alındığı tarih: 16.07.2013 ,  
adres:<http://tr.wikipedia.org/wiki/ASIMO>.
- [3]<[http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-5471/8991\\_read-39464/](http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-5471/8991_read-39464/)> ,  
alındığı tarih : Mayıs 2009.
- [4]<<http://www.theroboticschallenge.org/node/58>>, alındığı tarih : 2013.
- [5]**Atlas (robot)**, *Wikipedia*, alındığı tarih 04.04.2014, adres :  
[http://en.wikipedia.org/wiki/Atlas\\_\(robot\)](http://en.wikipedia.org/wiki/Atlas_(robot)).
- [6]<<http://www.army-technology.com/projects/petman/>>, alındığı tarih :  
17.05.2014.
- [7]<<http://spectrum.ieee.org/automaton/robotics/humanoids/france-developing-advanced-humanoid-robot-romeo>>, alındığı tarih : 13.12.2010.
- [8]<<http://global.kawada.jp/mechatronics/hrp4.html>>, alındığı tarih : 17.05.2014.
- [9]<<http://www.roboticturnkeysolutions.com/robots/motoman/datasheet/DA20.pdf>> ,  
alındığı tarih : 17.05.2014.
- [10]<<http://www.abb.com/cawp/abbzh254/8657f5e05ede6ac5c1257861002c8ed2.aspx>>, alındığı tarih : 17.05.2014.
- [11]<<http://www.maxonmotor.com/maxon/view/catalog>>, alındığı tarih: 03.03.2014.
- [12]<<http://harmonicdrive.de/en/download/cataloguebrochure/>>, alındığı tarih:  
28.03.2014.
- [13]**Mark W. Spong, Seth Hutchinson, and M.Vidyasagar**, Robot Modeling and Control, First Edition, FOHN WILEY & SONS, INC.
- [14]**T.asfour and R. Dillmann**, Human-like Motion of a Humanoid Robot Arm Based on a Closed-Form Solution of the Inverse Kinematics Problem, Industrial Applications of Informatics and Microsystems (IAIM) Computer Science Department, University of Karlsruhe, Germany, Oct 27-21, 2003.
- [15]**W. Khalil and E. Dombre**, Modeling, Identification and Control of Robots.
- [16]**F.L. Lewis, C.T. Abdallah, D.M. Dawson**, Control of Robot Manipulators.



## **EKLER**

**EK A :** Kullanılan motorların katalogları

**EK B :** Kullanılan HarmonicDrive<sup>®</sup> diřli mekanizmalarının katalogu

**EK C :** İleri kinematik çözümlü MATLAB<sup>®</sup> kodu.

**EK D :** Ters kinematik çözümlü MATLAB<sup>®</sup> kodu.

**EK E :** Trapez hız profili ile yörünge planlanmasının MATLAB<sup>®</sup> kodu.

**EK F :** Kübik hız profili ile yörünge planlanmasının MATLAB<sup>®</sup> kodu.

**EK G :** Jacobian matrisi.

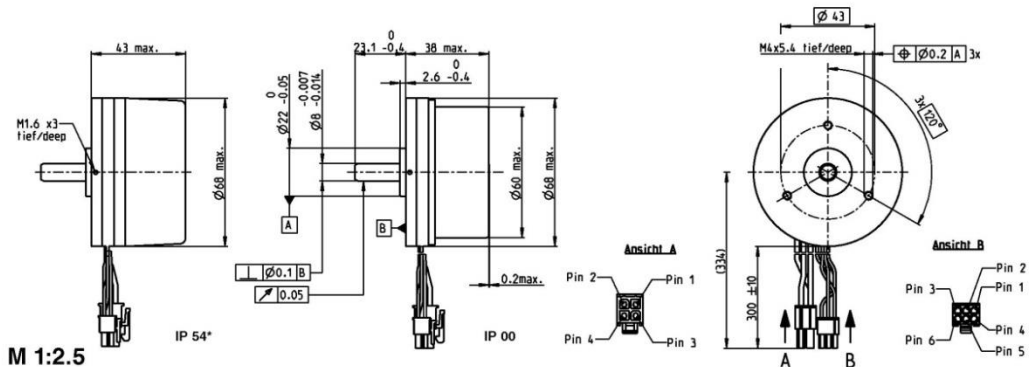
**EK H :**  $D$  matrisinin belirlenmesinin MATLAB<sup>®</sup> kodu.

**EK I :**  $g$  matrisinin belirlenmesinin MATLAB<sup>®</sup> kodu.

**EC 60 flat** Ø60 mm, brushless, 100 Watt

**NEW**

maxon flat motor



M 1:2.5

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

**Article numbers**

IP 54* (with cover)	412819	408057	412821	412825
IP 00 (without cover)	412823	411678		

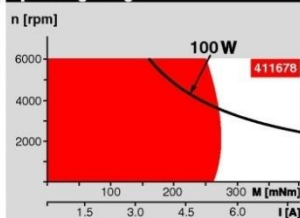
**Motor Data** (provisional)

Values at nominal voltage		12	12	24	24	48	48
1 Nominal voltage	V	12	12	24	24	48	48
2 No load speed	rpm	3710	3710	4250	4250	3970	3970
3 No load current	mA	671	671	419	419	187	187
4 Nominal speed	rpm	3290	3190	3850	3740	3580	3500
5 Nominal torque (max. continuous torque)	mNm	216	267	221	284	255	317
6 Nominal current (max. continuous current)	A	7.37	8.88	4.33	5.39	2.28	2.77
7 Stall torque	mNm	2820	2820	3740	3740	4190	4190
8 Starting current	A	93.5	93.5	78.2	78.2	43.8	43.8
9 Max. efficiency	%	84	84	86	86	88	88
<b>Characteristics</b>							
10 Terminal resistance phase to phase	Ω	0.128	0.128	0.307	0.307	1.1	1.1
11 Terminal inductance phase to phase	mH	0.0615	0.0615	0.188	0.188	0.864	0.864
12 Torque constant	mNm/A	30.5	30.5	53.4	53.4	114	114
13 Speed constant	rpm/V	313	313	179	179	83.4	83.4
14 Speed/torque gradient	rpm/mNm	1.32	1.32	1.03	1.03	0.798	0.798
15 Mechanical time constant	ms	16.7	16.7	13	13	10.1	10.1
16 Rotor inertia	gcm <sup>2</sup>	1210	1210	1210	1210	1210	1210

**Specifications**

- Thermal data**
- 17 Thermal resistance housing-ambient 4.31 (2.41) K/W
  - 18 Thermal resistance winding-housing 3.8 K/W
  - 19 Thermal time constant winding 40 s
  - 20 Thermal time constant motor 155 (86.9) s
  - 21 Ambient temperature -40...+100°C
  - 22 Max. permissible winding temperature +125°C
- Mechanical data (preloaded ball bearings)**
- 23 Max. permissible speed 6000 rpm
  - 24 Axial play at axial load < 15.0 N 0 mm
  - > 15.0 N 0.14 mm preloaded
  - 25 Radial play 12 N
  - 26 Max. axial load (dynamic) 170 N
  - 27 Max. force for press fits (static) (static, shaft supported) 6000 N
  - 28 Max. radial loading, 7.5 mm from flange 100 N

**Operating Range**



- Comments**
- Continuous operation**  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.
  - Short term operation**  
The motor may be briefly overloaded (recurring).
  - Assigned power rating**

**Other specifications**

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

Values listed in the table are nominal.

**Connection motor** (Cable AWG 16)

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

**Connector Article number**

Molex 39-01-2040

**Connection Sensors** (Cable AWG 26)

- grey Hall sensor 1 Pin 1
- grey Hall sensor 2 Pin 2
- grey Hall sensor 3 Pin 3
- grey GND Pin 4
- blue V<sub>bat</sub> 4.5...18 VDC Pin 5
- N.C. N.C. Pin 6

**Connector Article number**

Molex 430-25-0600

Wiring diagram for Hall sensors see p. 29

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

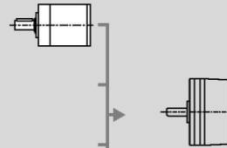
198 maxon EC motor

**maxon Modular System**

Overview on page 16 - 21

**Planetary Gearhead**

Ø52 mm  
4 - 30 Nm  
Page 246



**Recommended Electronics:**

- ESCON 50/5 Page 292
- DEC Module 50/5 299
- DEC 70/10 305
- EPOS2 24/5 313
- EPOS2 50/5 313
- EPOS2 70/10 313
- EPOS2 P 24/5 316
- EPOS3 70/10 319
- Notes 20

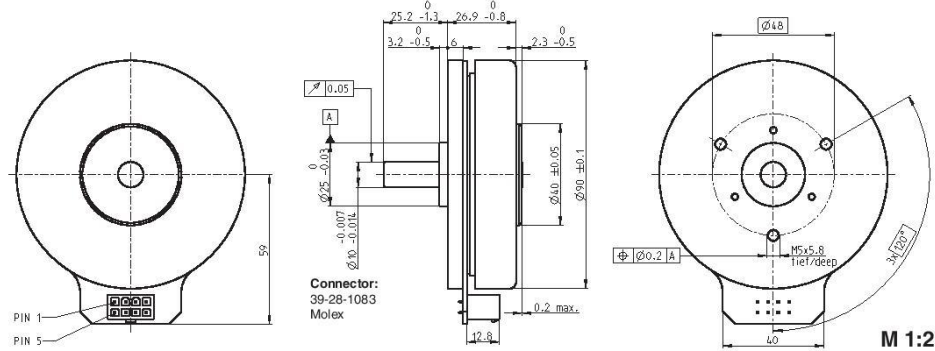
**Encoder MILE**  
512 - 2048 CPT,  
2 channels  
Page 263

May 2012 edition / subject to change

Şekil A.1 : EC60 motor kataloğu.

# EC 90 flat Ø90 mm, brushless, 90 Watt

maxon flat motor



**Connector:**  
39-28-1083  
Molex

**Part Numbers**

	323772	429271	244879
with Hall sensors			

**Motor Data**

Values at nominal voltage		24	36	48
1 Nominal voltage	V	24	36	48
2 No load speed	rpm	3190	3120	2080
3 No load current	mA	544	348	135
4 Nominal speed	rpm	2590	2510	1610
5 Nominal torque (max. continuous torque)	mNm	444	560	533
6 Nominal current (max. continuous current)	A	6.06	4.76	2.27
7 Stall torque	mNm	4690	5730	4460
8 Starting current	A	70	69	21.1
9 Max. efficiency	%	84	87	85
<b>Characteristics</b>				
10 Terminal resistance phase to phase	Ω	0.343	0.522	2.28
11 Terminal inductance phase to phase	mH	0.264	0.625	2.5
12 Torque constant	mNm/A	70.5	109	217
13 Speed constant	rpm/V	135	88	44
14 Speed/torque gradient	rpm/mNm	0.659	0.423	0.462
15 Mechanical time constant	ms	21.1	13.6	14.8
16 Rotor inertia	gcm <sup>2</sup>	3060	3060	3060

**Specifications**

**Thermal data**

17 Thermal resistance housing-ambient	1.91 K/W
18 Thermal resistance winding-housing	2.6 K/W
19 Thermal time constant winding	46.6 s
20 Thermal time constant motor	283 s
21 Ambient temperature	-40...+100°C
22 Max. permissible winding temperature	+125°C

**Mechanical data (preloaded ball bearings)**

23 Max. permissible speed	5000 rpm
24 Axial play at axial load < 15 N	0 mm
> 15 N	0.14 mm
25 Radial play	preloaded
26 Max. axial load (dynamic)	12 N
27 Max. force for press fits (static)	183 N
(static, shaft supported)	8000 N
28 Max. radial loading, 7.5 mm from flange	60 N

**Other specifications**

29 Number of pole pairs	12
30 Number of phases	3
31 Weight of motor	600 g

Values listed in the table are nominal.

**Connection**

Pin 1	Hall sensor 1
Pin 2	Hall sensor 2
Pin 3	V <sub>Hall</sub> 4.5...18 VDC
Pin 4	Motor winding 3
Pin 5	Hall sensor 3
Pin 6	GND
Pin 7	Motor winding 1
Pin 8	Motor winding 2

Wiring diagram for Hall sensors see p. 31

**Cable**

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

**Operating Range**

**Comments**

- Continuous operation**  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.
- Short term operation**  
The motor may be briefly overloaded (recurring).
- Assigned power rating**

**maxon Modular System**

Overview on page 20 - 25

12 Planetary Gearhead	Encoder MILE
3 Ø52 mm	800 - 6400 CPT,
4 - 30 Nm	2 channels
Page 274	Page 294

**Recommended Electronics:**

ESCON 50/5	Page 321
ESCON Module 50/5	321
ESCON 70/10	321
DECS 50/5	324
DEC Module 50/5	325
EPOS2 24/5, 50/5, 70/10	331
EPOS2 P 24/5	334
EPOS3 70/10 EtherCAT	337
Notes	24

June 2013 edition / subject to change

maxon EC motor 225

Şekil A.2 : EC90 motor katalogu.

## EK B

Tabelle / Table 23.2

CPL-2A Baugröße	Unter- setzung	Grenze für wieder- holbares Spitzen- drehmoment	Grenze für Durchschnitts- drehmoment	Nennrehmo- ment bei 2000 min <sup>-1</sup>	Grenze für Kollisions- drehmoment	Max. Antriebs- drehzahl [min <sup>-1</sup> ]		Grenze für mittlere Antriebsdrehzahl [min <sup>-1</sup> ]		Massen- trägheits- moment <sup>2)</sup>	Gewicht
						Max. input speed [rpm]	Fett <sup>1)</sup> Grease Lub. <sup>1)</sup>	Öl Oil Lub.	Fett <sup>1)</sup> Grease Lub. <sup>1)</sup>		
Size	Ratio	Limit for repeated peak torque	Limit for repeated peak torque	Rated torque at 2000 rpm	Limit for momentary peak torque	Öl Oil Lub.	Fett <sup>1)</sup> Grease Lub. <sup>1)</sup>	Öl Oil Lub.	Fett <sup>1)</sup> Grease Lub. <sup>1)</sup>	[kgm <sup>2</sup> ]	[kg]
	i	T <sub>R</sub> [Nm]	T <sub>A</sub> [Nm]	T <sub>N</sub> [Nm]	T <sub>M</sub> [Nm]						
14	50	18	6,9	5,4	35	14000	8500	6500	3500	0,02 x 10 <sup>4</sup>	0,055
	80	23	11	7,8	47						
	100	28	11	7,8	54						
17	50	34	26	16	70	10000	7300	6500	3500	0,049 x 10 <sup>4</sup>	0,1
	80	43	27	22	87						
	100	54	39	24	110						
	120	54	39	24	86						
20	50	56	34	25	98	10000	6500	6500	3500	0,112 x 10 <sup>4</sup>	0,14
	80	74	47	34	127						
	100	82	49	40	147						
	120	87	49	40	147						
	160	92	49	40	147						
25	50	98	55	39	186	7500	5600	5600	3500	0,263 x 10 <sup>4</sup>	0,24
	80	137	87	63	255						
	100	157	108	67	284						
	120	167	108	67	304						
32	50	216	108	76	382	7000	4800	4600	3500	0,928 x 10 <sup>4</sup>	0,54
	80	304	167	118	568						
	100	333	216	137	647						
	120	353	216	137	686						
	160	372	216	137	686						

Şekil B.1 : Kullanılan Harmonic Drive® dişli mekanizmalarının katalogu.

## EK C

```
clear all
clc

syms teta1 teta2 teta3 teta4 teta5 teta6 a6 d1 d3 d5

syms r11 r12 r13 r14 r21 r22 r23 r24 r31 r32 r33 r34 X Y
Z

T = [r11 r12 r13 X;
     r21 r22 r23 Y;
     r31 r32 r33 Z;
     0 0 0 1];

A1 = [ cos(teta1), 0, -sin(teta1), 0;
       sin(teta1), 0,  cos(teta1), 0;
       0, -1,          0, d1;
       0, 0,          0, 1];

A2 = [ -sin(teta2), 0,  cos(teta2), 0;
       cos(teta2), 0,  sin(teta2), 0;
       0, 1,          0, 0;
       0, 0,          0, 1];

A3 = [ cos(teta3), 0, -sin(teta3), 0;
       sin(teta3), 0,  cos(teta3), 0;
       0, -1,          0, d3;
       0, 0,          0, 1];

A4 = [ cos(teta4), 0,  sin(teta4), 0;
       sin(teta4), 0, -cos(teta4), 0;
       0, 1,          0, 0;
       0, 0,          0, 1];

A5 = [ cos(teta5), 0, -sin(teta5), 0;
       sin(teta5), 0,  cos(teta5), 0;
       0, -1,          0, d5;
       0, 0,          0, 1];

A6 = [ sin(teta6),  cos(teta6), 0, a6*sin(teta6);
       -cos(teta6), sin(teta6), 0, -a6*cos(teta6);
       0,          0, 1,          0;
       0,          0, 0,          1];

simplify(A1*A2*A3*A4*A5*A6)
```

## EKD

```
function [teta1, teta2, teta3, teta4, teta5, teta6] =
invkininfsol_x2(X, Y, Z, euler_p, euler_t, euler_z, dx,
err_o)
a1 = 0; a2 = 0; a3 = 0; a4 = 0; a5 = 0; a6 = 166.31;
d1 = 75.5; d2 = 0; d3 = 225; d4 = 0; d5 = 214; d6 = 0;
err = 0.003;
teta_mind = [-180,-15,-180,-115,-180,-115];
teta_maxd = [180,195,180,115,180,115];
teta_minr = teta_mind*pi/180;
teta_maxr = teta_maxd*pi/180;
R = [cos(euler_p)*cos(euler_t)*cos(euler_z) -
      sin(euler_p)*sin(euler_z), -
      cos(euler_p)*cos(euler_t)*sin(euler_z) -
      sin(euler_p)*cos(euler_z),
      cos(euler_p)*sin(euler_t);...
      sin(euler_p)*cos(euler_t)*cos(euler_z)+cos(euler_p)*
      sin(euler_z), -
      sin(euler_p)*cos(euler_t)*sin(euler_z)+cos(euler_p)*
      cos(euler_z), sin(euler_p)*sin(euler_t);...
      -sin(euler_t)*cos(euler_z)
      , sin(euler_t)*sin(euler_z)
      , cos(euler_t)];
for h=1:1:3
  for f=1:1:3
    if ((abs(R(h,f))<=0.0000001) | (abs((abs(R(h,f))-
      1))<=0.0000001))
      R(h,f) = round(R(h,f));
    end
  end
end
end
Xw = X - a6*R(1,1)
Yw = Y - a6*R(2,1)
Zw = Z - a6*R(3,1)
W = [Xw;Yw;Zw];
i=1;
for x=-d3:dx:d3
  r1f = sqrt(d3^2-x^2);
  r2f = sqrt(d5^2-(Xw-x)^2);
  % r1^2 = Ye^2 + (Ze-d1)^2
  %r2^2 = (Zw-Ze)^2+(Yw-Ye)^2 then solve for Xe and Ye
  Af = (Yw^2 + r1f^2 + Zw^2 - r2f^2-d1^2)/(2*Yw);
  Bf = (Zw-d1)/Yw;
  Ze_1f = (2*(Af*Bf+d1) + sqrt(4*Af^2*Bf^2 + 4*d1^2 +
    8*Af*Bf*d1- 4*(Bf^2+1)*(Af^2+d1^2-r1f^2)))/
    (2*(Bf^2+1));
  Ze_2f = (2*(Af*Bf+d1) - sqrt(4*Af^2*Bf^2 + 4*d1^2 +
    8*Af*Bf*d1- 4*(Bf^2+1)*(Af^2+d1^2-r1f^2)))/
    (2*(Bf^2+1));
```

```

Ye_1f = Af-Bf*Ze_1f;
Ye_2f = Af-Bf*Ze_2f;
if ((Yw==0) & (Zw==0))
    Ze_1f = (r2f^2-r1f^2+d1^2)/(2*d1);
    Ze_2f = (r2f^2-r1f^2+d1^2)/(2*d1);
    Ye_1f = sqrt(r2f^2-Ze_1f^2);
    Ye_2f = -sqrt(r2f^2-Ze_2f^2);
end
if ((Yw==0) & (Zw~=0))
    Ze_1f = (r2f^2-r1f^2-Zw^2+d1^2)/(2*d1-2*Zw);
    Ze_2f = (r2f^2-r1f^2-Zw^2+d1^2)/(2*d1-2*Zw);
    Ye_1f = sqrt(r2f^2-Ze_1f^2-Zw^2+2*Zw*Ze_1f);
    Ye_2f = -sqrt(r2f^2-Ze_2f^2-Zw^2+2*Zw*Ze_2f);
    if ((Zw==d1) & (abs((abs(r1f)-abs(r2f))))<err))
        Ze_1f = r1f+d1;
        Ze_2f = -r1f+d1;
        Ye_1f = 0;
        Ye_2f = 0;
    end
    if ((Zw==d1) & (abs((abs(r1f)-abs(r2f))))>err))
        r1f = -1;
        r2f = -1;
    end
end
if (imag(r1f)>0) | (imag(r2f)>0) | (imag(Ye_1f)>0) |
(imag(Ze_1f)>0)
    r1f=-1;
    r2f=-1;
end
if (x==d3)
    C = 1;
    C2 = 1;
end
if (x~=d3)
    C = norm(abs(Y4_1f-Y4d));
    C2 = norm(abs(Y4_2f-Y4d));
end
e1 = [x;Ye_1f;Ze_1f];
e2 = [x;Ye_2f;Ze_2f];
Z4_1 = (W - e1)/norm(W - e1);
Z4_2 = (W - e2)/norm(W - e2);
Z4skew_1 = [0 -Z4_1(3) Z4_1(2);Z4_1(3) 0 -Z4_1(1);-
            Z4_1(2) Z4_1(1) 0];
Z4skew_2 = [0 -Z4_2(3) Z4_2(2);Z4_2(3) 0 -Z4_2(1);-
            Z4_2(2) Z4_2(1) 0];
rs = [0;0;d1];
ro_1 = (rs - e1)/norm(rs-e1);
ro_2 = (rs - e2)/norm(rs-e2);
Y4_1f = (-Z4skew_1*ro_1)/norm(-Z4skew_1*ro_1);
Y4_2f = (-Z4skew_2*ro_2)/norm(-Z4skew_2*ro_2);
if(Z4skew_1*ro_1==0)

```

```

    Y4_1f =
    [R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
end
if(Z4skew_2*ro_2==0)
    Y4_2f =
    [R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
end
Y4skew_1 = [0 -Y4_1f(3) Y4_1f(2);Y4_1f(3) 0 -
            Y4_1f(1);-Y4_1f(2) Y4_1f(1) 0];
Y4skew_2 = [0 -Y4_2f(3) Y4_2f(2);Y4_2f(3) 0 -
            Y4_2f(1);-Y4_2f(2) Y4_2f(1) 0];
X4_1 = (Y4skew_1*Z4_1)/norm(Y4skew_1*Z4_1);
X4_2 = (Y4skew_2*Z4_2)/norm(Y4skew_2*Z4_2);
T4d_1 = [X4_1, Y4_1f, Z4_1, e1;0 0 0 1];
T4d_2 = [X4_2, Y4_2f, Z4_2, e2;0 0 0 1];
Y4d =
    [R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
B = norm(abs(Y4_1f-Y4d));
B2 = norm(abs(Y4_2f-Y4d));
Y3_1 = (e1-W)/norm(e1-W);
Y3_2 = (e2-W)/norm(e2-W);
Y3_1skew = [0 -Y3_1(3) Y3_1(2);Y3_1(3) 0 -Y3_1(1);-
            Y3_1(2) Y3_1(1) 0];
Y3_2skew = [0 -Y3_2(3) Y3_2(2);Y3_2(3) 0 -Y3_2(1);-
            Y3_2(2) Y3_2(1) 0];
ro_1skew = [0 -ro_1(3) ro_1(2);ro_1(3) 0 -ro_1(1);-
            ro_1(2) ro_1(1) 0];
ro_2skew = [0 -ro_2(3) ro_2(2);ro_2(3) 0 -ro_2(1);-
            ro_2(2) ro_2(1) 0];
    if(e1(3)<=d1)
        Z3_1 = (ro_1skew*Y3_1)/norm(ro_1skew*Y3_1);
    end
    if(e2(3)<=d1)
        Z3_2 = (-ro_2skew*Y3_2)/norm(ro_2skew*Y3_2);
    end
    if(e1(3)>d1)
        Z3_1 = (-ro_1skew*Y3_1)/norm(ro_1skew*Y3_1);
    end
    if(e2(3)>d1)
        Z3_2 = (-ro_2skew*Y3_2)/norm(ro_2skew*Y3_2);
    end
    if(ro_1skew*Y3_1==0)
        Z3_1 = [-T4d_1(2,4);T4d_1(1,4);0]/norm([-
            T4d_1(2,4);T4d_1(1,4);0]);
    end
    if(ro_1skew*Y3_1==0)
        Z3_2 = [-T4d_2(2,4);T4d_2(1,4);0]/norm([-
            T4d_2(2,4);T4d_2(1,4);0]);
    end
X3_1 = (Y3_1skew*Z3_1)/norm(Y3_1skew*Z3_1);
X3_2 = (Y3_2skew*Z3_2)/norm(Y3_2skew*Z3_2);

```

```

T3d_1 = [X3_1,Y3_1,Z3_1,e1;0 0 0 1];
T3d_2 = [X3_2,Y3_2,Z3_2,e2;0 0 0 1];

Ws = [X;Y;Z];
Y5_1 = (W-Ws)/norm(W-Ws);
Y5_2 = (W-Ws)/norm(W-Ws);
rew_1 = (e1-W)/norm(e1-W);
rew_2 = (e2-W)/norm(e2-W);
rew_1skew = [0 -rew_1(3) rew_1(2);rew_1(3) 0 -
             rew_1(1);-rew_1(2) rew_1(1) 0];
rew_2skew = [0 -rew_2(3) rew_2(2);rew_2(3) 0 -
             rew_2(1);-rew_2(2) rew_2(1) 0];

Z5d =
[R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
Z5d =
[R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
Y5_1skew = [0 -Y5_1(3) Y5_1(2);Y5_1(3) 0 -Y5_1(1);-
Y5_1(2) Y5_1(1) 0];
Y5_2skew = [0 -Y5_2(3) Y5_2(2);Y5_2(3) 0 -Y5_2(1);-
Y5_2(2) Y5_2(1) 0];
Z5_1f = (-Y5_1skew*rew_1)/norm(Y5_1skew*rew_1);
Z5_2f = (-Y5_2skew*rew_2)/norm(Y5_2skew*rew_2);
if(Y5_1skew*rew_1==0)
    Z5_1f =
    [R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
end
if(Y5_2skew*rew_2==0)
    Z5_2f =
    [R(1,3);R(2,3);R(3,3)]/norm([R(1,3);R(2,3);R(3,3)]);
end
X5_1 = Y5_1skew*Z5_1f/norm(Y5_1skew*Z5_1f);
X5_2 = Y5_2skew*Z5_2f/norm(Y5_2skew*Z5_2f);
T5d_1 = [X5_1 , Y5_1, Z5_1f , W;0 0 0 1];
T5d_2 = [X5_2 , Y5_2, Z5_2f , W;0 0 0 1];
T = [R,Ws;0,0,0,1];
if ((r1f>=0)&(abs(Y4_1f(1)-
Y4d(1))<err_o)&(abs(Y4_1f(2)-
Y4d(2))<err_o)&(abs(Y4_1f(3)-
Y4d(3))<err_o)&(abs(Z5_1f(2)-
Z5d(2))<err_o)&(abs(Z5_1f(2)-
Z5d(2))<err_o)&(abs(Z5_1f(3)-Z5d(3))<err_o))
    teta1_1 = atan2(T4d_1(2,4),T4d_1(1,4));
    teta2_1 =
atan2(sqrt(T4d_1(1,4)^2+T4d_1(2,4)^2),(T4d_1(3,4)-d1)) -
pi/2;
    teta3_1 = atan2((T3d_1(2,1)*cos(teta1_1) -
T3d_1(1,1)*sin(teta1_1)),(- T3d_1(3,1)*cos(teta2_1) -
T3d_1(1,1)*cos(teta1_1)*sin(teta2_1) -
T3d_1(2,1)*sin(teta1_1)*sin(teta2_1)));
    teta4_1 = atan2((T4d_1(3,1)*sin(teta2_1) -
T4d_1(1,1)*cos(teta1_1)*cos(teta2_1) -

```

```

T4d_1(2,1)*cos(teta2_1)*sin(teta1_1)), (T4d_1(2,1)*cos(tet
a1_1)*sin(teta3_1)-T4d_1(3,1)*cos(teta2_1)*cos(teta3_1)-
T4d_1(1,1)*sin(teta1_1)*sin(teta3_1)-
T4d_1(1,1)*cos(teta1_1)*cos(teta3_1)*sin(teta2_1)-
T4d_1(2,1)*cos(teta3_1)*sin(teta1_1)*sin(teta2_1)));
    teta5_1 =
atan2((T5d_1(2,1)*cos(teta1_1)*cos(teta3_1) -
T5d_1(1,1)*cos(teta3_1)*sin(teta1_1) +
T5d_1(3,1)*cos(teta2_1)*sin(teta3_1) +
T5d_1(1,1)*cos(teta1_1)*sin(teta2_1)*sin(teta3_1) +
T5d_1(2,1)*sin(teta1_1)*sin(teta2_1)*sin(teta3_1)), (T5d_1
(3,1)*sin(teta2_1)*sin(teta4_1) -
T5d_1(3,1)*cos(teta2_1)*cos(teta3_1)*cos(teta4_1) -
T5d_1(1,1)*cos(teta1_1)*cos(teta2_1)*sin(teta4_1) +
T5d_1(2,1)*cos(teta1_1)*cos(teta4_1)*sin(teta3_1) -
T5d_1(1,1)*cos(teta4_1)*sin(teta1_1)*sin(teta3_1) -
T5d_1(2,1)*cos(teta2_1)*sin(teta1_1)*sin(teta4_1) -
T5d_1(1,1)*cos(teta1_1)*cos(teta3_1)*cos(teta4_1)*sin(tet
a2_1) -
T5d_1(2,1)*cos(teta3_1)*cos(teta4_1)*sin(teta1_1)*sin(tet
a2_1)));
    teta6_1 =
atan2((R(2,1)*cos(teta1_1)*cos(teta3_1)*sin(teta5_1) -
R(1,1)*cos(teta3_1)*sin(teta1_1)*sin(teta5_1) +
R(3,1)*cos(teta2_1)*sin(teta3_1)*sin(teta5_1) +
R(3,1)*cos(teta5_1)*sin(teta2_1)*sin(teta4_1) -
R(3,1)*cos(teta2_1)*cos(teta3_1)*cos(teta4_1)*cos(teta5_1
) -
R(1,1)*cos(teta1_1)*cos(teta2_1)*cos(teta5_1)*sin(teta4_1
) +
R(2,1)*cos(teta1_1)*cos(teta4_1)*cos(teta5_1)*sin(teta3_1
) -
R(1,1)*cos(teta4_1)*cos(teta5_1)*sin(teta1_1)*sin(teta3_1
) -
R(2,1)*cos(teta2_1)*cos(teta5_1)*sin(teta1_1)*sin(teta4_1
) +
R(1,1)*cos(teta1_1)*sin(teta2_1)*sin(teta3_1)*sin(teta5_1
) +
R(2,1)*sin(teta1_1)*sin(teta2_1)*sin(teta3_1)*sin(teta5_1
) -
R(1,1)*cos(teta1_1)*cos(teta3_1)*cos(teta4_1)*cos(teta5_1
)*sin(teta2_1) -
R(2,1)*cos(teta3_1)*cos(teta4_1)*cos(teta5_1)*sin(teta1_1
)*sin(teta2_1)), (R(2,2)*cos(teta1_1)*cos(teta3_1)*sin(tet
a5_1) - R(1,2)*cos(teta3_1)*sin(teta1_1)*sin(teta5_1) +
R(3,2)*cos(teta2_1)*sin(teta3_1)*sin(teta5_1) +
R(3,2)*cos(teta5_1)*sin(teta2_1)*sin(teta4_1) -
R(3,2)*cos(teta2_1)*cos(teta3_1)*cos(teta4_1)*cos(teta5_1
) -
R(1,2)*cos(teta1_1)*cos(teta2_1)*cos(teta5_1)*sin(teta4_1
) +

```

```

R(2,2)*cos(teta1_1)*cos(teta4_1)*cos(teta5_1)*sin(teta3_1
) -
R(1,2)*cos(teta4_1)*cos(teta5_1)*sin(teta1_1)*sin(teta3_1
) -
R(2,2)*cos(teta2_1)*cos(teta5_1)*sin(teta1_1)*sin(teta4_1
) +
R(1,2)*cos(teta1_1)*sin(teta2_1)*sin(teta3_1)*sin(teta5_1
) +
R(2,2)*sin(teta1_1)*sin(teta2_1)*sin(teta3_1)*sin(teta5_1
) -
R(1,2)*cos(teta1_1)*cos(teta3_1)*cos(teta4_1)*cos(teta5_1
)*sin(teta2_1) -
R(2,2)*cos(teta3_1)*cos(teta4_1)*cos(teta5_1)*sin(teta1_1
)*sin(teta2_1));
    teta_s1 =
[teta1_1,teta2_1,teta3_1,teta4_1,teta5_1,teta6_1];

    teta1_2 = atan2(T4d_2(2,4),T4d_2(1,4));
    teta2_2 =
atan2(sqrt(T4d_2(1,4)^2+T4d_2(2,4)^2),(T4d_2(3,4)-d1)) -
pi/2;
    teta3_2 = atan2((T3d_2(2,1)*cos(teta1_2) -
T3d_2(1,1)*sin(teta1_2)),(- T3d_2(3,1)*cos(teta2_2) -
T3d_2(1,1)*cos(teta1_2)*sin(teta2_2) -
T3d_2(2,1)*sin(teta1_2)*sin(teta2_2)));
    teta4_2 = atan2((T4d_2(3,1)*sin(teta2_2) -
T4d_2(1,1)*cos(teta1_2)*cos(teta2_2) -
T4d_2(2,1)*cos(teta2_2)*sin(teta1_2)),(T4d_2(2,1)*cos(tet
a1_2)*sin(teta3_2)-T4d_2(3,1)*cos(teta2_2)*cos(teta3_2) -
T4d_2(1,1)*sin(teta1_2)*sin(teta3_2) -
T4d_2(1,1)*cos(teta1_2)*cos(teta3_2)*sin(teta2_2) -
T4d_2(2,1)*cos(teta3_2)*sin(teta1_2)*sin(teta2_2)));
    teta5_2 =
atan2((T5d_2(2,1)*cos(teta1_2)*cos(teta3_2) -
T5d_2(1,1)*cos(teta3_2)*sin(teta1_2) +
T5d_2(3,1)*cos(teta2_2)*sin(teta3_2) +
T5d_2(1,1)*cos(teta1_2)*sin(teta2_2)*sin(teta3_2) +
T5d_2(2,1)*sin(teta1_2)*sin(teta2_2)*sin(teta3_2)),(T5d_2
(3,1)*sin(teta2_2)*sin(teta4_2) -
T5d_2(3,1)*cos(teta2_2)*cos(teta3_2)*cos(teta4_2) -
T5d_2(1,1)*cos(teta1_2)*cos(teta2_2)*sin(teta4_2) +
T5d_2(2,1)*cos(teta1_2)*cos(teta4_2)*sin(teta3_2) -
T5d_2(1,1)*cos(teta4_2)*sin(teta1_2)*sin(teta3_2) -
T5d_2(2,1)*cos(teta2_2)*sin(teta1_2)*sin(teta4_2) -
T5d_2(1,1)*cos(teta1_2)*cos(teta3_2)*cos(teta4_2)*sin(tet
a2_2) -
T5d_2(2,1)*cos(teta3_2)*cos(teta4_2)*sin(teta1_2)*sin(tet
a2_2)));
    teta6_2 =
atan2((R(2,1)*cos(teta1_2)*cos(teta3_2)*sin(teta5_2) -
R(1,1)*cos(teta3_2)*sin(teta1_2)*sin(teta5_2) +

```

```

R(3,1)*cos(teta2_2)*sin(teta3_2)*sin(teta5_2) +
R(3,1)*cos(teta5_2)*sin(teta2_2)*sin(teta4_2) -
R(3,1)*cos(teta2_2)*cos(teta3_2)*cos(teta4_2)*cos(teta5_2
) -
R(1,1)*cos(teta1_2)*cos(teta2_2)*cos(teta5_2)*sin(teta4_2
) +
R(2,1)*cos(teta1_2)*cos(teta4_2)*cos(teta5_2)*sin(teta3_2
) -
R(1,1)*cos(teta4_2)*cos(teta5_2)*sin(teta1_2)*sin(teta3_2
) -
R(2,1)*cos(teta2_2)*cos(teta5_2)*sin(teta1_2)*sin(teta4_2
) +
R(1,1)*cos(teta1_2)*sin(teta2_2)*sin(teta3_2)*sin(teta5_2
) +
R(2,1)*sin(teta1_2)*sin(teta2_2)*sin(teta3_2)*sin(teta5_2
) -
R(1,1)*cos(teta1_2)*cos(teta3_2)*cos(teta4_2)*cos(teta5_2
)*sin(teta2_2) -
R(2,1)*cos(teta3_2)*cos(teta4_2)*cos(teta5_2)*sin(teta1_2
)*sin(teta2_2)), (R(2,2)*cos(teta1_2)*cos(teta3_2)*sin(tet
a5_2) - R(1,2)*cos(teta3_2)*sin(teta1_2)*sin(teta5_2) +
R(3,2)*cos(teta2_2)*sin(teta3_2)*sin(teta5_2) +
R(3,2)*cos(teta5_2)*sin(teta2_2)*sin(teta4_2) -
R(3,2)*cos(teta2_2)*cos(teta3_2)*cos(teta4_2)*cos(teta5_2
) -
R(1,2)*cos(teta1_2)*cos(teta2_2)*cos(teta5_2)*sin(teta4_2
) +
R(2,2)*cos(teta1_2)*cos(teta4_2)*cos(teta5_2)*sin(teta3_2
) -
R(1,2)*cos(teta4_2)*cos(teta5_2)*sin(teta1_2)*sin(teta3_2
) -
R(2,2)*cos(teta2_2)*cos(teta5_2)*sin(teta1_2)*sin(teta4_2
) +
R(1,2)*cos(teta1_2)*sin(teta2_2)*sin(teta3_2)*sin(teta5_2
) +
R(2,2)*sin(teta1_2)*sin(teta2_2)*sin(teta3_2)*sin(teta5_2
) -
R(1,2)*cos(teta1_2)*cos(teta3_2)*cos(teta4_2)*cos(teta5_2
)*sin(teta2_2) -
R(2,2)*cos(teta3_2)*cos(teta4_2)*cos(teta5_2)*sin(teta1_2
)*sin(teta2_2));
    tetas2 =
[teta1_2,teta2_2,teta3_2,teta4_2,teta5_2,teta6_2];
    if ((teta_minr<=tetas1)&(tetas1<=teta_maxr))
        if (i==1)
            Xe = x;
            Ze_1 = Ze_1f;
            Ye_1 = Ye_1f;
            Ye_2 = Ye_2f;
            Ze_2 = Ze_2f;
            i=i+1;

```

```

        tetas_best1 = tetas1;
        tetas_best2 = tetas2;
    end
    if (B<C)
        Xe = x;
        Ze_1 = Ze_1f;
        Ye_1 = Ye_1f;
        tetas_best1 = tetas1;
    end
end
if ((teta_minr<=tetas2)&(tetas2<=teta_maxr))
    if (i==1)
        Xe = x;
        Ze_1 = Ze_1f;
        Ye_1 = Ye_1f;
        Ye_2 = Ye_2f;
        Ze_2 = Ze_2f;
        i=i+1;
        tetas_best1 = tetas1;
        tetas_best2 = tetas2;
    end
    if(B2<C2)
        Xe = x;
        Ye_2 = Ye_2f;
        Ze_2 = Ze_2f;
        tetas_best2 = tetas2;
    end
end
if (B<=B2)
    tetas_best = tetas_best1;
else
    tetas_best = tetas_best2;
end
end
end
teta1 = tetas_best(1);
teta2 = tetas_best(2);
teta3 = tetas_best(3);
teta4 = tetas_best(4);
teta5 = tetas_best(5);
teta6 = tetas_best(6);
end

```

## EKE

```
function[th_1,th_2,th_3,th_4,th_5,th_6,vel_1,vel_2,vel_3,
vel_4,vel_5,vel_6,acc_1,acc_2,acc_3,acc_4,acc_5,acc_6,time,
T] =
trapez_velocity_profile(th1_i,th2_i,th3_i,th4_i,th5_i,th6
_i,th1_f,th2_f,th3_f,th4_f,th5_f,th6_f,dt)

thi = [th1_i th2_i th3_i th4_i th5_i th6_i];
thf = [th1_f th2_f th3_f th4_f th5_f th6_f];
th = thf - thi;
max_vels_motor_rpm = [3190 3190 4250 4250 4250 4250];%rpm
max_vels_motor_radpersec = max_vels_motor_rpm*pi/30;
%rad/sec
max_vels = [max_vels_motor_radpersec(1)/160
max_vels_motor_radpersec(2)/160
max_vels_motor_radpersec(3)/120
max_vels_motor_radpersec(4)/120
max_vels_motor_radpersec(5)/120
max_vels_motor_radpersec(6)/120];%rad/sec output
(1:160,1:160,1:120,1:120,1:120,1:120)
time_maxaccs = 0.25; %time to achieve max vel
max_accs = max_vels/time_maxaccs; % rad/s^2
max_vel = max_vels(1); % rad/s
th_max = th(1);
if (abs(th(2))>abs(th_max))
    th_max = th(2);
    max_vel = max_vels(2);
end
if (abs(th(3))>abs(th_max))
    th_max = th(3);
    max_vel = max_vels(3);
end
if (abs(th(4))>abs(th_max))
    th_max = th(4);
    max_vel = max_vels(4);
end
if (abs(th(5))>abs(th_max))
    th_max = th(5);
    max_vel = max_vels(5);
end
if (abs(th(6))>abs(th_max))
    th_max = th(6);
    max_vel = max_vels(6);
end
s=1;
p=0.01;
for k=max_vel:-p:max_vels(2)
    for i=max_accs(6):-p:max_accs(2)
        max_accd = i;
```

```

max_veld = k;
acc_timed =max_veld/max_accd;
total_timed = acc_timed + abs(th_max)/max_veld;
for j=1:6
    vel_maxd(j) = (th(j) / (total_timed -
acc_timed));
    accd(j) = vel_maxd(j)/acc_timed;
end
if (abs(max_accs)>=abs(accd)-1e-2)
    if (abs(max_vels)>=abs(vel_maxd)-1e-2)
        if (s==1)
            max_acc = max_accd;
            acc_time = acc_timed;
            total_time = total_timed;
            vel_max = vel_maxd;
            acc = accd;
            s=2;
        end
    end
end
end
end
end
end

```

```

dt=0.0001;
k=1;
for t = 0:dt:total_time
    if(t<=acc_time)
        vel_1(k) = acc(1)*t;
        vel_2(k) = acc(2)*t;
        vel_3(k) = acc(3)*t;
        vel_4(k) = acc(4)*t;
        vel_5(k)= acc(5)*t;
        vel_6(k) = acc(6)*t;

        th_1(k) = thi(1) + acc(1)*t*t/2;
        th_2(k) = thi(2) + acc(2)*t*t/2;
        th_3(k) = thi(3) + acc(3)*t*t/2;
        th_4(k) = thi(4) + acc(4)*t*t/2;
        th_5(k) = thi(5) + acc(5)*t*t/2;
        th_6(k) = thi(6) + acc(6)*t*t/2;

        acc_1(k) = acc(1);
        acc_2(k) = acc(2);
        acc_3(k) = acc(3);
        acc_4(k) = acc(4);
        acc_5(k) = acc(5);
        acc_6(k) = acc(6);
    end

    if ((t>acc_time) && (t<(total_time-acc_time)))

```

```

    vel_1(k) = vel_max(1);
    vel_2(k) = vel_max(2);
    vel_3(k) = vel_max(3);
    vel_4(k) = vel_max(4);
    vel_5(k) = vel_max(5);
    vel_6(k) = vel_max(6);

    th_1(k) = th_1(k-1) + vel_1(k) * (dt);
    th_2(k) = th_2(k-1) + vel_2(k) * (dt);
    th_3(k) = th_3(k-1) + vel_3(k) * (dt);
    th_4(k) = th_4(k-1) + vel_4(k) * (dt);
    th_5(k) = th_5(k-1) + vel_5(k) * (dt);
    th_6(k) = th_6(k-1) + vel_6(k) * (dt);

    acc_1(k) = 0;
    acc_2(k) = 0;
    acc_3(k) = 0;
    acc_4(k) = 0;
    acc_5(k) = 0;
    acc_6(k) = 0;
end
if (t>=(total_time-acc_time))
    vel_1(k) = vel_max(1) - acc(1) * (t+acc_time-
total_time);
    vel_2(k) = vel_max(2) - acc(2) * (t+acc_time-
total_time);
    vel_3(k) = vel_max(3) - acc(3) * (t+acc_time-
total_time);
    vel_4(k) = vel_max(4) - acc(4) * (t+acc_time-
total_time);
    vel_5(k) = vel_max(5) - acc(5) * (t+acc_time-
total_time);
    vel_6(k) = vel_max(6) - acc(6) * (t+acc_time-
total_time);

    th_1(k) = th_1(k-1) + vel_1(k) * dt -
acc(1) * dt * dt / 2;
    th_2(k) = th_2(k-1) + vel_2(k) * dt -
acc(2) * dt * dt / 2;
    th_3(k) = th_3(k-1) + vel_3(k) * dt -
acc(3) * dt * dt / 2;
    th_4(k) = th_4(k-1) + vel_4(k) * dt -
acc(4) * dt * dt / 2;
    th_5(k) = th_5(k-1) + vel_5(k) * dt -
acc(5) * dt * dt / 2;
    th_6(k) = th_6(k-1) + vel_6(k) * dt -
acc(6) * dt * dt / 2;
    acc_1(k) = -acc(1);
    acc_2(k) = -acc(2);
    acc_3(k) = -acc(3);
    acc_4(k) = -acc(4);

```

```

        acc_5(k) = -acc(5);
        acc_6(k) = -acc(6);
    end
k=k+1;
end
time=0:dt:total_time;
vel = [vel_1; vel_2; vel_3; vel_4; vel_5; vel_6 ];
th = [th_1; th_2; th_3; th_4; th_5; th_6];
acc = [acc_1;acc_2;acc_3;acc_4;acc_5;acc_6];
t_ref = time';
th1_ref = th_1';th2_ref = th_2';th3_ref = th_3';th4_ref =
th_4';th5_ref = th_5';th6_ref = th_6';
vel1_ref = vel_1';vel2_ref = vel_2';vel3_ref =
vel_3';vel4_ref = vel_4';vel5_ref = vel_5';vel6_ref =
vel_6';
acc1_ref = acc_1';acc2_ref = acc_2';acc3_ref =
acc_3';acc4_ref = acc_4';acc5_ref = acc_5';acc6_ref =
acc_6';
T =total_time;
end

```

## EKF

```
function[th_1,th_2,th_3,th_4,th_5,th_6,vel_1,vel_2,vel_3,
vel_4,vel_5,vel_6,acc_1,acc_2,acc_3,acc_4,acc_5,acc_6,time,
T] =
cubic_velocity_profile(th1_i,th2_i,th3_i,th4_i,th5_i,th6_
i,th1_f,th2_f,th3_f,th4_f,th5_f,th6_f,dt)
dt = 0.0001;

thi = [th1_i th2_i th3_i th4_i th5_i th6_i];%rad
thf = [th1_f th2_f th3_f th4_f th5_f th6_f];%rad

th = thf-thi;
max_vels_motor_rpm = [3190 3190 4250 4250 4250 4250];%rpm
max_vels_motor_radpersec = max_vels_motor_rpm*pi/30;
%rad/sec
max_vels = [max_vels_motor_radpersec(1)/160
max_vels_motor_radpersec(2)/160
max_vels_motor_radpersec(3)/120
max_vels_motor_radpersec(4)/120
max_vels_motor_radpersec(5)/120
max_vels_motor_radpersec(6)/120];%rad/sec output
(1:160,1:160,1:120,1:120,1:120,1:120)
time_maxaccs = 0.1; %time to achieve max vel
max_accs = max_vels/time_maxaccs; % rad/s^2

time_s =
[(3*abs(th(1)))/(2*max_vels(1)), (3*abs(th(2)))/(2*max_vels
(2)), (3*abs(th(3)))/(2*max_vels(3)), (3*abs(th(4)))/(2*max
_vels(4)), (3*abs(th(5)))/(2*max_vels(5)), (3*abs(th(6)))/(
2*max_vels(6)), (sqrt(6*abs(th(1))/max_accs(1))), (sqrt(6*
abs(th(2))/max_accs(2))), (sqrt(6*abs(th(3))/max_accs(3)))
, (sqrt(6*abs(th(4))/max_accs(4))), (sqrt(6*abs(th(5))/max_
accs(5))), (sqrt(6*abs(th(6))/max_accs(6)))]];

total_time = max(time_s);
max_acc =
[(6*abs(th(1))/total_time^2), (6*abs(th(2))/total_time^2),
(6*abs(th(3))/total_time^2), (6*abs(th(4))/total_time^2), (
6*abs(th(5))/total_time^2), (6*abs(th(6))/total_time^2)];
max_vel =
[3*abs(th(1))/(2*total_time), 3*abs(th(2))/(2*total_time),
3*abs(th(3))/(2*total_time), 3*abs(th(4))/(2*total_time), 3
*abs(th(5))/(2*total_time), 3*abs(th(6))/(2*total_time)];
i=1;
for (t=0:dt:total_time)

    th_1(i) = thi(1) + (3*(t/total_time)^2 -
2*(t/total_time)^3)*th(1);
```

```

    th_2(i) = thi(2) + (3*(t/total_time)^2 -
2*(t/total_time)^3)*th(2);
    th_3(i) = thi(3) + (3*(t/total_time)^2 -
2*(t/total_time)^3)*th(3);
    th_4(i) = thi(4) + (3*(t/total_time)^2 -
2*(t/total_time)^3)*th(4);
    th_5(i) = thi(5) + (3*(t/total_time)^2 -
2*(t/total_time)^3)*th(5);
    th_6(i) = thi(6) + (3*(t/total_time)^2 -
2*(t/total_time)^3)*th(6);
    vel_1(i) = (6*t/total_time^2 -
6*t^2/total_time^3)*th(1);
    vel_2(i) = (6*t/total_time^2 -
6*t^2/total_time^3)*th(2);
    vel_3(i) = (6*t/total_time^2 -
6*t^2/total_time^3)*th(3);
    vel_4(i) = (6*t/total_time^2 -
6*t^2/total_time^3)*th(4);
    vel_5(i) = (6*t/total_time^2 -
6*t^2/total_time^3)*th(5);
    vel_6(i) = (6*t/total_time^2 -
6*t^2/total_time^3)*th(6);
    acc_1(i) = (6/(total_time^2) -
12*t/(total_time^3))*th(1);
    acc_2(i) = (6/(total_time^2) -
12*t/(total_time^3))*th(2);
    acc_3(i) = (6/(total_time^2) -
12*t/(total_time^3))*th(3);
    acc_4(i) = (6/(total_time^2) -
12*t/(total_time^3))*th(4);
    acc_5(i) = (6/(total_time^2) -
12*t/(total_time^3))*th(5);
    acc_6(i) = (6/(total_time^2) -
12*t/(total_time^3))*th(6);
    i=i+1;
end
i=i-1;
time=0:dt:total_time;
t_ref = time';
th1_ref = th_1';
th2_ref = th_2';th3_ref = th_3';th4_ref = th_4';th5_ref =
th_5';th6_ref = th_6';
vel1_ref = vel_1';vel2_ref = vel_2';vel3_ref =
vel_3';vel4_ref = vel_4';vel5_ref = vel_5';vel6_ref =
vel_6';
acc1_ref = acc_1';acc2_ref = acc_2';acc3_ref =
acc_3';acc4_ref = acc_4';acc5_ref = acc_5';acc6_ref =
acc_6';
T =total_time;
end

```

## EKG

$$J(1,1) = -d_5*(s\theta_4*(c\theta_1*s\theta_3 - c\theta_3*s\theta_1*s\theta_2) + c\theta_2*c\theta_4*s\theta_1) - a_6*s\theta_6*(c\theta_5*(c\theta_4*(c\theta_1*s\theta_3 - c\theta_3*s\theta_1*s\theta_2) - c\theta_2*s\theta_1*s\theta_4) + s\theta_5*(c\theta_1*c\theta_3 + s\theta_1*s\theta_2*s\theta_3)) - d_3*c\theta_2*s\theta_1 - a_6*c\theta_6*(s\theta_4*(c\theta_1*s\theta_3 - c\theta_3*s\theta_1*s\theta_2) + c\theta_2*c\theta_4*s\theta_1)$$

$$J(1,2) = -c\theta_1*(d_5*(c\theta_4*s\theta_2 + c\theta_2*c\theta_3*s\theta_4) + d_3*s\theta_2 + a_6*c\theta_6*(c\theta_4*s\theta_2 + c\theta_2*c\theta_3*s\theta_4) - a_6*s\theta_6*(c\theta_5*(s\theta_2*s\theta_4 - c\theta_2*c\theta_3*c\theta_4) + c\theta_2*s\theta_3*s\theta_5))$$

$$J(1,3) = d_5*c\theta_1*s\theta_2*s\theta_3*s\theta_4 - a_6*c\theta_3*c\theta_6*s\theta_1*s\theta_4 - d_5*c\theta_3*s\theta_1*s\theta_4 + a_6*s\theta_1*s\theta_3*s\theta_5*s\theta_6 - a_6*c\theta_3*c\theta_4*c\theta_5*s\theta_1*s\theta_6 + a_6*c\theta_1*c\theta_6*s\theta_2*s\theta_3*s\theta_4 + a_6*c\theta_1*c\theta_3*s\theta_2*s\theta_5*s\theta_6 + a_6*c\theta_1*c\theta_4*c\theta_5*s\theta_2*s\theta_3*s\theta_6$$

$$J(1,4) = a_6*c\theta_5*s\theta_1*s\theta_3*s\theta_4*s\theta_6 - d_5*c\theta_4*s\theta_1*s\theta_3 - a_6*c\theta_1*c\theta_2*c\theta_6*s\theta_4 - d_5*c\theta_1*c\theta_3*c\theta_4*s\theta_2 - a_6*c\theta_4*c\theta_6*s\theta_1*s\theta_3 - a_6*c\theta_1*c\theta_3*c\theta_4*c\theta_6*s\theta_2 - a_6*c\theta_1*c\theta_2*c\theta_4*c\theta_5*s\theta_6 - d_5*c\theta_1*c\theta_2*s\theta_4 + a_6*c\theta_1*c\theta_3*c\theta_5*s\theta_2*s\theta_4*s\theta_6$$

$$J(1,5) = a_6*s\theta_6*(c\theta_1*c\theta_5*s\theta_2*s\theta_3 - c\theta_3*c\theta_5*s\theta_1 + c\theta_1*c\theta_2*s\theta_4*s\theta_5 + c\theta_4*s\theta_1*s\theta_3*s\theta_5 + c\theta_1*c\theta_3*c\theta_4*s\theta_2*s\theta_5)$$

$$J(1,6) = a_6*s\theta_1*s\theta_3*s\theta_4*s\theta_6 - a_6*c\theta_3*c\theta_6*s\theta_1*s\theta_5 - a_6*c\theta_1*c\theta_2*c\theta_4*s\theta_6 - a_6*c\theta_1*c\theta_2*c\theta_5*c\theta_6*s\theta_4 - a_6*c\theta_4*c\theta_5*c\theta_6*s\theta_1*s\theta_3 + a_6*c\theta_1*c\theta_3*s\theta_2*s\theta_4*s\theta_6 + a_6*c\theta_1*c\theta_6*s\theta_2*s\theta_3*s\theta_5 - a_6*c\theta_1*c\theta_3*c\theta_4*c\theta_5*c\theta_6*s\theta_2$$

$$J(2,1) = d_3*c\theta_1*c\theta_2 - a_6*s\theta_6*(c\theta_5*(c\theta_4*(s\theta_1*s\theta_3 + c\theta_1*c\theta_3*s\theta_2) + c\theta_1*c\theta_2*s\theta_4) + s\theta_5*(c\theta_3*s\theta_1 - c\theta_1*s\theta_2*s\theta_3)) - d_5*(s\theta_4*(s\theta_1*s\theta_3 + c\theta_1*c\theta_3*s\theta_2) - c\theta_1*c\theta_2*c\theta_4) - a_6*c\theta_6*(s\theta_4*(s\theta_1*s\theta_3 + c\theta_1*c\theta_3*s\theta_2) - c\theta_1*c\theta_2*c\theta_4)$$

$$J(2,2) = -s\theta_1*(d_5*(c\theta_4*s\theta_2 + c\theta_2*c\theta_3*s\theta_4) + d_3*s\theta_2 + a_6*c\theta_6*(c\theta_4*s\theta_2 + c\theta_2*c\theta_3*s\theta_4) - a_6*s\theta_6*(c\theta_5*(s\theta_2*s\theta_4 - c\theta_2*c\theta_3*c\theta_4) + c\theta_2*s\theta_3*s\theta_5))$$

$$J(2,3) = d_5*c\theta_1*c\theta_3*s\theta_4 + a_6*c\theta_1*c\theta_3*c\theta_6*s\theta_4 - a_6*c\theta_1*s\theta_3*s\theta_5*s\theta_6 + d_5*s\theta_1*s\theta_2*s\theta_3*s\theta_4 + a_6*c\theta_1*c\theta_3*c\theta_4*c\theta_5*s\theta_6 + a_6*c\theta_6*s\theta_1*s\theta_2*s\theta_3*s\theta_4 + a_6*c\theta_3*s\theta_1*s\theta_2*s\theta_5*s\theta_6 + a_6*c\theta_4*c\theta_5*s\theta_1*s\theta_2*s\theta_3*s\theta_6$$

$$J(2,4) = d_5*c\theta_1*c\theta_4*s\theta_3 - d_5*c\theta_2*s\theta_1*s\theta_4 + a_6*c\theta_1*c\theta_4*c\theta_6*s\theta_3 - a_6*c\theta_2*c\theta_6*s\theta_1*s\theta_4 - d_5*c\theta_3*c\theta_4*s\theta_1*s\theta_2 - a_6*c\theta_3*c\theta_4*c\theta_6*s\theta_1*s\theta_2 - a_6*c\theta_2*c\theta_4*c\theta_5*s\theta_1*s\theta_6 - a_6*c\theta_1*c\theta_5*s\theta_3*s\theta_4*s\theta_6 + a_6*c\theta_3*c\theta_5*s\theta_1*s\theta_2*s\theta_4*s\theta_6$$

$$J(2,5) = a_6*s\theta_6*(c\theta_1*c\theta_3*c\theta_5 - c\theta_1*c\theta_4*s\theta_3*s\theta_5 + c\theta_5*s\theta_1*s\theta_2*s\theta_3 + c\theta_2*s\theta_1*s\theta_4*s\theta_5 + c\theta_3*c\theta_4*s\theta_1*s\theta_2*s\theta_5)$$

$$J(2,6) = a_6*c\theta_1*c\theta_3*c\theta_6*s\theta_5 - a_6*c\theta_2*c\theta_4*s\theta_1*s\theta_6 - a_6*c\theta_1*s\theta_3*s\theta_4*s\theta_6 + a_6*c\theta_1*c\theta_4*c\theta_5*c\theta_6*s\theta_3 - a_6*c\theta_2*c\theta_5*c\theta_6*s\theta_1*s\theta_4 + a_6*c\theta_3*s\theta_1*s\theta_2*s\theta_4*s\theta_6 + a_6*c\theta_6*s\theta_1*s\theta_2*s\theta_3*s\theta_5 - a_6*c\theta_3*c\theta_4*c\theta_5*c\theta_6*s\theta_1*s\theta_2$$

$$J(3,1) = 0$$

$$J(3,2) = d_5*c\theta_3*s\theta_2*s\theta_4 - d_5*c\theta_2*c\theta_4 - a_6*c\theta_2*c\theta_4*c\theta_6 - d_3*c\theta_2 + a_6*c\theta_3*c\theta_6*s\theta_2*s\theta_4 + a_6*c\theta_2*c\theta_5*s\theta_4*s\theta_6 - a_6*s\theta_2*s\theta_3*s\theta_5*s\theta_6 + a_6*c\theta_3*c\theta_4*c\theta_5*s\theta_2*s\theta_6$$

$$J(3,3) = c\theta_2*(d_5*s\theta_3*s\theta_4 + a_6*c\theta_6*s\theta_3*s\theta_4 + a_6*c\theta_3*s\theta_5*s\theta_6 + a_6*c\theta_4*c\theta_5*s\theta_3*s\theta_6)$$

$$J(3,4) = d_5*s\theta_2*s\theta_4 - d_5*c\theta_2*c\theta_3*c\theta_4 + a_6*c\theta_6*s\theta_2*s\theta_4 - a_6*c\theta_2*c\theta_3*c\theta_4*c\theta_6 + a_6*c\theta_4*c\theta_5*s\theta_2*s\theta_6 + a_6*c\theta_2*c\theta_3*c\theta_5*s\theta_4*s\theta_6$$

$$J(3,5) = a_6*s\theta_6*(c\theta_2*c\theta_5*s\theta_3 - s\theta_2*s\theta_4*s\theta_5 + c\theta_2*c\theta_3*c\theta_4*s\theta_5)$$

$$J(3,6) = a_6*c\theta_4*s\theta_2*s\theta_6 + a_6*c\theta_2*c\theta_3*s\theta_4*s\theta_6 + a_6*c\theta_2*c\theta_6*s\theta_3*s\theta_5 + a_6*c\theta_5*c\theta_6*s\theta_2*s\theta_4 - a_6*c\theta_2*c\theta_3*c\theta_4*c\theta_5*c\theta_6$$

$$J(4,1) = 0$$

$$J(4,2) = -s\theta_1$$

$$J(4,3) = c\theta_1 * c\theta_2$$

$$J(4,4) = c\theta_1 * s\theta_2 * s\theta_3 - c\theta_3 * s\theta_1$$

$$J(4,5) = c\theta_1 * c\theta_2 * c\theta_4 - s\theta_4 * (s\theta_1 * s\theta_3 + c\theta_1 * c\theta_3 * s\theta_2)$$

$$J(4,6) = s\theta_5 * (c\theta_4 * (s\theta_1 * s\theta_3 + c\theta_1 * c\theta_3 * s\theta_2) + c\theta_1 * c\theta_2 * s\theta_4) - c\theta_5 * (c\theta_3 * s\theta_1 - c\theta_1 * s\theta_2 * s\theta_3)$$

$$J(5,1) = 0$$

$$J(5,2) = c\theta_1$$

$$J(5,3) = c\theta_2 * s\theta_1$$

$$J(5,4) = c\theta_1 * c\theta_3 + s\theta_1 * s\theta_2 * s\theta_3$$

$$J(5,5) = s\theta_4 * (c\theta_1 * s\theta_3 - c\theta_3 * s\theta_1 * s\theta_2) + c\theta_2 * c\theta_4 * s\theta_1$$

$$J(5,6) = c\theta_5 * (c\theta_1 * c\theta_3 + s\theta_1 * s\theta_2 * s\theta_3) - s\theta_5 * (c\theta_4 * (c\theta_1 * s\theta_3 - c\theta_3 * s\theta_1 * s\theta_2) - c\theta_2 * s\theta_1 * s\theta_4)$$

$$J(6,1) = 1$$

$$J(6,2) = 0$$

$$J(6,3) = -s\theta_2$$

$$J(6,4) = c\theta_2 * s\theta_3$$

$$J(6,5) = -c\theta_4 * s\theta_2 - c\theta_2 * c\theta_3 * s\theta_4$$

$$J(6,6) = c\theta_2 * c\theta_5 * s\theta_3 - s\theta_5 * (s\theta_2 * s\theta_4 - c\theta_2 * c\theta_3 * c\theta_4)$$

## EK H

```
function out = Dmatrix(inp)
teta1 = inp(1);
teta2 = inp(2);
teta3 = inp(3);
teta4 = inp(4);
teta5 = inp(5);
teta6 = inp(6);
acc1 = inp(7);
acc2 = inp(8);
acc3 = inp(9);
acc4 = inp(10);
acc5 = inp(11);
acc6 = inp(12);
accs = [acc1;acc2;acc3;acc4;acc5;acc6];
a6 = 166.31/1000;%m
d1 = 75.5/1000;%m
d3 = 225/1000;%m
d5 = 214/1000;%m
m1 = 1.3168; %kg
m2 = 1.22315; %kg
m3 = 0.983142; %kg
m4 = 1.08946; %kg
m5 = 0.983321;%kg
m6 = 1.10972;%kg yüksüz
% m6 = 3.72661;%3 kg yükle % m6 = 5.74179;%5 kg yükle
I1 = [0.00124317,-9.57806e-06,-0.000342202;-9.57806e-
06,0.0027207,-9.73026e-06;-0.000342202,-9.73026e-
06,0.00238551]; %kg*m^2
I2 = [0.00110719,-0.000114989,-5.07478e-05;-
0.000114989,0.00188872,-0.0001273;-5.07478e-05,-
0.0001273,0.00229854]; %kg*m^2
I3 = [0.00170739,9.8613e-10,-1.66948e-10;9.8613e-
10,0.00130479,-0.000214806;-1.66948e-10,-
0.000214806,0.000815547];%kg*m^2
I4 = [0.00126424,0.000135009,-
0.000102553;0.000135009,0.00147851,0.000316009;-
0.000102553,0.000316009,0.00111851];%kg*m^2
I5 = [0.00170742,7.82017e-10,-1.85372e-09;7.82017e-
10,0.000815679,0.000214671;-1.85372e-
09,0.000214671,0.0013047];%kg*m^2
I6 = [0.000524483,8.05412e-06,0.000257773;8.05412e-
06,0.00203666,-1.37253e-06;0.000257773,-1.37253e-
06,0.00186637];%kg*m^2 yüksüz
% I6 =
[0.004941499,0.000004854,0.000325608;0.000004854,0.005852
964,-0.000001733;0.000325608,-
0.000001733,0.005675856];%kg*m^2 3 kg yükle
```

```

% I6 = [0.013471657,-0.000016424,0.000331398;-
0.000016424,0.010388004,-0.000001764;0.000331398,-
0.000001764,0.010209771];%kg*m^2 5 kg ykle
Jvc1 =[ 0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, 0, 0];

Jwc1 = [0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, 0, 0;
        1, 0, 0, 0, 0, 0];

R1 = [ cos(tetal), 0, -sin(tetal);
       sin(tetal), 0,  cos(tetal);
       0, -1, 0];

Jvc2 =[ 0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, 0, 0];

Jwc2 = [ 0, -sin(tetal), 0, 0, 0, 0;
        0,  cos(tetal), 0, 0, 0, 0;
        1, 0, 0, 0, 0, 0];

R2 = [ -cos(tetal)*sin(teta2), -sin(tetal),
cos(tetal)*cos(teta2);
       -sin(tetal)*sin(teta2),  cos(tetal),
cos(teta2)*sin(tetal);
       -cos(teta2), 0, -sin(teta2)];

Jvc3 = [ -d3*cos(teta2)*sin(tetal), -
d3*cos(tetal)*sin(teta2), 0, 0, 0, 0;
        d3*cos(tetal)*cos(teta2), -
d3*sin(tetal)*sin(teta2), 0, 0, 0, 0;
        0, - d3*cos(tetal)^2*cos(teta2) -
d3*cos(teta2)*sin(tetal)^2, 0, 0, 0, 0];

Jwc3 = [ 0, -sin(tetal), cos(tetal)*cos(teta2), 0, 0, 0;
        0,  cos(tetal), cos(teta2)*sin(tetal), 0, 0, 0;
        1, 0, -sin(teta2), 0, 0, 0];

R3 =[ - sin(tetal)*sin(teta3) -
cos(tetal)*cos(teta3)*sin(teta2), -cos(tetal)*cos(teta2),
cos(tetal)*sin(teta2)*sin(teta3) - cos(teta3)*sin(tetal);
       cos(tetal)*sin(teta3) -
cos(teta3)*sin(tetal)*sin(teta2), -cos(teta2)*sin(tetal),
cos(tetal)*cos(teta3) + sin(tetal)*sin(teta2)*sin(teta3);
       -cos(teta2)*cos(teta3),
sin(teta2),
cos(teta2)*sin(teta3)];

```

```

Jvc4 = [ -d3*cos(teta2)*sin(teta1),-
d3*cos(teta1)*sin(teta2), 0, 0, 0, 0;
          d3*cos(teta1)*cos(teta2),-
d3*sin(teta1)*sin(teta2), 0, 0, 0, 0;
          0, - d3*cos(teta1)^2*cos(teta2) -
d3*cos(teta2)*sin(teta1)^2, 0, 0, 0, 0];

Jwc4 = [ 0, -sin(teta1), cos(teta1)*cos(teta2),
cos(teta1)*sin(teta2)*sin(teta3) - cos(teta3)*sin(teta1),
0, 0;
          0, cos(teta1), cos(teta2)*sin(teta1),
cos(teta1)*cos(teta3) + sin(teta1)*sin(teta2)*sin(teta3),
0, 0;
          1, 0, -sin(teta2),
cos(teta2)*sin(teta3), 0, 0];

R4 = [ - cos(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*sin(teta4),
cos(teta1)*sin(teta2)*sin(teta3) - cos(teta3)*sin(teta1),
cos(teta1)*cos(teta2)*cos(teta4) -
sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2));
          cos(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4), cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3),
sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1);

sin(teta2)*sin(teta4) - cos(teta2)*cos(teta3)*cos(teta4),
cos(teta2)*sin(teta3),
- cos(teta4)*sin(teta2) -
cos(teta2)*cos(teta3)*sin(teta4)];

Jvc5 = [ - d5*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)) -
d3*cos(teta2)*sin(teta1), -
cos(teta1)*(d5*(cos(teta4)*sin(teta2) +
cos(teta2)*cos(teta3)*sin(teta4)) + d3*sin(teta2)), -
d5*sin(teta4)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3)), -
d5*cos(teta1)*cos(teta2)*sin(teta4) -
d5*cos(teta4)*sin(teta1)*sin(teta3) -
d5*cos(teta1)*cos(teta3)*cos(teta4)*sin(teta2), 0, 0;
          d3*cos(teta1)*cos(teta2) -
d5*(sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*cos(teta4)), -

```

```

sin(teta1)*(d5*(cos(teta4)*sin(teta2) +
cos(teta2)*cos(teta3)*sin(teta4)) + d3*sin(teta2)),
d5*sin(teta4)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3)),
d5*cos(teta1)*cos(teta4)*sin(teta3) -
d5*cos(teta2)*sin(teta1)*sin(teta4) -
d5*cos(teta3)*cos(teta4)*sin(teta1)*sin(teta2), 0, 0;
0, d5*cos(teta3)*sin(teta2)*sin(teta4) -
d5*cos(teta2)*cos(teta4) -
d3*cos(teta2), d5*cos(teta2)*sin(teta3)*sin(teta4),
d5*sin(teta2)*sin(teta4) -
d5*cos(teta2)*cos(teta3)*cos(teta4), 0, 0];

```

```

Jwc5 = [ 0, -sin(teta1), cos(teta1)*cos(teta2),
cos(teta1)*sin(teta2)*sin(teta3) - cos(teta3)*sin(teta1),
cos(teta1)*cos(teta2)*cos(teta4) -
sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)), 0;
0, cos(teta1), cos(teta2)*sin(teta1),
cos(teta1)*cos(teta3) + sin(teta1)*sin(teta2)*sin(teta3),
sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1), 0;
1, 0, -sin(teta2),
cos(teta2)*sin(teta3),
- cos(teta4)*sin(teta2) -
cos(teta2)*cos(teta3)*sin(teta4), 0];

```

```

R5 = [ - cos(teta5)*(cos(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) +
cos(teta1)*cos(teta2)*sin(teta4)) -
sin(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3)),
sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*cos(teta4),
sin(teta5)*(cos(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) +
cos(teta1)*cos(teta2)*sin(teta4)) -
cos(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3));
cos(teta5)*(cos(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4)) +
sin(teta5)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3)), -
sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*cos(teta4)*sin(teta1),
cos(teta5)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3)) -

```

```

sin(teta5) * (cos(teta4) * (cos(teta1) * sin(teta3) -
cos(teta3) * sin(teta1) * sin(teta2)) -
cos(teta2) * sin(teta1) * sin(teta4));

cos(teta5) * (sin(teta2) * sin(teta4) -
cos(teta2) * cos(teta3) * cos(teta4)) +
cos(teta2) * sin(teta3) * sin(teta5),
cos(teta4) * sin(teta2) + cos(teta2) * cos(teta3) * sin(teta4),
cos(teta2) * cos(teta5) * sin(teta3) -
sin(teta5) * (sin(teta2) * sin(teta4) -
cos(teta2) * cos(teta3) * cos(teta4))]);

Jvc6 = [ - d5 * (sin(teta4) * (cos(teta1) * sin(teta3) -
cos(teta3) * sin(teta1) * sin(teta2)) +
cos(teta2) * cos(teta4) * sin(teta1)) -
a6 * sin(teta6) * (cos(teta5) * (cos(teta4) * (cos(teta1) * sin(teta3) -
cos(teta3) * sin(teta1) * sin(teta2)) -
cos(teta2) * sin(teta1) * sin(teta4)) +
sin(teta5) * (cos(teta1) * cos(teta3) +
sin(teta1) * sin(teta2) * sin(teta3))) -
d3 * cos(teta2) * sin(teta1) -
a6 * cos(teta6) * (sin(teta4) * (cos(teta1) * sin(teta3) -
cos(teta3) * sin(teta1) * sin(teta2)) +
cos(teta2) * cos(teta4) * sin(teta1)), -
cos(teta1) * (d5 * (cos(teta4) * sin(teta2) +
cos(teta2) * cos(teta3) * sin(teta4)) + d3 * sin(teta2) +
a6 * cos(teta6) * (cos(teta4) * sin(teta2) +
cos(teta2) * cos(teta3) * sin(teta4)) -
a6 * sin(teta6) * (cos(teta5) * (sin(teta2) * sin(teta4) -
cos(teta2) * cos(teta3) * cos(teta4)) +
cos(teta2) * sin(teta3) * sin(teta5))),
d5 * cos(teta1) * sin(teta2) * sin(teta3) * sin(teta4) -
a6 * cos(teta3) * cos(teta6) * sin(teta1) * sin(teta4) -
d5 * cos(teta3) * sin(teta1) * sin(teta4) +
a6 * sin(teta1) * sin(teta3) * sin(teta5) * sin(teta6) -
a6 * cos(teta3) * cos(teta4) * cos(teta5) * sin(teta1) * sin(teta6)
+
a6 * cos(teta1) * cos(teta6) * sin(teta2) * sin(teta3) * sin(teta4)
+
a6 * cos(teta1) * cos(teta3) * sin(teta2) * sin(teta5) * sin(teta6)
+
a6 * cos(teta1) * cos(teta4) * cos(teta5) * sin(teta2) * sin(teta3)
* sin(teta6),
a6 * cos(teta5) * sin(teta1) * sin(teta3) * sin(teta4) * sin(teta6)
- d5 * cos(teta4) * sin(teta1) * sin(teta3) -
a6 * cos(teta1) * cos(teta2) * cos(teta6) * sin(teta4) -
d5 * cos(teta1) * cos(teta3) * cos(teta4) * sin(teta2) -
a6 * cos(teta4) * cos(teta6) * sin(teta1) * sin(teta3) -
a6 * cos(teta1) * cos(teta3) * cos(teta4) * cos(teta6) * sin(teta2)
-
a6 * cos(teta1) * cos(teta2) * cos(teta4) * cos(teta5) * sin(teta6)

```

```

- d5*cos(teta1)*cos(teta2)*sin(teta4) +
a6*cos(teta1)*cos(teta3)*cos(teta5)*sin(teta2)*sin(teta4)
*sin(teta6),
a6*sin(teta6)*(cos(teta1)*cos(teta5)*sin(teta2)*sin(teta3)
) - cos(teta3)*cos(teta5)*sin(teta1) +
cos(teta1)*cos(teta2)*sin(teta4)*sin(teta5) +
cos(teta4)*sin(teta1)*sin(teta3)*sin(teta5) +
cos(teta1)*cos(teta3)*cos(teta4)*sin(teta2)*sin(teta5)),
a6*sin(teta1)*sin(teta3)*sin(teta4)*sin(teta6) -
a6*cos(teta3)*cos(teta6)*sin(teta1)*sin(teta5) -
a6*cos(teta1)*cos(teta2)*cos(teta4)*sin(teta6) -
a6*cos(teta1)*cos(teta2)*cos(teta5)*cos(teta6)*sin(teta4)
-
a6*cos(teta4)*cos(teta5)*cos(teta6)*sin(teta1)*sin(teta3)
+
a6*cos(teta1)*cos(teta3)*sin(teta2)*sin(teta4)*sin(teta6)
+
a6*cos(teta1)*cos(teta6)*sin(teta2)*sin(teta3)*sin(teta5)
-
a6*cos(teta1)*cos(teta3)*cos(teta4)*cos(teta5)*cos(teta6)
*sin(teta2);
    d3*cos(teta1)*cos(teta2) -
a6*sin(teta6)*(cos(teta5)*(cos(teta4)*(sin(teta1)*sin(tet
a3) + cos(teta1)*cos(teta3)*sin(teta2)) +
cos(teta1)*cos(teta2)*sin(teta4)) +
sin(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3))) -
d5*(sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*cos(teta4)) -
a6*cos(teta6)*(sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*cos(teta4)), -
sin(teta1)*(d5*(cos(teta4)*sin(teta2) +
cos(teta2)*cos(teta3)*sin(teta4)) + d3*sin(teta2) +
a6*cos(teta6)*(cos(teta4)*sin(teta2) +
cos(teta2)*cos(teta3)*sin(teta4)) -
a6*sin(teta6)*(cos(teta5)*(sin(teta2)*sin(teta4) -
cos(teta2)*cos(teta3)*cos(teta4)) +
cos(teta2)*sin(teta3)*sin(teta5))),
d5*cos(teta1)*cos(teta3)*sin(teta4) +
a6*cos(teta1)*cos(teta3)*cos(teta6)*sin(teta4) -
a6*cos(teta1)*sin(teta3)*sin(teta5)*sin(teta6) +
d5*sin(teta1)*sin(teta2)*sin(teta3)*sin(teta4) +
a6*cos(teta1)*cos(teta3)*cos(teta4)*cos(teta5)*sin(teta6)
+
a6*cos(teta6)*sin(teta1)*sin(teta2)*sin(teta3)*sin(teta4)
+
a6*cos(teta3)*sin(teta1)*sin(teta2)*sin(teta5)*sin(teta6)
+
a6*cos(teta4)*cos(teta5)*sin(teta1)*sin(teta2)*sin(teta3)

```

```

* sin(teta6), d5*cos(teta1)*cos(teta4)*sin(teta3) -
d5*cos(teta2)*sin(teta1)*sin(teta4) +
a6*cos(teta1)*cos(teta4)*cos(teta6)*sin(teta3) -
a6*cos(teta2)*cos(teta6)*sin(teta1)*sin(teta4) -
d5*cos(teta3)*cos(teta4)*sin(teta1)*sin(teta2) -
a6*cos(teta3)*cos(teta4)*cos(teta6)*sin(teta1)*sin(teta2)
-
a6*cos(teta2)*cos(teta4)*cos(teta5)*sin(teta1)*sin(teta6)
-
a6*cos(teta1)*cos(teta5)*sin(teta3)*sin(teta4)*sin(teta6)
+
a6*cos(teta3)*cos(teta5)*sin(teta1)*sin(teta2)*sin(teta4)
*sin(teta6),
a6*sin(teta6)*(cos(teta1)*cos(teta3)*cos(teta5) -
cos(teta1)*cos(teta4)*sin(teta3)*sin(teta5) +
cos(teta5)*sin(teta1)*sin(teta2)*sin(teta3) +
cos(teta2)*sin(teta1)*sin(teta4)*sin(teta5) +
cos(teta3)*cos(teta4)*sin(teta1)*sin(teta2)*sin(teta5)),
a6*cos(teta1)*cos(teta3)*cos(teta6)*sin(teta5) -
a6*cos(teta2)*cos(teta4)*sin(teta1)*sin(teta6) -
a6*cos(teta1)*sin(teta3)*sin(teta4)*sin(teta6) +
a6*cos(teta1)*cos(teta4)*cos(teta5)*cos(teta6)*sin(teta3)
-
a6*cos(teta2)*cos(teta5)*cos(teta6)*sin(teta1)*sin(teta4)
+
a6*cos(teta3)*sin(teta1)*sin(teta2)*sin(teta4)*sin(teta6)
+
a6*cos(teta6)*sin(teta1)*sin(teta2)*sin(teta3)*sin(teta5)
-
a6*cos(teta3)*cos(teta4)*cos(teta5)*cos(teta6)*sin(teta1)
*sin(teta2);
0, d5*cos(teta3)*sin(teta2)*sin(teta4) -
d5*cos(teta2)*cos(teta4) -
a6*cos(teta2)*cos(teta4)*cos(teta6) - d3*cos(teta2) +
a6*cos(teta3)*cos(teta6)*sin(teta2)*sin(teta4) +
a6*cos(teta2)*cos(teta5)*sin(teta4)*sin(teta6) -
a6*sin(teta2)*sin(teta3)*sin(teta5)*sin(teta6) +
a6*cos(teta3)*cos(teta4)*cos(teta5)*sin(teta2)*sin(teta6)
, cos(teta2)*(d5*sin(teta3)*sin(teta4) +
a6*cos(teta6)*sin(teta3)*sin(teta4) +
a6*cos(teta3)*sin(teta5)*sin(teta6) +
a6*cos(teta4)*cos(teta5)*sin(teta3)*sin(teta6)), d5*sin(te
ta2)*sin(teta4) - d5*cos(teta2)*cos(teta3)*cos(teta4) +
a6*cos(teta6)*sin(teta2)*sin(teta4) -
a6*cos(teta2)*cos(teta3)*cos(teta4)*cos(teta6) +
a6*cos(teta4)*cos(teta5)*sin(teta2)*sin(teta6) +
a6*cos(teta2)*cos(teta3)*cos(teta5)*sin(teta4)*sin(teta6)
, a6*sin(teta6)*(cos(teta2)*cos(teta5)*sin(teta3) -
sin(teta2)*sin(teta4)*sin(teta5) +
cos(teta2)*cos(teta3)*cos(teta4)*sin(teta5)), a6*cos(teta4)
)*sin(teta2)*sin(teta6) +

```

```

a6*cos(teta2)*cos(teta3)*sin(teta4)*sin(teta6) +
a6*cos(teta2)*cos(teta6)*sin(teta3)*sin(teta5) +
a6*cos(teta5)*cos(teta6)*sin(teta2)*sin(teta4) -
a6*cos(teta2)*cos(teta3)*cos(teta4)*cos(teta5)*cos(teta6)
];
Jwc6 = [ 0, -sin(teta1), cos(teta1)*cos(teta2),
cos(teta1)*sin(teta2)*sin(teta3) - cos(teta3)*sin(teta1),
cos(teta1)*cos(teta2)*cos(teta4) -
sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)),
sin(teta5)*(cos(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) +
cos(teta1)*cos(teta2)*sin(teta4)) -
cos(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3));
0, cos(teta1), cos(teta2)*sin(teta1),
cos(teta1)*cos(teta3) + sin(teta1)*sin(teta2)*sin(teta3),
sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1),
cos(teta5)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3)) -
sin(teta5)*(cos(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4));
1, 0, -sin(teta2),
cos(teta2)*sin(teta3),
- cos(teta4)*sin(teta2) -
cos(teta2)*cos(teta3)*sin(teta4),
cos(teta2)*cos(teta5)*sin(teta3) -
sin(teta5)*(sin(teta2)*sin(teta4) -
cos(teta2)*cos(teta3)*cos(teta4))];

R6 = [ - cos(teta6)*(sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*cos(teta4)) -
sin(teta6)*(cos(teta5)*(cos(teta4)*(sin(teta1)*sin(teta3)
+ cos(teta1)*cos(teta3)*sin(teta2)) +
cos(teta1)*cos(teta2)*sin(teta4)) +
sin(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3))),
sin(teta6)*(sin(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) -
cos(teta1)*cos(teta2)*cos(teta4)) -
cos(teta6)*(cos(teta5)*(cos(teta4)*(sin(teta1)*sin(teta3)
+ cos(teta1)*cos(teta3)*sin(teta2)) +
cos(teta1)*cos(teta2)*sin(teta4)) +
sin(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3))),
sin(teta5)*(cos(teta4)*(sin(teta1)*sin(teta3) +
cos(teta1)*cos(teta3)*sin(teta2)) +

```

```

cos(teta1)*cos(teta2)*sin(teta4) -
cos(teta5)*(cos(teta3)*sin(teta1) -
cos(teta1)*sin(teta2)*sin(teta3));
    cos(teta6)*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)) +
sin(teta6)*(cos(teta5)*(cos(teta4)*(cos(teta1)*sin(teta3)
- cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4)) +
sin(teta5)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3))),
cos(teta6)*(cos(teta5)*(cos(teta4)*(cos(teta1)*sin(teta3)
- cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4)) +
sin(teta5)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3))) -
sin(teta6)*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)),
cos(teta5)*(cos(teta1)*cos(teta3) +
sin(teta1)*sin(teta2)*sin(teta3)) -
sin(teta5)*(cos(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4));

```

```

sin(teta6)*(cos(teta5)*(sin(teta2)*sin(teta4) -
cos(teta2)*cos(teta3)*cos(teta4)) +
cos(teta2)*sin(teta3)*sin(teta5)) -
cos(teta6)*(cos(teta4)*sin(teta2) +
cos(teta2)*cos(teta3)*sin(teta4)),
cos(teta6)*(cos(teta5)*(sin(teta2)*sin(teta4) -
cos(teta2)*cos(teta3)*cos(teta4)) +
cos(teta2)*sin(teta3)*sin(teta5)) +
sin(teta6)*(cos(teta4)*sin(teta2) +
cos(teta2)*cos(teta3)*sin(teta4)),
cos(teta2)*cos(teta5)*sin(teta3) -
sin(teta5)*(sin(teta2)*sin(teta4) -
cos(teta2)*cos(teta3)*cos(teta4))]];

```

```

D = m1*Jvc1'*Jvc1 + m2*Jvc2'*Jvc2 + m3*Jvc3'*Jvc3 +
m4*Jvc4'*Jvc4 + m5*Jvc5'*Jvc5 + m6*Jvc6'*Jvc6 +
Jwc1'*R1*I1*R1'*Jwc1 + Jwc2'*R2*I2*R2'*Jwc2 +
Jwc3'*R3*I3*R3'*Jwc3 + Jwc4'*R4*I4*R4'*Jwc4 +
Jwc5'*R5*I5*R5'*Jwc5 + Jwc6'*R6*I6*R6'*Jwc6;
tav = D*accs;
out(1) = tav(1)*1000; %mNm
out(2) = tav(2)*1000; %mNm
out(3) = tav(3)*1000; %mNm
out(4) = tav(4)*1000; %mNm
out(5) = tav(5)*1000; %mNm
out(6) = tav(6)*1000; %mNm

```

## EK I

```
function out = Gmatrix(inp)

teta1 = inp(1);
teta2 = inp(2);
teta3 = inp(3);
teta4 = inp(4);
teta5 = inp(5);
teta6 = inp(6);

a6 = 166.31;
d1 = 75.5;
d3 = 225;
d5 = 214;

m1 = 1.3168; %kg
m2 = 1.22315; %kg
m3 = 0.983142; %kg
m4 = 1.08946; %kg
m5 = 0.983321;%kg
m6 = 1.10972;%kg yüksüz
% m6 = 3.72661;%3 kg yükle
% m6 = 5.74179;%5 kg yükle

m2c = 76.77;%mm
m3c = 205.19;%mm
m4c = 70.55;%mm
m5c = 194.2;%mm
m6c = 80.29;%mm

g1 = (981*m4*(m4c*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)) +
d3*cos(teta2)*sin(teta1))/100 +
(981*m5*(m5c*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)) +
d3*cos(teta2)*sin(teta1))/100 +
(981*m6*(d5*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)) -
m6c*(cos(teta6)*(sin(teta4)*(cos(teta1)*sin(teta3) -
cos(teta3)*sin(teta1)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(teta1)) +
sin(teta6)*(cos(teta5)*(cos(teta4)*(cos(teta1)*sin(teta3)
- cos(teta3)*sin(teta1)*sin(teta2)) -
cos(teta2)*sin(teta1)*sin(teta4)) +
```

```

sin(teta5)*(cos(tetal)*cos(teta3) +
sin(tetal)*sin(teta2)*sin(teta3))) +
a6*sin(teta6)*(cos(teta5)*(cos(teta4)*(cos(tetal)*sin(tet
a3) - cos(teta3)*sin(tetal)*sin(teta2)) -
cos(teta2)*sin(tetal)*sin(teta4)) +
sin(teta5)*(cos(tetal)*cos(teta3) +
sin(tetal)*sin(teta2)*sin(teta3))) +
d3*cos(teta2)*sin(tetal) +
a6*cos(teta6)*(sin(teta4)*(cos(tetal)*sin(teta3) -
cos(teta3)*sin(tetal)*sin(teta2)) +
cos(teta2)*cos(teta4)*sin(tetal)))/100 +
(981*m2*m2c*cos(teta2)*sin(tetal))/100 +
(981*m3*m3c*cos(teta2)*sin(tetal))/100;
g2 = (981*cos(tetal)*(d3*m4*sin(teta2) + d3*m5*sin(teta2)
+ d3*m6*sin(teta2) + m2*m2c*sin(teta2) +
m3*m3c*sin(teta2) + d5*m6*cos(teta4)*sin(teta2) +
m4*m4c*cos(teta4)*sin(teta2) +
m5*m5c*cos(teta4)*sin(teta2) +
a6*m6*cos(teta4)*cos(teta6)*sin(teta2) +
d5*m6*cos(teta2)*cos(teta3)*sin(teta4) +
m4*m4c*cos(teta2)*cos(teta3)*sin(teta4) +
m5*m5c*cos(teta2)*cos(teta3)*sin(teta4) -
m6*m6c*cos(teta4)*cos(teta6)*sin(teta2) +
a6*m6*cos(teta2)*cos(teta3)*cos(teta6)*sin(teta4) -
m6*m6c*cos(teta2)*cos(teta3)*cos(teta6)*sin(teta4) -
a6*m6*cos(teta2)*sin(teta3)*sin(teta5)*sin(teta6) -
a6*m6*cos(teta5)*sin(teta2)*sin(teta4)*sin(teta6) +
m6*m6c*cos(teta2)*sin(teta3)*sin(teta5)*sin(teta6) +
m6*m6c*cos(teta5)*sin(teta2)*sin(teta4)*sin(teta6) +
a6*m6*cos(teta2)*cos(teta3)*cos(teta4)*cos(teta5)*sin(tet
a6) -
m6*m6c*cos(teta2)*cos(teta3)*cos(teta4)*cos(teta5)*sin(te
ta6)))/100;
g3 =
(981*m6*(m6c*(sin(teta6)*(sin(teta5)*(sin(tetal)*sin(teta
3) + cos(tetal)*cos(teta3)*sin(teta2)) -
cos(teta4)*cos(teta5)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3))) -
cos(teta6)*sin(teta4)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3))) +
d5*sin(teta4)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3)) -
a6*sin(teta6)*(sin(teta5)*(sin(tetal)*sin(teta3) +
cos(tetal)*cos(teta3)*sin(teta2)) -
cos(teta4)*cos(teta5)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3))) +
a6*cos(teta6)*sin(teta4)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3)))/100 +
(981*m4*m4c*sin(teta4)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3)))/100 +

```

```

(981*m5*m5c*sin(teta4)*(cos(teta3)*sin(tetal) -
cos(tetal)*sin(teta2)*sin(teta3)))/100;
g4 = (981*m4*m4c*(cos(teta4)*(sin(tetal)*sin(teta3) +
cos(tetal)*cos(teta3)*sin(teta2)) +
cos(tetal)*cos(teta2)*sin(teta4))/100 -
(981*m6*(m6c*(cos(teta6)*(cos(teta4)*(sin(tetal)*sin(teta
3) + cos(tetal)*cos(teta3)*sin(teta2)) +
cos(tetal)*cos(teta2)*sin(teta4)) -
cos(teta5)*sin(teta6)*(sin(teta4)*(sin(tetal)*sin(teta3)
+ cos(tetal)*cos(teta3)*sin(teta2)) -
cos(tetal)*cos(teta2)*cos(teta4)) -
d5*(cos(teta4)*(sin(tetal)*sin(teta3) +
cos(tetal)*cos(teta3)*sin(teta2)) +
cos(tetal)*cos(teta2)*sin(teta4)) -
a6*cos(teta6)*(cos(teta4)*(sin(tetal)*sin(teta3) +
cos(tetal)*cos(teta3)*sin(teta2)) +
cos(tetal)*cos(teta2)*sin(teta4)) +
a6*cos(teta5)*sin(teta6)*(sin(teta4)*(sin(tetal)*sin(teta
3) + cos(tetal)*cos(teta3)*sin(teta2)) -
cos(tetal)*cos(teta2)*cos(teta4)))/100 +
(981*m5*m5c*(cos(teta4)*(sin(tetal)*sin(teta3) +
cos(tetal)*cos(teta3)*sin(teta2)) +
cos(tetal)*cos(teta2)*sin(teta4))/100;
g5 = -(981*m6*sin(teta6)*(a6 -
m6c)*(cos(tetal)*cos(teta5)*sin(teta2)*sin(teta3) -
cos(teta3)*cos(teta5)*sin(tetal) +
cos(tetal)*cos(teta2)*sin(teta4)*sin(teta5) +
cos(teta4)*sin(tetal)*sin(teta3)*sin(teta5) +
cos(tetal)*cos(teta3)*cos(teta4)*sin(teta2)*sin(teta5)))/
100;
g6 = (981*m6*(a6 -
m6c)*(cos(tetal)*cos(teta2)*cos(teta4)*sin(teta6) +
cos(teta3)*cos(teta6)*sin(tetal)*sin(teta5) -
sin(tetal)*sin(teta3)*sin(teta4)*sin(teta6) +
cos(tetal)*cos(teta2)*cos(teta5)*cos(teta6)*sin(teta4) +
cos(teta4)*cos(teta5)*cos(teta6)*sin(tetal)*sin(teta3) -
cos(tetal)*cos(teta3)*sin(teta2)*sin(teta4)*sin(teta6) -
cos(tetal)*cos(teta6)*sin(teta2)*sin(teta3)*sin(teta5) +
cos(tetal)*cos(teta3)*cos(teta4)*cos(teta5)*cos(teta6)*si
n(teta2))/100;

out(1) = g1;
out(2) = g2;
out(3) = g3;
out(4) = g4;
out(5) = g5;
out(6) = g6;

```

end



## ÖZGEÇMİŞ

**Ad Soyad:** Musa Özgün GÜLEÇ

**Doğum Yeri ve Tarihi :** KONYA / 21.07.1989

**Adres :** Kayışdağı Mh. Çobançeşme Sk. No:29 Daire :10  
ATAŞEHİR / İstanbul

**E-posta :** ozgun\_gulec@yahoo.com

**Lisans :** Makine Mühendisliği

### **Yayın ve Patent Listesi :**

- Türk Patent Enstitüsü, Patent No: 2008/09679, 2,5 eksenli halatlı iskele sistemi